

# HybridGS: Decoupling Transients and Statics with 2D and 3D Gaussian Splatting

Jingyu Lin<sup>1</sup> Jiaqi Gu<sup>2</sup> Lubin Fan<sup>2</sup> Bojian Wu<sup>3</sup> Yujing Lou<sup>4</sup>

Renjie Chen<sup>1</sup> Ligang Liu<sup>1</sup> Jieping Ye<sup>2</sup>

<sup>1</sup>University of Science and Technology of China <sup>2</sup>Individual Researcher

<sup>3</sup>Zhejiang University <sup>3</sup>Shanghai Jiaotong University

<https://gujiaqivadin.github.io/hybrids/>



Figure 1. **HybridGS** is the first hybrid representation that combines multi-view consistent 3D Gaussians and single-view independent 2D Gaussians, which is used to decouple the transients and statics presented in the scene. Our results demonstrate reasonable decompositions.

## Abstract

*scenes, even in the presence of distracting elements.*

Generating high-quality novel view renderings of 3D Gaussian Splatting (3DGS) in scenes featuring transient objects is challenging. We propose a novel hybrid representation, termed as **HybridGS**, using 2D Gaussians for transient objects per image and maintaining traditional 3D Gaussians for the whole static scenes. Note that, the 3DGS itself is better suited for modeling static scenes that assume multi-view consistency, but the transient objects appear occasionally and do not adhere to the assumption, thus we model them as **planar objects from a single view**, represented with 2D Gaussians. Our novel representation decomposes the scene from the perspective of fundamental viewpoint consistency, making it more reasonable. Additionally, we present a novel multi-view regulated supervision method for 3DGS that leverages information from co-visible regions, further enhancing the distinctions between the transients and statics. Then, we propose a straightforward yet effective multi-stage training strategy to ensure robust training and high-quality view synthesis across various settings. Experiments on benchmark datasets show our state-of-the-art performance of novel view synthesis in both indoor and outdoor

## 1. Introduction

3D Gaussians Splatting (3DGS) [11] has recently gained popularity for novel view synthesis due to its high-quality rendering, efficiency and low-memory cost. Its applications span across virtual reality, augmented reality, and robotics, etc [3, 6, 15, 16, 25, 31, 35, 39, 50]. Typically, most current approaches assume that the input images are posed and free of noise, which are the prerequisite needed to represent a complete scene well. However, this assumption is often not met. For example, images taken casually with a mobile phone usually contain messy dynamic objects, or transient objects. Therefore, it is hard to model the scene because there are always many transient occlusions. Technically, 3DGS is capable of modeling static scenes effectively because the static objects in images satisfy geometric consistency constraints across different viewpoints. However, transient objects do not follow this assumption. If the images containing transient objects are naively used in training 3DGS, such variations can lead to difficulties in achieving coherent blending of individual Gaussian representations,

potentially resulting in artifacts or inaccuracies.

To address this issue, numerous methods have been proposed to improve the scene representation ability involving transient objects. In the era of Neural Radiance Fields (NeRF) [20], RobustNeRF [26] introduced a robust loss function that minimizes the influence of photometrically inconsistent observations during training, thereby enhancing the overall quality of the representation. NeRF On-the-go [24] was the first to leverage the uncertainty of transients by incorporating DINOv2 [22] features into a shallow MLP. It focused on loss supervision in the areas of low uncertainty to effectively remove transients from dynamic scenes. These two methods have established widely-used benchmark datasets featuring transients, aiming to evaluate the reconstruction of clear static scenes with improved results of novel view synthesis. For 3DGS, SpotLessSplats (SLS-mlp) [27] integrated the designs of those aforementioned methods, employing adaptive method to detect outliers through clustering in the feature space, achieving the best current results.

However, these previous approaches neglect two critical issues: (1) **The ambiguity between semantics and transients**. Utilizing features from visual models to capture the semantic characteristics of transients can inherently lead to confusion in distinguishing between them. (2) **The absence of explicit transient modeling**. These methods primarily focus on achieving a robust statics by merely mitigating the impact of transients, rather than comprehensively modeling the whole scene themselves.

In response to the issues mentioned above, our key observation is that transients lack multi-view consistency and usually only appear in a single view. Therefore, we consider them as **planar objects at that view**. That is to say, this planar representation is a unique feature of the current view. Based on this, we propose **HybridGS**, a hybrid representation with 2D Gaussians for transient objects per image and 3D Gaussians for the whole static scenes as illustrated in Fig. 1, which naturally decouples the transients and statics based on their intrinsic properties. With this novel representation, while modeling transients as planar objects, the fundamental consistency of viewpoints is also maintained for statics. To further enhance the performance, a multi-view regulated supervision method for 3DGS is adopted to incorporates information from co-visible regions across multiple views, leading to more accurate representation. Additionally, we employ a simple yet effective multi-stage training strategy that ensures robust training and consistent view synthesis across various scenarios, thereby guaranteeing better stability and convergence during the training process. Extensive experiments on benchmark datasets demonstrate our state-of-the-art quality of novel view synthesis in both indoor and outdoor scenes, leading to the accurate decoupling of transients and statics as well as high efficiency

in training and inference stage. The contributions are summarized as follows.

- We are the first to introduce a novel hybrid representation that combines image-specific 2D Gaussians with static 3D Gaussians, enabling effective modeling of transient objects within casually captured images.
- We develop a multi-view supervision scheme for 3DGS that utilizes overlapping regions across multiple views, which enhances the model’s capability to distinguish between static and transient elements, ultimately improving the overall quality of the novel view synthesis.
- Our HybridGS achieves the state-of-the-art performance in benchmark datasets, which outperforms the previous methods and sets a new standard for novel view synthesis in the scenes with transients.

## 2. Related Work

### 2.1. Novel View Synthesis

NeRF (Neural Radiance Fields) [1, 2, 13, 17, 20, 21, 23, 33, 34] revolutionizes 3D scene reconstruction by representing scenes as continuous neural networks that map 3D coordinates to color and density values, which allows for the synthesis of photorealistic views from arbitrary camera positions. Recently, 3D Gaussian Splatting (3DGS) [9, 11, 12, 18, 28, 44, 45] has gained attention as an effective method for novel view synthesis. It starts with an initial point cloud and transforms it into optimizable 3D Gaussian primitives, known for their high-quality rendering and fast processing speeds. Nevertheless, existing methods require the assumption that only static scenes are present in the input images to effectively utilize multi-view geometric consistency to train a coherent scene representation.

### 2.2. Modeling Transients and Statics in NeRF/3DGS

In casually captured footage of real-world scenes, there are often numerous dynamic occlusions, such as moving pedestrians and vehicles. Many works have tried to address challenges in scenes featuring both static and transient objects. RobustNeRF [26] was the pioneer work that introduces a robust loss function that reduces the influence of photometrical inconsistency during training, thereby improving the overall reconstruction quality on its proposed dataset. Other methods [4, 19, 40] also applied the similar idea to strengthen the robustness to outliers by down-weighting or discarding inconsistent observations based on the magnitude of color residuals. NeRF On-the-go was the first to introduce DINOv2 [22] features to predict pixel uncertainty, in an attempt to achieve clean static reconstructions without transients. It also introduced a benchmark dataset of casually captured images, aimed at evaluating static reconstruction performance for novel views.

