Baking Gaussian Splatting into Diffusion Denoiser for Fast and Scalable Single-stage Image-to-3D Generation and Reconstruction

Yuanhao Cai[1,*], He Zhang[2], Kai Zhang[2], Yixun Liang[3,†],
Mengwei Ren[2], Fujun Luan[2], Qing Liu[2], Soo Ye Kim[2], Jianming Zhang[2],
Zhifei Zhang[2], Yuqian Zhou[2], Yulun Zhang[4,†], Xiaokang Yang[4], Zhe Lin[2], Alan Yuille[1]
[1] Johns Hopkins University, [2] Adobe Research, [3] HKUST, [4] Shanghai Jiao Tong University

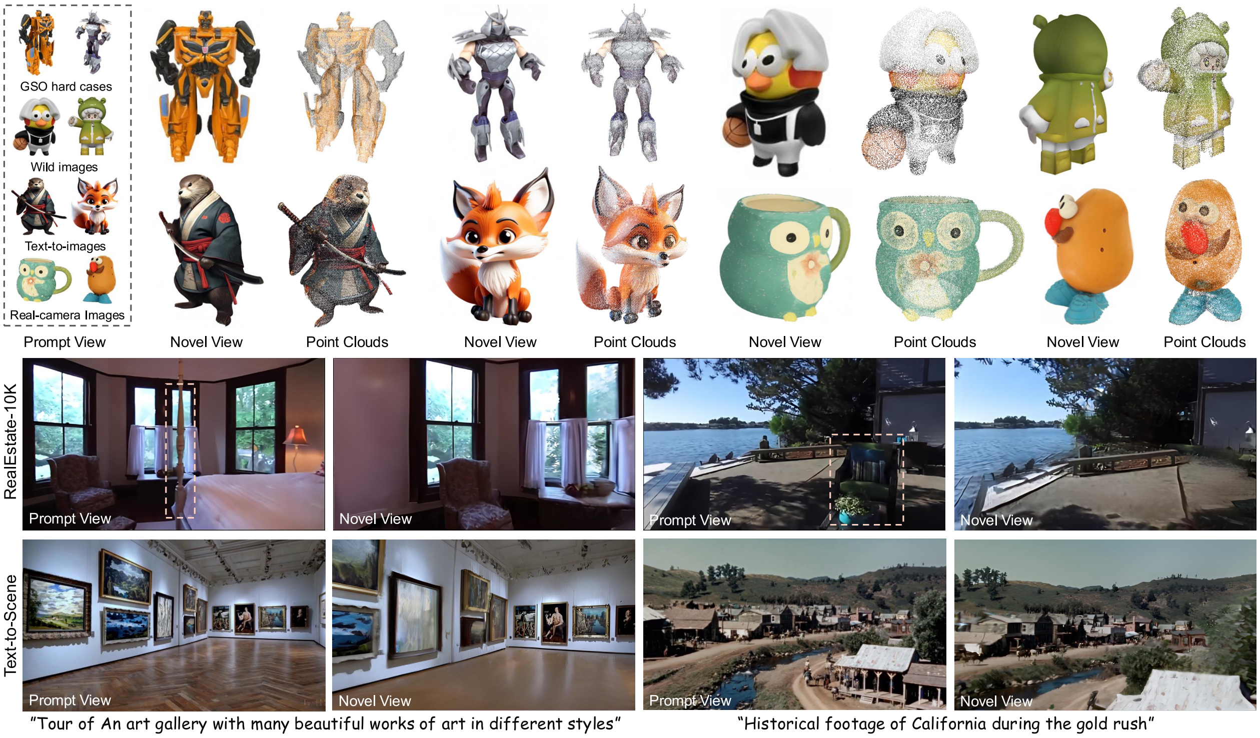


Figure 1. Single-view object generation (upper) and scene reconstruction (lower) results of our method. For single-view object generation, the prompt views are shown in the left dashed box. The generated novel views and 3D Gaussian point clouds are depicted on the right. For single-view scene reconstruction, our model can handle hard cases with occlusion and rotation, as illustrated in the dashed boxes of the third row. The prompt views of object and scene text-to-3D demos are generated by stable diffusion [58] and Sora [4], respectively.