Following these two methods and their datasets, recent methods of 3DGS have also explored techniques for separating transients and statics to reconstruct the statics more effectively. For example, SpotLessSplats [27] further extended previous methods by utilizing semantic features [22] combined with robust optimization strategies to effectively eliminate transient distractors. Besides, several approaches [4, 5, 7, 8, 10, 14, 19, 36, 37, 41, 42, 46] have focused on reconstructing statics and illumination in datasets with unconstrained photo collections, that typically employ appearance embedding vectors to model the unique lighting and styles of each image, allowing for more accurate static reconstruction despite varying illumination status.

Our method specifically focuses on distinguishing between transients and statics in casually captured unordered images with minimal illumination changes. Different from previous methods [24, 26, 27], we propose a hybrid Gaussian representation with 2D and 3D Gaussians with multi-view information to model the whole 3D scene without any additional semantic features.

### 2.3. Image Representation with 2D Gaussians

Recently, inspired by 3DGS, some methods [48, 49] in the field of image representation and compression have started to use 2D Gaussians to represent images. GaussianImage [48] leverages 2D Gaussians on the pixel plane to achieve comparable reconstruction quality while offering enhanced compression and accelerated rendering capabilities. Image-GS [49] creates a content-adaptive image representation by fitting a target image by adaptively allocating and progressively optimizing a set of 2D Gaussians.

**Note that** the *2D Gaussians* used here differ from the *2D Surfels* (also referred to as *2D Gaussians*) introduced by [9]. While 2D Surfels are designed to enable perspective-correct splatting through 2D surface modeling, ray-splat intersection, and volumetric integration in 3D space, and the 2D Gaussians in GaussianImage [48] is a flexible, compact, and content-adaptive image representation within 2D space.

Here, we incorporate the trainable 2D Gaussians to learn transient and uncertainty masks during the training process, decomposing the whole scene with 2D and 3D Gaussians.

## 3. Preliminaries

**3D Gaussian Splatting (3DGS)** represent a 3D scene explicitly with a set of 3D anisotropic Gaussians  $\{\mathcal{G}_{3d}\}$ . Each Gaussian is parameterized by its centroid  $\mathbf{x}_{3d} \in \mathbb{R}^3$ , scale  $\mathbf{S}_{3d} \in \mathbb{R}^3$ , rotation matrix  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ , opacity  $\alpha_{3d} \in \mathbb{R}$  and color  $\mathbf{c}_{3d} \in \mathbb{R}^3$  encoded in spherical harmonic (SH) coefficients. The 3D covariance matrix  $\Sigma_{3d}$  of the 3D Gaussian is obtained from  $\mathbf{R}_{3d}$  and  $\mathbf{S}_{3d}$ :

$$\Sigma_{3d} = \mathbf{R}_{3d} \mathbf{S}_{3d} \mathbf{S}_{3d}^T \mathbf{R}_{3d}^T \quad (1)$$

The 3D Gaussians are defined in world space for a spatial point  $\mathbf{y}_{3d}$  following:

$$\mathcal{G}_{3d}(\mathbf{y}_{3d}) = e^{-\frac{1}{2}(\mathbf{y}_{3d} - \mathbf{x}_{3d})^T \Sigma_{3d}^{-1} (\mathbf{y}_{3d} - \mathbf{x}_{3d})} \quad (2)$$

Given  $\mathbf{J}$ , the Jacobian of the affine projective transformation and  $\mathbf{W}$ , a viewing transformation, the covariance matrix  $\Sigma'_{3d}$  in camera coordinates is defined as follows:

$$\Sigma'_{3d} = \mathbf{J} \mathbf{W} \Sigma_{3d} \mathbf{W}^T \mathbf{J}^T \quad (3)$$

These 3D Gaussians are projected to 2D imaging plane and blended through a fast, differentiable  $\alpha$ -blending process to render 2D images. The color  $\mathbf{C}_{3d}$  of each pixel is computed using N-ordered 2D splats with the following formula:

$$\mathbf{C}_{3d} = \sum_{i \in \mathbf{N}} \alpha_{3d}' \mathbf{c}_{3d_i} \prod_{j=1}^{i-1} (1 - \alpha_{3d}'_j) \quad (4)$$

where  $\alpha_{3d}'$  is the final opacity calculated by  $\alpha_{3d}' = \alpha_{3d_i} * e^{-\frac{1}{2}(\mathbf{y}'_{3d} - \mathbf{x}'_{3d})^T \Sigma'_{3d}^{-1} (\mathbf{y}'_{3d} - \mathbf{x}'_{3d})}$ . And  $\mathbf{x}'_{3d}$  and  $\mathbf{y}'_{3d}$  are coordinates in the projected space. Typically, the centroids of 3D Gaussians are initialized using the sparse SfM point clouds obtained from the set of input images.

**2D Gaussian Splatting (2DGS)** is initially introduced by [48] in the task of implicit image representation. Since 2D Gaussians  $\{\mathcal{G}_{2d}\}$  are no longer oriented to 3D scenes but to 2D space, many bloated operations and redundant parameters are discarded, such as viewing transformation, spherical harmonics, etc. Concretely, a basic 2D Gaussian is described by its centroid  $\mathbf{x}_{2d} \in \mathbb{R}^2$ , 2D covariance matrix  $\Sigma_{2d} \in \mathbb{R}^2$ , color  $\mathbf{c}_{2d} \in \mathbb{R}^3$  and opacity  $\alpha_{2d} \in \mathbb{R}$ . The 2D covariance matrix is also factorized into a rotation matrix  $\mathbf{R}_{2d} \in \mathbb{R}^{2 \times 2}$  and a scale matrix  $\mathbf{S}_{2d} \in \mathbb{R}^2$  as:

$$\Sigma_{2d} = \mathbf{R}_{2d} \mathbf{S}_{2d} \mathbf{S}_{2d}^T \mathbf{R}_{2d}^T \quad (5)$$

For rasterization, since the depth and camera parameters are not included in 2D Gaussians, a simplified accumulated summation of N-ordered 2D splats is calculated as follows:

$$\mathbf{C}_{2d} = \sum_{i \in \mathbf{N}} \alpha_{2d}' \mathbf{c}_{2d_i} \quad (6)$$

where  $\alpha_{2d}'$  is the opacity calculated by  $\alpha_{2d}' = \alpha_{2d_i} * e^{-\frac{1}{2}(\mathbf{y}_{2d} - \mathbf{x}_{2d})^T \Sigma_{2d}^{-1} (\mathbf{y}_{2d} - \mathbf{x}_{2d})}$ . And  $\mathbf{x}_{2d}$  and  $\mathbf{y}_{2d}$  are coordinates of centroid and any pixel in the 2D image space.

Specifically, 2D Gaussians can be considered as projections of 3D Gaussians from a certain perspective, illustrating the relationship of Gaussians between 2D and 3D space. In our case, we combine the single-view independent 2D Gaussians and multi-view consistent 3D Gaussians to represent the entire 3D scene, which helps fundamentally decoupling transients and statics, based on the principles of multi-view geometry.

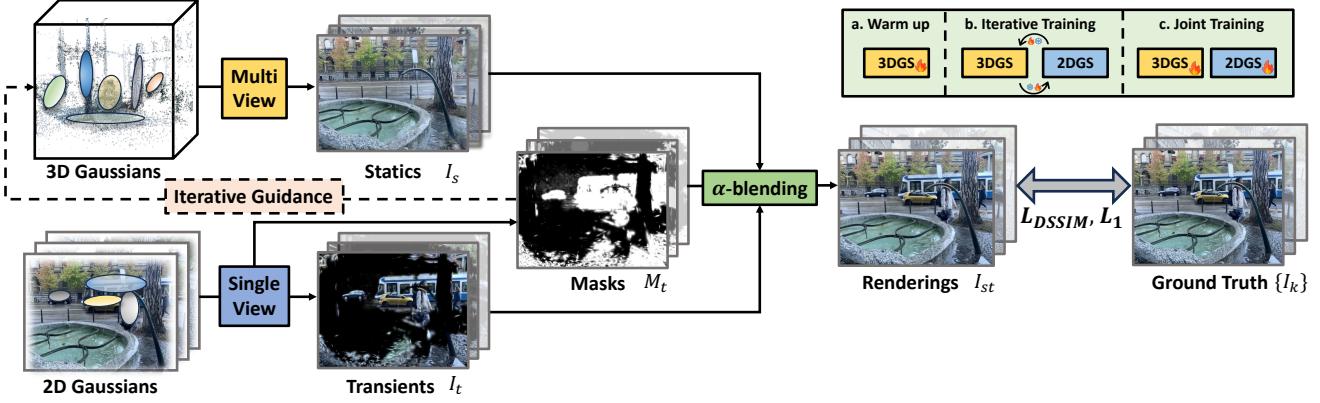


Figure 2. **Overview.** Given a casually captured image sequence, we decompose the whole scene into 2D Gaussians for transient objects and 3D Gaussians for static scenes. To warm up, we start by training a basic 3DGS to capture static elements. This is followed by iterative training of 2D and 3D Gaussians, where our transients and statics are combined using an  $\alpha$ -blending strategy with masks to produce the final renderings. The masks provide guidance for 3D Gaussians in the iterative training stage. During the joint-training, both 2D and 3D Gaussians are trained to further optimize the decomposition results.

## 4. Method

We delineate the pipeline of our approach as illustrated in Fig. 2. Essentially, given a set of input images  $\{I_k|k = 1, 2, \dots, N\}$  with the corresponding camera parameters, for each view  $I$ , the goal of our method is to reasonably decouple the transients  $I_t$  and statics  $I_s$  as follows:

$$I = M_t \odot I_t + (1 - M_t) \odot I_s, \quad (7)$$

where the  $M_t \in [0, 1]$  represents the transient mask, and  $\odot$  is per pixel multiplication. When a pixel's value in  $M_t$  approaches 1, it indicates a higher probability that the location is transient; conversely, a lower value suggests a greater likelihood that area is static.

To achieve the goal, we decompose the whole 3D scene with two components: (1) **Multi-view consistent 3D Gaussians** is used for rendering  $I_s$ , which leverages multi-view information from images and models the static scene with a set of unified 3D Gaussians. The multi-view inputs regulate the 3D Gaussians to be consistent and robust across the different views. (2) **Single-view independent 2D Gaussians** is responsible for modeling  $I_t$ , which enables our approach to handle the fact that images are casually captured with varying transients. Concretely, we form a set of view-independent 2D Gaussians to model transients as planar objects from a single view. This novel combination allows for a more precise and reasonable representation of 3D scenes, enabling better performance on novel views.

This section is organized as follows: we introduce our hybrid representation of scenes using 3D Gaussians in Sec. 4.1 and 2D Gaussians in Sec. 8.1, followed by a comprehensive overview of the training pipeline in Sec. 4.3.

### 4.1. Modeling Statics with 3D Gaussians

Generally speaking, the initialized point clouds obtained by COLMAP [29, 30] can only represent areas in the scene that satisfy multi-view consistency, so 3D Gaussians tend to reconstruct static scenes, such as buildings, grounds, etc.

However, in scenes with transients, continuously training the 3D Gaussians may cause the transient objects to be overfitted into the Gaussian kernels because of constraints imposed by the RGB loss. This approach fails to capture the true geometry and instead overemphasizes the RGB information, leading to annoying artifacts from other viewpoints and ultimately impacting quality of novel view synthesis.

We address this problem by adopting an enhanced multi-view regulated supervision scheme. Unlike previous methods that rasterize and supervise only a single image per iteration, our approach introduces two major changes. (1) We increase the number of images per iteration to  $K$ , enabling gradient back-propagation to simultaneously consider mutual information from multiple views. This batch-wise inputs allow each optimization step to leverage multi-view insights to distinguish and infer transients and static elements. (2) We adopt a sparse training schedule that focuses optimization only on the 3D Gaussians within the co-visible areas of the cross-view frustums. This method not only sharpens the training focus but also reduces computational costs, as detailed in Alg. 1, and outputs rasterized  $\hat{I}_s$ .

### 4.2. Modeling Transients with 2D Gaussians

Different from other methods that utilize semantic features, we introduce 2D Gaussians to model the transients as planar objects from a single view for each image. This approach is motivated by several key considerations: (1) 2DGS has been proven to be an efficient method for representing images,

---

**Algorithm 1:** Multi-view Regulated Supervision.

---

**Input:** A set of images  $\{I_k, k = 1, 2, \dots, N\}$ ;  
 Multi-view size  $K$ ; Maximum iterations  $T$ ;  
**Output:** Optimized 3D Gaussian parameters  $\theta_{3d}$   
 within the co-visible regions;

Initialize  $\theta_{3d}$  ;  
**for** each iteration  $t = 1$  to  $T$  **do**  
 |  $I_{1,\dots,K} \leftarrow \text{RandomSample}(K, N)$ ;  
 |  $F_{\text{inter}} \leftarrow \text{ComputeFrustum}(I_1)$ ;  
 | **for**  $i \leftarrow 2$  to  $K$  **do**  
 | |  $F_{\text{inter}} \leftarrow F_{\text{inter}} \cap \text{ComputeFrustum}(I_i)$ ;  
 | **end**  
 |  $\mathcal{G}_{3d} \leftarrow \text{ExtractGaussians}(I, F_{\text{inter}})$ ;  
 | **for** each Gaussian  $G_j \in \mathcal{G}_{3d}$  **do**  
 | |  $\theta \leftarrow \theta - \eta \cdot \nabla \mathcal{L}(G_j(\theta))$ ;  
 | **end**  
**end**

---

capable of modeling an arbitrary image effectively. (2) The formation and rasterization processes of 2DGS closely resemble those of 3DGS, allowing both to be expressed within a unified framework. (3) Degenerating 3D objects that do not satisfy multi-view geometric consistency into 2D representation can effectively decouple transients from static scenes on a fundamental level.

Specifically, 2D Gaussians are rasterized into an image  $\hat{I}_t$  with a transient mask  $\hat{M}_t$  for modeling uncertainty, each pixel  $y_{2d}$  of which is calculated as:

$$\hat{M}_t(y_{2d}) = \sum_{i \in \mathbb{N}} \alpha_{2d,i}' = \sum_{i \in \mathbb{N}} \alpha_{2d,i} \cdot e^{-\frac{1}{2}(y_{2d} - x_{2d})^T \Sigma_{2d}^{-1} (y_{2d} - x_{2d})}. \quad (8)$$

It is noteworthy that we adopt uncertainty of 2D Gaussians to model the mask, rather than using 3D results. Because after modeling a static scene with 3D Gaussians, the 2D Gaussians are responsible for learning the residual part of the image. Therefore, naturally, the results rendered by the 2D Gaussian implicitly include a mask for transients.

### 4.3. Multi-stage Training Scheme

Because 2D and 3D Gaussians are trained within the same framework, it is crucial to balance their relationship effectively. To that end, we have proposed a multi-stage training strategy that ensures stable convergence and enhances overall performance.