Abstract

Existing feedforward image-to-3D methods mainly rely on 2D multi-view diffusion models that cannot guarantee 3D consistency. These methods easily collapse when changing the prompt view direction and mainly handle object-centric cases. In this paper, we propose a novel single-stage 3D diffusion model, DiffusionGS, for object generation and scene reconstruction from a single view. DiffusionGS directly outputs 3D Gaussian point clouds at each timestep to enforce view consistency and allow the model to generate robustly given prompt views of any directions, beyond object-centric inputs. Plus, to improve the capability and generality of DiffusionGS, we scale up 3D training data by developing a scene-object mixed training strategy. Experiments show that DiffusionGS yields improvements of 2.20 dB/23.25 and 1.34 dB/19.16 in PSNR/FID for objects and scenes than the state-of-the-art methods, without depth estimator. Plus, our method enjoys over 5× faster speed (∼6s on an A100 GPU). Project page: https://caiyuanhao1998.github.io/project/DiffusionGS/

1. Introduction

Image-to-3D task is important and challenging. It aims to reconstruct or generate a 3D representation of scenes or ob-

[*] Work done during the internship in Adobe Research. † Corresponding Authors.

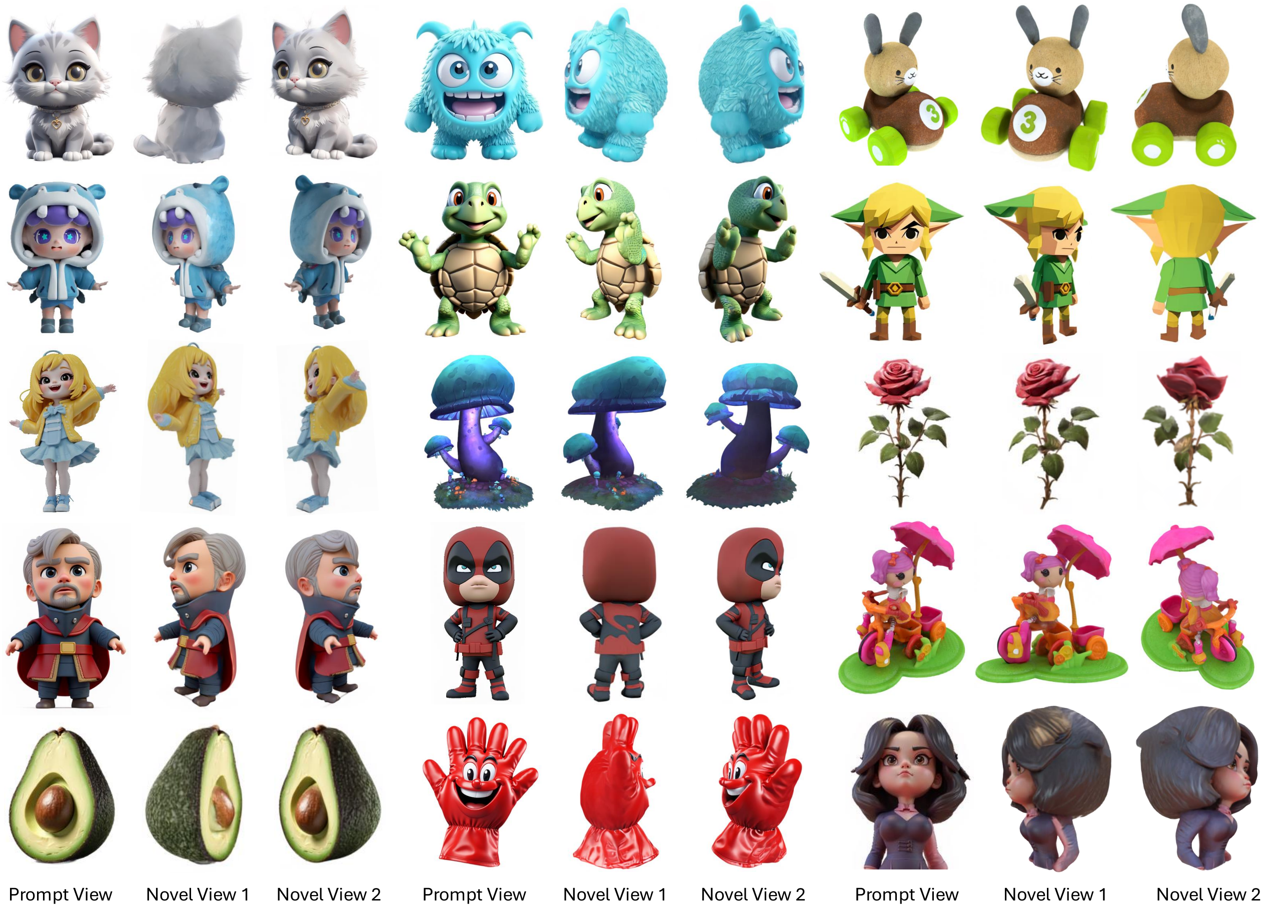


Figure 2. Single-view object generation results of our method on GSO [13], wild images, and text-to-images prompted by stable diffusion or FLUX. Our DiffusionGS can robustly handle hard cases with furry appearance, shadow, flat illustration, complex geometry, and specularity.

jects given only a single-view image. It has wide applications in AR/VR [30], film making [23], robotics [28, 88], animation [46, 56], gaming [40], and so on.