**Warm up Pre-training.** The training process begins by using 3DGS to capture the essential structure of the entire static scene. This stage focuses on modeling the scene with a set of unified 3D Gaussians with our multi-view regulated supervision as detailed in Sec. 4.1, producing foundational, albeit low-quality, rendered images  $\hat{I}_s$  using a combination of DSSIM [38] and L1 losses following:

$$\mathcal{L}_{\text{warmup}} = \lambda L_{\text{DSSIM}}(\hat{I}_s, I) + (1 - \lambda) L_1(\hat{I}_s, I). \quad (9)$$

where the  $\lambda$  is set to 0.2. During the warm-up, the densification of 3D Gaussians is conducted to gradually increase the number, allowing for a more detailed representation of the static scene structure.

**Iterative Training.** The second stage employs iterative training between 2DGS and 3DGS to progressively optimize both components. During the alternation, the 3DGS will provide more accurate static scene renderings, which in turn enables the 2DGS to refine its transient representations  $\hat{I}_t$  and mask predictions  $\hat{M}_t$  with Eq. 10. Conversely, the improved masks from 2DGS allow the 3DGS to focus more precisely on rendering the static regions of the scene with Eq. 11. The rendered image  $\hat{I}$  is a composition of  $\hat{I}_s$  and  $\hat{I}_t$  with  $\hat{M}_t$  following Eq. 7. This process iteratively refines the transients and statics, improving the overall quality and accuracy of the masks. The loss functions are as follows:

$$\mathcal{L}_{\text{iter2d}} = \lambda L_{\text{DSSIM}}(\hat{I}, I) + (1 - \lambda) L_1(\hat{I}, I), \quad (10)$$

$$\mathcal{L}_{\text{iter3d}} = (1 - \hat{M}_t) \odot \mathcal{L}_{\text{iter2d}}. \quad (11)$$

Note that, during training, 2DGS and 3DGS are trained alternately. When training one branch, the gradient backpropagation of the other branch is turned off.

**Joint Fine-tuning.** The final stage involves joint fine-tuning, integrating 2DGS and 3DGS into a unified framework. This stage focuses on fine-tuning on the transient mask and reducing the error between the rendered image  $\hat{I}$ , and the ground truth image  $I$ . The loss  $\mathcal{L}_{\text{joint}}$  used during the joint fine-tuning is a combination as follows:

$$\mathcal{L}_{\text{joint}} = \beta L_{\text{DSSIM}}(\hat{I}, I) + (1 - \beta) L_1(\hat{I}, I), \quad (12)$$

where the  $\beta$  is set to 0.2. Here, both the 2DGS and 3DGS branches will enable gradient backpropagation to simultaneously update all parameters.

## 5. Experiments

### 5.1. Experimental Setup

**Datasets.** Similar to previous methods, we evaluate our HybridGS on two challenging datasets: NeRF On-the-go [24] and RobustNeRF [26]. NeRF On-the-go contains multiple casually captured indoor and outdoor sequences with varying ratios of distractors (from 5% to over 30%). We use the version of the dataset where all images were undistorted as [14]. The RobustNeRF dataset includes several scenes exemplifying different types of distractors. To simulate capturing over extended periods, distractor objects are repositioned between frames to interfere with statically placed objects (from 1 (Statue) to 150 (Yoda)). The splits includes distractor-filled training set and distractor-free testing set.

**Implementation Details.** Our HybridGS, developed on top of open-source implementation gsplat [43] and Taming-3DGS [28], incorporates custom CUDA kernels, boosting the rasterization of  $\alpha$ -blending for 3DGS and weighted

Table 1. **Comparison on NeRF On-the-go Dataset.** For better visualization, the best results are highlighted in **bold**. Our method shows generally superior performance over state-of-the-art methods. \* indicates that the results are reproduced from the official implementation.

Method	Low Occlusion						Medium Occlusion						High Occlusion					
	Mountain			Fountain			Corner			Patio			Spot			Patio-High		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RobustNeRF [26]	17.54	0.496	0.383	15.65	0.318	0.576	23.04	0.764	0.244	20.39	0.718	0.251	20.65	0.625	0.391	20.54	0.578	0.366
NeRF On-the-go [24]	20.15	0.644	0.259	20.11	0.609	0.314	24.22	0.806	0.190	20.78	0.754	0.219	23.33	0.787	0.189	21.41	0.718	0.235
3DGS [11]	19.40	0.638	<b>0.213</b>	19.96	0.659	<b>0.185</b>	20.90	0.713	0.241	17.48	0.704	0.199	20.77	0.693	0.316	17.29	0.604	0.363
SLS-mlp* [27]	19.84	0.580	0.294	20.19	0.612	0.258	24.03	0.795	0.258	21.55	<b>0.838</b>	<b>0.065</b>	23.52	0.756	<b>0.185</b>	20.31	0.664	0.259
HybridGS(Ours)	<b>21.73</b>	<b>0.693</b>	0.284	<b>21.11</b>	<b>0.674</b>	0.252	<b>25.03</b>	<b>0.847</b>	<b>0.151</b>	<b>21.98</b>	0.812	0.169	<b>24.33</b>	<b>0.794</b>	0.196	<b>21.77</b>	<b>0.741</b>	<b>0.211</b>

Table 2. **Comparison on RobustNeRF Dataset.** For better visualization, the best results are highlighted in **bold**. Our method significantly outperforms all baseline methods. Crab (1) scene had a test set with same set of views as those in the train set, which is fixed in Crab (2).

Method	Statue			Android			Yoda			Crab (1)			Crab (2)		
	PSNR↑	SSIM↑	LPIPS↓												
RobustNeRF [26]	20.60	0.76	0.15	23.28	0.75	0.13	29.78	0.82	0.15	32.22	0.94	0.06	-	-	-
NeRF On-the-go [24]	21.58	0.77	0.24	23.50	0.75	0.21	29.96	0.83	0.24	-	-	-	-	-	-
3DGS [11]	21.02	0.81	0.16	23.11	0.81	0.13	26.33	0.91	0.14	31.80	0.96	0.08	29.74	-	-
SLS-mlp [27]	22.54	0.84	0.13	25.05	<b>0.85</b>	0.09	33.66	<b>0.96</b>	0.10	35.85	<b>0.97</b>	0.08	34.43	-	-
HybridGS(Ours)	<b>22.93</b>	<b>0.87</b>	<b>0.10</b>	<b>25.15</b>	<b>0.85</b>	<b>0.07</b>	<b>35.32</b>	<b>0.96</b>	<b>0.07</b>	<b>36.31</b>	<b>0.97</b>	<b>0.05</b>	<b>35.17</b>	<b>0.96</b>	<b>0.08</b>

Table 3. **Ablation on components in datasets with varying occlusions.** The best results are highlighted in **bold**.

Dataset(ratio)	On-the-go low.(5%)			On-the-go medium.(17%)			On-the-go high.(26%)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Ours	<b>21.42</b>	0.684	<b>0.268</b>	<b>23.50</b>	<b>0.827</b>	<b>0.167</b>	<b>23.05</b>	<b>0.768</b>	0.203
w/o iterative Training	21.39	<b>0.687</b>	0.269	23.11	0.815	0.174	22.55	0.753	0.218
w/o Joint Training	21.05	0.654	0.305	23.18	0.815	0.179	23.01	0.748	0.223
w/o Multi-view	21.02	0.658	0.300	23.06	0.812	0.176	22.71	0.763	<b>0.202</b>

summation for 2DGS. The number of views  $K$  is 4 in our practice. For a scene consisting of 100 images, the process starts with a 1k-step warm-up. This is followed by iterative training, which includes 10k steps for 2DGS and 1k steps for 3DGS. The final joint training involves 30k steps. The entire training process takes approximately 0.18 GPU hours. The number of training steps scales proportionally with the size of the dataset. All experiments are performed on a single NVIDIA RTX 4090 GPU using PyTorch.

**Metrics.** We follow common practice by employing PSNR, SSIM [38], and LPIPS [47] for evaluation.

## 5.2. Evaluation

### 5.2.1 Comparison on the NeRF On-the-go Dataset

As shown in Tab. 1 and Fig. 3, our method demonstrates superior performance across different scenarios with varying levels of occlusion. In Tab. 1, our method outperforms the previous SOTA method across multiple metrics, with an average improvement of 1.10 dB in PSNR and 5.27% in SSIM, demonstrating the superiority of our approach.

Furthermore, Fig. 3 provides a qualitative comparison, illustrating the visual fidelity of our results against other methods. Our method produces more clear and detailed images, closely resembling the ground truth, particularly in

Table 4. **Ablation on the number of 2D Gaussians of Corner in NeRF On-the-go dataset.** For better visualization, the best results are highlighted in **bold**.

Number	PSNR↑	SSIM↑	LPIPS↓	Storage (MB)	
				2DGS	3DGS
1k	24.493	0.840	<b>0.150</b>	3.5	42.1
5k	24.733	0.843	0.158	17.3	41.9
10k	<b>25.034</b>	<b>0.847</b>	0.151	34.7	42.2
15k	24.512	0.839	0.167	42.6	52
25k	24.698	0.840	0.167	69.4	42.2

challenging scenes with complex geometry and occlusions. This highlights our model’s capability to effectively handle diverse visual conditions both in indoor and outdoor scenes.

### 5.2.2 Comparison on the RobustNeRF Dataset

We present the quantitative results of RobustNeRF in Tab. 2. Our method demonstrates superior performance quantitatively compared to all previous approaches. In scenes with numerous transients, such as the Yoda scene, we’ve noticed that the performance metrics of the original 3DGS significantly degrade, occasionally even dropping below those of NeRF-related methods. By leveraging multi-view shared information and incorporating 2D Gaussians to effectively represent transients, our approach surpasses semantic-based methods like SLS-mlp [27]. As shown in Fig. 4, our results have clearer overall boundaries and structured fine-grained details, while SLS-mlp [27] may lead to blurriness in static objects and occluded areas. This further confirms that, even when faced with numerous transients, our method consistently maintains stable and reliable performance.

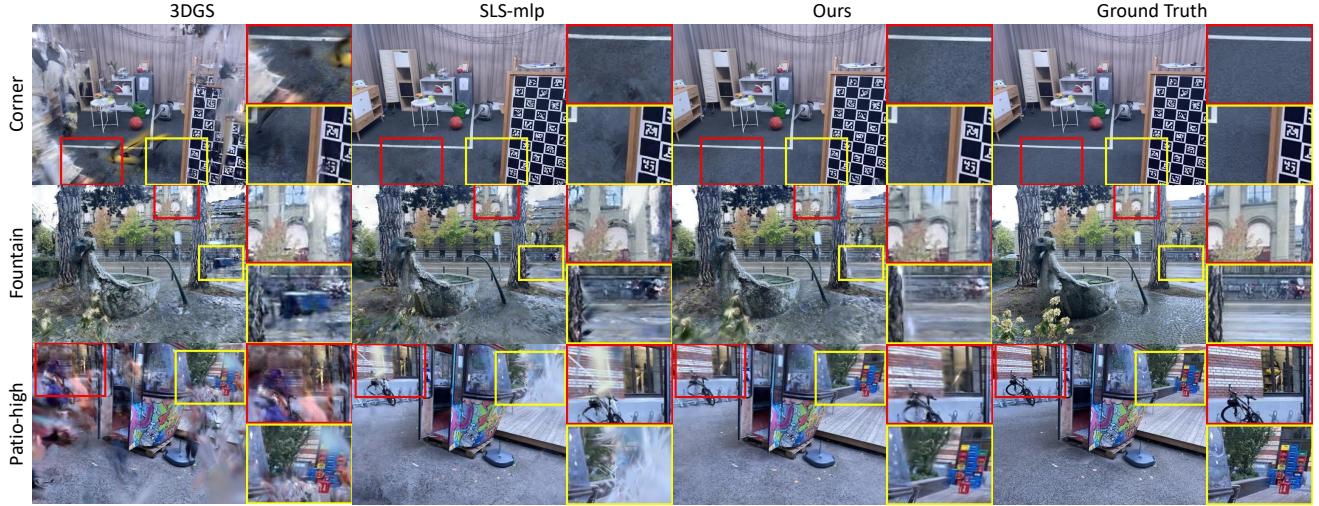


Figure 3. **Visualization of novel view synthesis results on the testing set of NeRF On-the-go dataset.** Our method demonstrates superior results by effectively reducing artifacts and providing clearer boundaries. This results in a cleaner statics compared to other methods, showcasing enhanced visual quality and precision in novel views.



Figure 4. **Visualization of novel view synthesis results on RobustNeRF dataset.**

### 5.3. Ablation Studies

**Components.** To assess the effectiveness of each component in our framework, we performed an ablation study, as presented in Tab. 3. We systematically removed iterative training, joint training, and multi-view supervision to evaluate their effectiveness. Each component, as expected, enhances overall performance, demonstrating their individual and combined significance. In scenes with low occlusion, multi-view supervision allows the model to concentrate on static elements by integrating data from various perspectives. Conversely, in highly occluded environments, joint training with 2D and 3D Gaussians markedly improves performance, highlighting the model’s capability to manage complex scenes with significant occlusions and transient changes. Iterative training consistently boosts results across different occlusion levels. In summary, the results underscore that integrating diverse training strategies and multi-view information significantly enhances the rendering quality of unseen views, especially in challenging conditions.



Figure 5. **Comparison of transient masks on NeRF On-the-go dataset.**

**The Number of 2D Gaussians.** Tab. 4 presents an ablation study on the number of 2D Gaussians used in the corner sequence of NeRF On-the-go dataset. The results indicate that increasing the number of 2D Gaussians initially improves the performance. Specifically, 10k 2D Gaussians achieve the highest PSNR (25.034) and SSIM (0.847), while maintaining a competitive LPIPS score of 0.151 and reasonable storage requirements. Based on these findings, we select 10k as the optimal number in our experiments.

**Transients and Masks.** Benefiting from our method’s explicit modeling of transient objects, we can obtain transient masks without introducing any segmentation networks or other pre-trained features. Typically, these masks capture dynamic elements such as pedestrians and vehicles, as well as some structural details ignored by 3D Gaussians. In Fig. 6, our approach effectively learns the RGB and mask of transients even under strong occlusions, while maintaining the robustness of static elements. This capability underscores the strength of our method in discerning and isolating dynamic components within complex scenes.