Existing feedforward image-to-3D methods are mainly two-stage [31, 38, 69, 72]. They firstly adopt a 2D diffusion model to generate blocked multi-view images and secondly feed the multi-view images into a 3D reconstruction model. Without 3D model in the diffusion, these methods cannot enforce view consistency and easily collapse when the prompt view direction changes. Another less studied technical route [2, 65, 75] is to train a 3D diffusion model with 2D rendering loss. Yet, these methods mainly rely on triplane neural radiance field (NeRF) [45]. The volume rendering of NeRF is time-consuming and the triplane resolution is limited, preventing the model from scaling up to larger scenes. In addition, current methods mainly study object-level generation using only object-centric datasets to train, which limits the model generalization ability and leaves larger-scale scene-level cases less explored. Although few recent works [67, 70] study single-view scene reconstruction, they rely on monocular depth estimators and easily collapse under severe occlusion or large viewpoint changes.

To address these issues, we propose a novel single-stage 3D Gaussian Splatting (3DGS) [27] based diffusion model, DiffusionGS, for 3D object generation and scene reconstruction from a single view. Our DiffusionGS enforces 3D consistency of the generated contents by predicting multi-view pixel-aligned Gaussian primitives in every timestep. With the highly parallel rasterization and scalable imaging range, DiffusionGS enjoys a fast inference speed of ∼6 seconds and can be easily applied to large scenes. Unlike previous single-view scene reconstruction methods [67, 70, 80] that predict 3D representation only from the given view, DiffusionGS generates other views along the camera trajectory to predict more refined and structured Gaussian point clouds. Thus, our method can better perceive the geometry to reconstruct the scene **without using depth estimator**. As our goal is to build a general and large-scale 3D generation model, it is critical to fully exploit existing 3D scene and object data. Yet, directly training with scene and object data may lead to non-convergence because of the large domain discrepancy. Thus, we propose a scene-object mixed training strategy to handle this problem and learn a general prior of geometry and texture. Our mixed training strategy adapts

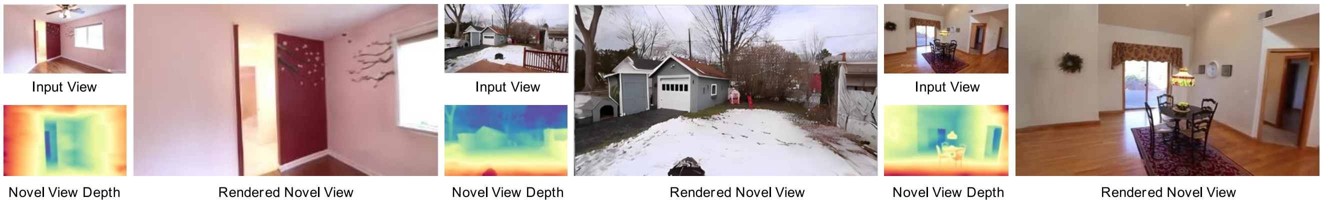


Figure 3. Single-view scene reconstruction of our method on indoor and outdoor scenes. The depth maps are rendered by GS point clouds.

DiffusionGS to both object and scene datasets by controlling the distribution of the selected views, camera condition, Gaussian point clouds, and imaging depths. In particular, we notice previous camera conditioning method Plücker coordinate [54] shows limitations in capturing depth and 3D geometry. Hence, we design a new camera conditioning method, Reference-Point Plücker Coordinates (RPPC), that encodes the point on each ray closest to the origin of the world coordinate system to help DiffusionGS better perceive the depth and 3D geometry across scene and object data. Finally, the model is further finetuned on object or scene data, respectively, to boost the performance.

Our contributions can be summarized as follows:

- We propose a novel framework, DiffusionGS, for 3D object generation and scene reconstruction from single view.
- We design a scene-object mixed training strategy to learn a more general prior from both 3D object and scene data.
- We customize a new camera pose conditioning method, RPPC, to better perceive the relative depth and geometry.
- Our method outperforms SOTA single-view object generation and scene reconstruction methods by 2.20 dB/23.25 and 1.34 dB/19.16 in PSNR/FID score, while enjoying a fast inference speed of ~6s on a single A100 GPU.