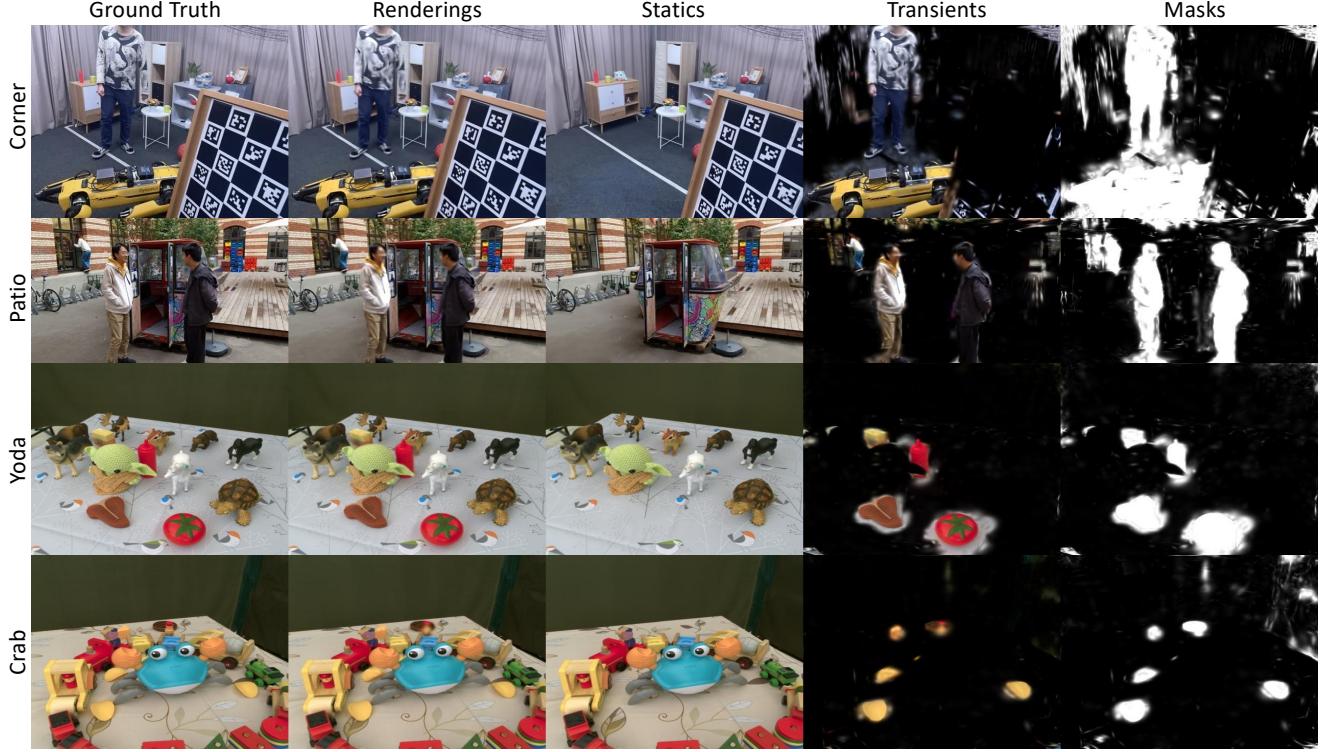


Figure 6. **Visualization of scene decomposition into transients and statics.** Our method achieves superior transient mask separation in both indoor and outdoor scenes. It effectively separates transients and statics, and the resulting renderings closely resemble the ground truth images, demonstrating its effectiveness. Top two rows are from NeRF On-the-go [24], bottom two rows are from RobustNeRF [26].



Figure 7. **Visualization on Photo Tourism dataset.**

We also compare our transient masks with SLS-mlp [27] semantic masks in Fig. 5. It is observed that the semantic mask does not perfectly align with the transients due to the semantic features not being sufficiently precise. Additionally, transients often encompass low-level information, such as motion blur or object shadows, which may not explicitly exhibit clear semantic features. By modeling these transients using 2D Gaussians at the pixel level, our method allows for a more fine-grained capture of transients.

#### 5.4. Discussions

While our method has demonstrated strong performance in distinguishing between transients and statics, it has not yet

accounted for the illumination variation in unconstrained photo collections. As illustrated in Fig. 11 on Photo Tourism [32], our method renders fine-grained architectural structures, but the overall photometric result tends towards an average photometric effect, akin to overcast conditions. A promising direction is to incorporate an appearance embedding module in our method for the future work.

## 6. Conclusion

In this paper, we introduce HybridGS, a novel hybrid representation with 2D and 3D Gaussians for decomposing 3D scenes from casually captured images. By introducing the hybrid Gaussian representation, our method addresses the challenges of handling transients as planar objects while preserving the integrity of the static scene. The multi-view supervision mechanism we developed plays a crucial role in enhancing the discrimination between static and transient elements across multiple views, reducing artifacts and inaccuracies in novel views. We have developed a multi-stage training strategy for the joint training of 2D and 3D Gaussians, ensuring the stable and robust training stage. Our experiments demonstrate that HybridGS significantly outperforms existing methods in terms of quality and efficiency.

In summary, HybridGS establishes a novel, robust and efficient foundation for 3D scene representation, paving the way for advancements in various vision-based applications.

## References

- [1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. [2](#)
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. [2](#), [1](#)
- [3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19457–19467, 2023. [1](#)
- [4] Jiahao Chen, Yipeng Qin, Lingjie Liu, Jiangbo Lu, and Guanbin Li. Nerf-hugs: Improved neural radiance fields in non-static scenes using heuristics-guided segmentation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19436–19446, 2024. [2](#), [3](#), [1](#)
- [5] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Feng Ying, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12933–12942, 2021. [3](#), [1](#)
- [6] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21401–21412, 2023. [1](#)
- [7] Hiba Dahmani. Swag: Splatting in the wild images with appearance-conditioned gaussians. In *ECCV*, 2024. [3](#), [1](#)
- [8] Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12479–12488, 2023. [3](#)
- [9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. [2](#), [3](#)
- [10] Karim Kassab, Antoine Schnepf, Jean-Yves Franceschi, Laurent Caraffa, Jeremie Mary, and Valérie Gouet-Brunet. Refinedfields: Radiance fields refinement for unconstrained scenes. *arXiv preprint arXiv:2312.00639*, 2023. [3](#)
- [11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. [1](#), [2](#), [6](#), [3](#)
- [12] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024. [2](#)
- [13] Jonávs. Kulhánek and Torsten Sattler. Tetra-nerf: Representing neural radiance fields using tetrahedra. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18412–18423, 2023. [2](#)
- [14] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. 2024. [3](#), [5](#), [1](#)
- [15] Pengzhi Li, Chengshuai Tang, Qinxuan Huang, and Zhiheng Li. Art3d: 3d gaussian splatting for text-guided artistic scenes generation, 2024. [1](#)
- [16] Zhihao Liang, Qi Zhang, Yingfa Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21644–21653, 2023. [1](#)
- [17] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5166–5175, 2024. [2](#)
- [18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. [2](#)
- [19] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2020. [2](#), [3](#), [1](#)
- [20] 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. In *ECCV*, 2020. [2](#)
- [21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. [2](#)
- [22] Maxime Oquab, Timothée Darcret, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaddin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINoV2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. [2](#), [3](#)
- [23] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Tom Funkhouser, and Vittorio Ferrari. Urban radiance fields. *CVPR*, 2022. [2](#)
- [24] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#), [5](#), [6](#), [8](#), [1](#), [4](#)

- [25] Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Angela Dai, and Matthias Nießner. L3dg: Latent 3d gaussian diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. 1
- [26] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J. Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20626–20636, 2023. 2, 3, 5, 6, 8, 1
- [27] Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David J. Fleet, and Andrea Tagliasacchi. SpotLessSplats: Ignoring distractors in 3d gaussian splatting. *arXiv:2406.20055*, 2024. 2, 3, 6, 8, 1
- [28] Saswat Mallick and Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. 2, 5
- [29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4
- [30] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 4
- [31] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 1
- [32] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press. 8, 1, 4, 5
- [33] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8238–8248, 2022. 2
- [34] Matthew Tancik, Ethan Weber, Evonne Ng, Rui long Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2
- [35] Linhan Wang, Kai Cheng, Shuo Lei, Shengkun Wang, Wei Yin, Chenyang Lei, Xiaoxiao Long, and Chang-Tien Lu. Dc-gaussian: Improving 3d gaussian splatting for reflective dash cam videos. *NeurIPS 2024*, 2024. 1
- [36] Shuaixian Wang, Haoran Xu, Yaokun Li, Jiwei Chen, and Guang Tan. Ie-nerf: Inpainting enhanced neural radiance fields in the wild. *ArXiv*, abs/2407.10695, 2024. 3
- [37] Yuze Wang, Junyi Wang, and Yue Qi. We-gs: An in-the-wild efficient 3d gaussian representation for unconstrained photo collections, 2024. 3, 1
- [38] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 5, 6
- [39] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 1
- [40] Tianhao Walter Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2neRF: Self-supervised decoupling of dynamic and static objects from a monocular video. In *Advances in Neural Information Processing Systems*, 2022. 2
- [41] Congrong Xu, Justin Kerr, and Angjoo Kanazawa. Splatfacto-w: A nerfstudio implementation of gaussian splatting for unconstrained photo collections. *ArXiv*, abs/2407.12306, 2024. 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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15855–15865, 2023. 3
- [43] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 5
- [44] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. 2
- [45] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. 2
- [46] 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. 3, 1
- [47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6
- [48] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, 2024. 3
- [49] Yunxiang Zhang, Alexandr Kuznetsov, Akshay Jindal, Kenneth Chen, Anton Sochenov, Anton Kaplanyan, and Qi Sun. Image-gs: Content-adaptive image representation via 2d gaussians, 2024. 3
- [50] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian:

Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. [1](#)

# HybridGS: Decoupling Transients and Statics with 2D and 3D Gaussian Splatting

## Supplementary Material

### 7. Positioning of Our Work

In the community, there are currently two predominant approaches for tackling the challenge of novel view synthesis in wild images with different complexities. We distinguish these approaches according to the primary datasets they utilize and provide a detailed comparison in Tab. 5.

NeRF On-the-go [24] processes casually captured images that lack inter-frame continuity, with the goal of eliminating the interference from transient objects to reconstruct statics.

Photo Tourism [32] gathers photo collections from the web, resulting in completely unconstrained conditions with more complex lighting variations and increased foreground interference. It focuses more on integrating appearance embedding to model the photometric changes in the scene.

In summary, our method belongs to the first category and aims to decompose transients and statics from casually captured images in scenes with minimal illumination changes. Experiments have demonstrated our state-of-the-art results on two widely used benchmark datasets, such as NeRF On-the-go [24] and RobustNeRF [26]. Handling varying lighting conditions will be our future work as discussed in the paper.

## 8. More Discussions

### 8.1. 2D Gaussians

The fitting capability of 2D Gaussians is inherited from 3D Gaussians. Given  $\mathbf{J}$ , the Jacobian of the affine projective transformation, and  $\mathbf{W}$ , the viewing transformation, the 3D Gaussians can be projected to 2D image plane and blended through a fast, differentiable  $\alpha$ -blending process to render 2D images following the Eq. 3 and Eq. 4. Therefore, the 2D Gaussians can be viewed as the projection of 3D Gaussians.

During training, the warm-up allows 3DGS to establish an initial model of the entire scene. It is noteworthy that intuitively, the residuals between the results of 3DGS rendering and the ground-truth would potentially model transients. However, the 3DGS itself is constrained only by RGB loss, therefore, the transient objects from different viewpoints are eventually fitted into the 3D Gaussians, leading to less effective fitting of static scenes with vanilla 3DGS. We address this issue by incorporating additional 2D Gaussians. During the iterative training stage, the 2DGS learns the residuals per view, focusing more on the unique elements of each image. The output soft mask or matting can effectively direct 3DGS to concentrate on areas with smaller residuals,

Table 5. Comparison of NeRF On-the-go and Photo Tourism.

	NeRF On-the-go [24]	Photo Tourism [32]
Data source	Casually captured photos	Web photos in the wild
Photometric	Similar lighting	Varying lighting over time
Scene	Indoor and outdoor scenes	Mostly outdoor scenes
Evaluation	Statics	Statics with their illumination
Related Works	[14, 24, 27] & Our HybridGS	[4, 5, 7, 19, 37, 46]

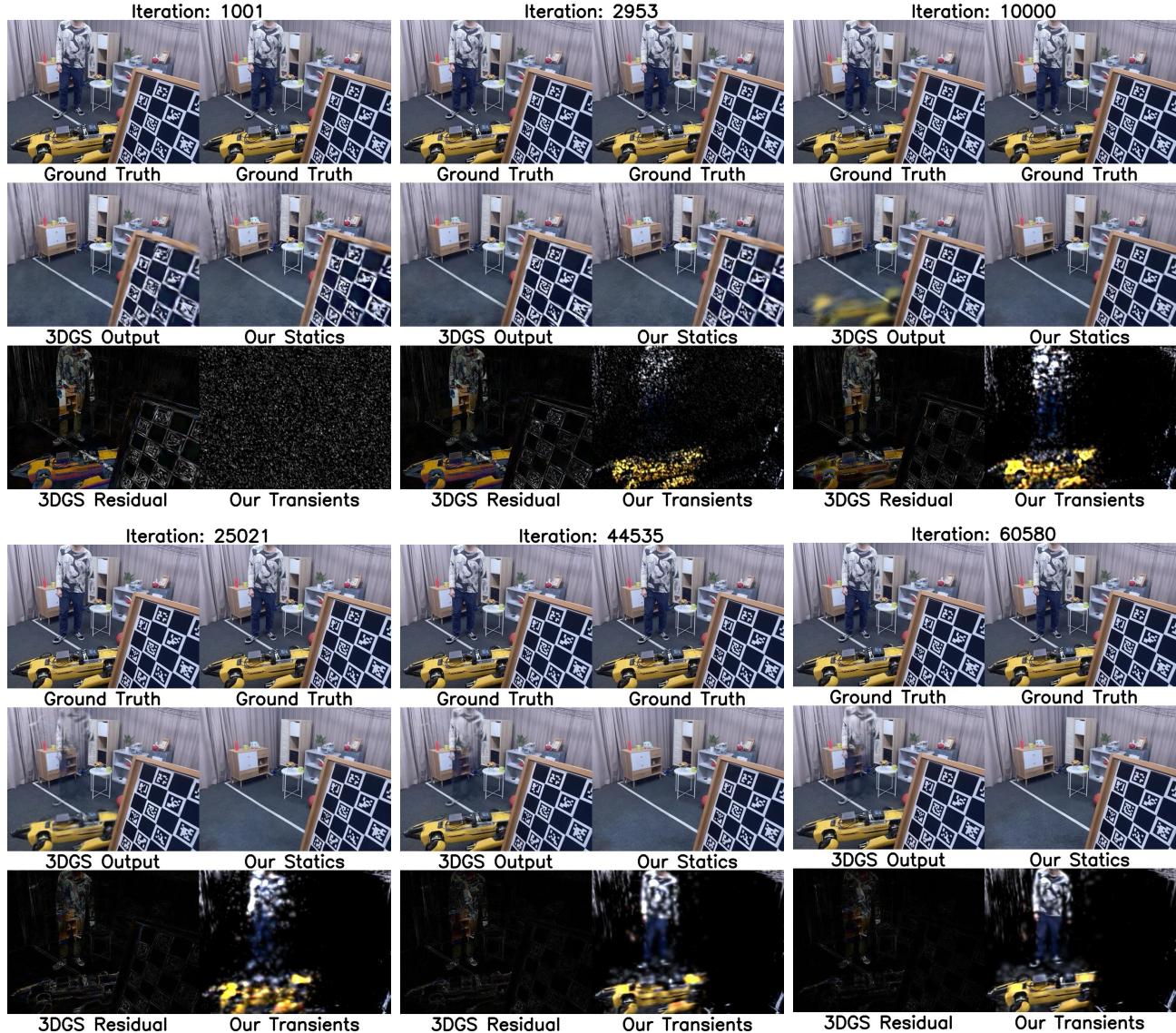
Table 6. We conduct ablation studies on a pure static scene *Garden* from the MipNeRF 360 dataset [2], and *Corner*, scene that includes both transients and statics, from the NeRF On-the-go dataset [24], to explore the potential influence of our designs.