2. Related Work

2.1. Diffusion Models for Image-to-3D Generation

Diffusion models [20, 21, 58, 63, 64] are proposed for image generation and recently have been applied to 3D generation, which can be divided into four categories. The first category [19, 25, 50, 51, 60, 85, 89] uses direct supervision on 3D models such as point clouds or meshes, which are hard to obtain in practice. The second kind of methods [57, 68, 71, 78, 79] use SDS loss [55] to distill a 3D model from a 2D diffusion. Yet, these methods require a time-consuming per-asset optimization. The third category [6, 18, 39, 41, 42, 48, 59] adds the camera poses as the input condition to finetune a 2D diffusion model to render fixed novel views. Yet, these methods cannot guarantee 3D consistency and easily collapse when prompt view direction changes. The last category [2, 3, 65, 75] trains a 3D diffusion model with 2D rendering loss. However, these methods mainly based on triplane-NeRF suffer from the limited resolution of triplane and slow speed of volume rendering.

2.2. Gaussian Splatting

3DGS [27] uses millions of Gaussian ellipsoid point clouds to represent objects or scenes and render views with rasterization. It achieves success in 3D/4D reconstruction [5, 10, 15, 16, 32, 44, 73, 77, 81], generation [9, 33, 47, 79, 85, 89], inverse rendering [24, 34, 74], SLAM [26, 76, 84], *etc.* Flash3D [67] and VistaDream [70] use a depth estimator to predict the positions of GS point clouds, but they show limitations in handling occluded scenes. DiffSplat [35] based on T2I prior compresses Gaussian primitives into VAE latent space, constraining its application in larger scenes. We aim to fill these research gaps.

3. Method

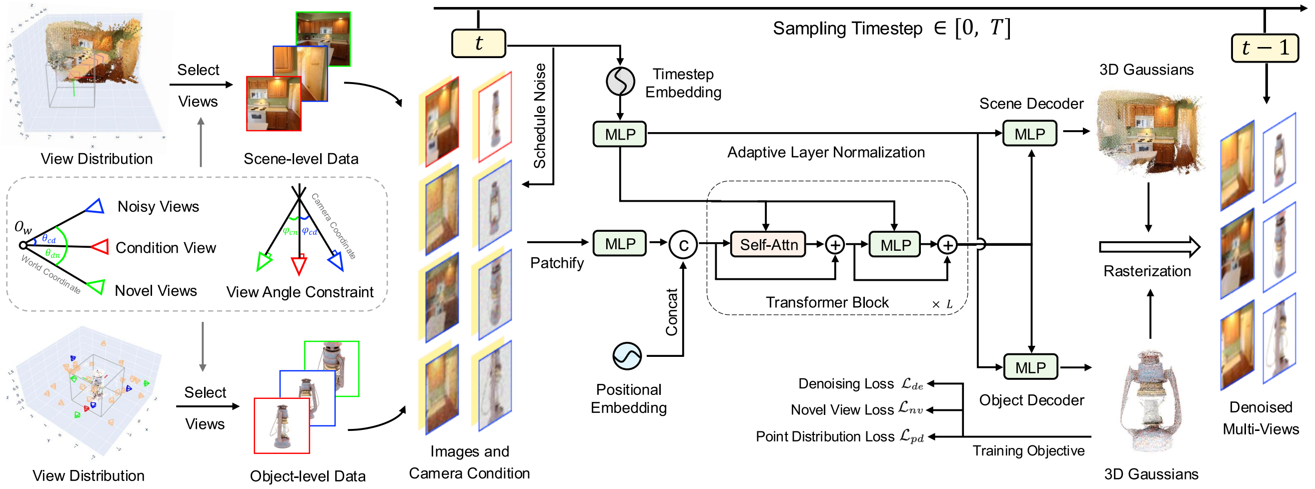
Fig. 4 depicts the pipeline of our method. Fig. 4 (a) shows the scene-object mixed training. For each scene or object, we pick up a view as the condition, N views as the noisy views to be denoised, and M novel views for supervision. Then in Fig. 4 (b), the clean and noisy views are fed into our DiffusionGS to predicts per-pixel 3D Gaussian primitives.