Method Settings	Garden			Corner		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS	29.323	0.924	0.050	20.148	0.686	0.290
+ Multi-view	29.572	0.925	0.053	21.758	0.769	0.235
+ Multi-view + 2DGS	29.512	0.924	0.054	25.020	0.842	0.165

als, which represent the common and shared elements of the scene. In the final joint training stage, we perform a deep integration of 2D and 3D Gaussians for fine-tuning. Therefore, in our method, 3D Gaussians tend to learn the elements that are consistent across different viewpoints, which we define as statics. Meanwhile, 2D Gaussians capture image-specific information, such as dynamic objects and occlusions, referred to as transients.

### 8.2. Multi-view Supervision

We have further investigated the performance of multi-view 3DGS in different scenarios in Tab. 6. To be specific, we select the static scene *Garden* from the MipNeRF 360 [2] dataset and the dynamic scene *Corner* from the NeRF On-the-go [24] dataset. The results indicate that in static scenes, employing multi-view supervision has minimal impact on achieving the best results for novel view synthesis. However, during training, it is noted that 3DGS tends to over-fit the training views, leading to a gradual decline in both visual quality and metric performance for novel views. In contrast, multi-view 3DGS demonstrates more stable convergence and effectively reduces the over-fitting problem. In dynamic scenes, obviously, 3DGS is prone to over-fitting, which adversely affects performance in novel view synthesis. On the other hand, multi-view 3DGS benefits from mutual supervision in areas visible to multiple views, significantly improving the visual results.



**Figure 8. Qualitative results compared to 3DGS at each randomly selected training step.** For HybridGS, the training steps for each stage are: Warm-up: 0~1,010, Iterative Training: 1,010~40,400, Joint Training: 40,400~60,600. Notably, this scene contains 101 images. Referring to Fig. 10, before the warm-up step (the top left), during the initial rough training phase, both our and 3DGS results are somewhat blurry. However, since we adopt a multi-view strategy, the occluded parts in our results are slightly clearer. As training progresses, by the 2953rd iteration (the top central), 3DGS reaches its optimum. Nevertheless, at this point, the background in the transients remains quite blurry for 3DGS, whereas our approach has already transitioned into the iterative training phase, allowing us to model static elements more accurately. Moving forward, we maintain stable training (top right and bottom left), largely due to our introduction of 2D Gaussians to decouple transients from statics. This effectively prevents over-fitting to transients that 3DGS begins to experience, leading to diminishing rendering quality. By the 44535th iteration (the bottom central), during the joint optimization phase, our results reach their optimum. The bottom right shows the results at the end of the training process.

## 9. More Implementation Details

### 9.1. Training

For the training of 3D Gaussians, we perform the densification of 3D Gaussians during the warm-up stage. Then, in the subsequent stages, we maintain a constant number

of existing 3D Gaussians and focus solely on optimizing their parameters. For 2D Gaussians, we maintain a constant number 10,000 per image throughout the entire training process without any densification. During the iterative training process, while optimizing 3DGS with 2DGS held fixed, we binarize the uncertainty mask obtained from 2DGS into 0s



Figure 9. Visualization on 6 remaining scenes of NeRF On-the-go [24] dataset.

and 1s using a threshold value of  $\epsilon = 0.1$ .

## 9.2. Datasets

We follow the same training/testing split and resolution settings as the official rules in NeRF On-the-go [24] and RobustNeRF [26]. Specifically, for the NeRF On-the-go dataset, we downsample images from most scenes by  $8 \times$  to  $504 \times 378$ . Note that *Arcdetriomphe* and *Patio* are downsampled by  $4 \times$  to  $480 \times 270$ . For the RobustNeRF dataset, all scenes are downsampled by  $8 \times$ , with *Android* and *Statue* resized to  $503 \times 377$ , and *Crab* and *Yoda* to  $431 \times 431$ .

Table 7. Comparison of storage and computational efficiency during the training and testing. We provide a comprehensive statistical evaluation on the *Corner* of NeRF On-the-go dataset.

Method	Storage (MB)		Training		Testing FPS (Hz)
	3DGS	2DGS	Time (Hours)	Memory (GB)	
3DGS [11]	410.70	0	0.35	3.21	116
Ours	42.20	34.70	0.18	2.20	160

## 9.3. Storage

Tab. 7 highlights the advantages of our method in terms of storage and computational efficiency. Compared to 3DGS,



Figure 10. **The PSNR on testing set of Corner during training process.** Training steps for each stage of our HybridGS: Warm-up : 0~1,010, Iterative Training : 1,010~40,400,

Joint Training : 40,400~60,600. We use the same data as in Fig. 8. Note that, 3DGS reaches its optimum at step 2953, but as training continues, 3DGS tends to overfit transients in dynamic scenes, leading to gradual decline in performance. In comparison, our method is able to train steadily. This directly validates the statements in Sec. 8.1 and 10.1.

our method use 2D Gaussians to model transients instead of forcing 3D Gaussians to fit them, significantly reducing both storage and computational requirements, showcasing its superior efficiency in both training and testing phases.

## 10. More Visualization Results

### 10.1. Training Process

To better demonstrate the changes during our training process, we select `IMG_7195.JPG` of *Corner* from NeRF On-the-go dataset as example, visualizing the statics and transients during different training stages and comparing them with vanilla 3DGS in Fig. 8. As training iterations increase, 3DGS tends to gradually integrate transient elements into the static components, rendering the residuals being almost incapable of capturing transient contents. In contrast, our HybridGS effectively distinguishes transients from statics over time, leading to consistent improvements in Fig. 10.

### 10.2. More Scenes

In addition to providing metrics and results on the 6 commonly used scenes of NeRF On-the-go [24] dataset, we also present the visualization results on the remaining scenes as shown in Fig. 9. These complex scenes include some variations in lighting and shadows. We find that in addition to removing dynamic objects, our statics can also eliminate elements lacking specific semantics, such as shadows of pedestrians (in *Drone* and *Train Station*) and cars (in *Arcdetriomphe* and *Train*). This separation of non-semantic transients illustrates that our method is fundamentally a versatile, low-level and semantics-free scene decomposition approach, effectively highlighting its generality and robustness.

### 10.3. More Datasets

We apply our method on Photo Tourism [32] dataset, which consists of unconstrained photo collections with photometric variations. As shown in Fig. 11, we have some intriguing and reasonable observations. First, the statics generated using 3D Gaussians are rendered under an average light condition derived from the training images, similar to the diffuse lighting on an overcast day. Additionally, we discover that besides modeling dynamic objects, 2D Gaussians also capture photometric differences in our transients, since the illumination difference is indeed a per-image characteristic. This finding perfectly aligns with the perspective we presented in Sec. 8.1 that transients can capture unique aspects of each image, broadening the scope for future research to further isolate photometric information from 2D Gaussians.



Figure 11. Visualization on Photo Tourism [32].