3.1. DiffusionGS

Preliminary of Diffusion. We first review denoising diffusion probabilistic model (DDPM) [20]. In the forward noising process, DDPM transforms the real data distribution $x_0 \sim q(x)$ to standard normal distribution $\mathcal{N}(0, \mathbf{I})$ by gradually applying noise to the real data x_0 : $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$ at every timestep $t \in [0, T]$, where $\bar{\alpha}_t$ are pre-scheduled hyper-parameters. Then x_t is sampled by $x_t = \bar{\alpha}_t x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$. The denoising process reverses the forward process by gradually using a neural network to predict ϵ_t. Similarly, 2D multi-view diffusion [38, 39, 59, 69] generates novel views by denoising images or latents at multiple viewpoints. However, these 2D diffusions do not have 3D models, thus suffering from view misalignment and easily collapsing when the prompt view direction changes. We solve these problems by baking 3D Gaussians into the diffusion denoiser.

Our 3D Diffusion. Different from the normal diffusion model that predicts noise, our DiffusionGS aims to recover clean 3D Gaussian point clouds. Thus, we design the denoiser to directly predict pixel-aligned 3D Gaussians [66] and be supervised at clean 2D multi-view renderings.

As shown in Fig. 4 (b), the input of DiffusionGS in the training phase are one clean condition view $\mathbf{x}_{con} \in$



(a) Scene-Object Mixed Training

(b) Denoiser Architecture of Our DiffusionGS in a Single Timestep

Figure 4. Pipeline. (a) When selecting the data for our scene-object mixed training, we impose two angle constraints on the positions and orientations of viewpoint vectors to guarantee the training convergence. (b) The denoiser architecture of DiffusionGS in a single timestep.

$\mathbb{R}^{H\times W\times 3}$ and N noisy views $\mathcal{X}_t = \{\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \cdots, \mathbf{x}_t^{(N)}\}$ concatenated with viewpoint conditions $\mathbf{v}_{con} \in \mathbb{R}^{H\times W\times 6}$ and $\mathcal{V} = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \cdots, \mathbf{v}^{(N)}\}$. Denote the clean counterparts of the noisy views as $\mathcal{X}_0 = \{\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, \cdots, \mathbf{x}_0^{(N)}\}$. The forward diffusion process adds noise to each view as

$$\mathbf{x}_t^{(i)} = \bar{\alpha}_t \mathbf{x}_0^{(i)} + \sqrt{1-\bar{\alpha}_t}\,\epsilon_t^{(i)}, \tag{1}$$

where $\epsilon_t^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ and $i = 1, 2, \cdots, N$. Then in each timestep t, the denoiser θ predicts the 3D Gaussians $\mathcal{G}_\theta$ to enforce view consistency. As the number of original 3D Gaussians is not a constant, we adopt the pixel-aligned 3D Gaussians [66] as the output, whose number is fixed. The predicted 3D Gaussians $\mathcal{G}_\theta$ is formulated as

$$\mathcal{G}_\theta(\mathcal{X}_t|\mathbf{x}_{con}, \mathbf{v}_{con}, t, \mathcal{V}) = \{G_t^{(k)}(\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)}, \alpha_t^{(k)}, \boldsymbol{c}_t^{(k)})\}, \tag{2}$$

where $1 \leq k \leq N_g$ and $N_g = (N+1)HW$ is the number of per-pixel Gaussian $G_t^{(k)}$. H and W are the height and width of the image. Each $G_t^{(k)}$ contains a center position $\boldsymbol{\mu}_t^{(k)} \in \mathbb{R}^3$, a covariance $\boldsymbol{\Sigma}_t^{(k)} \in \mathbb{R}^{3\times 3}$ controlling its shape, an opacity $\alpha_t^{(k)} \in \mathbb{R}$ characterizing the transmittance, and an RGB color $\boldsymbol{c}_t^{(k)} \in \mathbb{R}^3$. Specifically, $\boldsymbol{\mu}_t^{(k)} = \boldsymbol{o}^{(k)} + u_t^{(k)}\boldsymbol{d}^{(k)}$. $\boldsymbol{o}^{(k)}$ and $\boldsymbol{d}^{(k)}$ are the origin and direction of the k-th pixel-aligned ray. The distance $u_t^{(k)}$ is parameterized by

$$u_t^{(k)} = w_t^{(k)} u_{near} + (1 - w_t^{(k)}) u_{far}, \tag{3}$$

where u_{near} and u_{far} are the nearest and farthest distances. $w_t^{(k)} \in \mathbb{R}$ is the weight to control $u_t^{(k)}$. $\boldsymbol{\Sigma}_t^{(k)}$ is parameterized by a rotation matrix $\mathbf{R}_t^{(k)}$ and a scaling matrix $\mathbf{S}_t^{(k)}$. $w_t^{(k)}, \mathbf{R}_t^{(k)}, \mathbf{S}_t^{(k)}, \alpha_t^{(k)}$, and $\boldsymbol{c}_t^{(k)}$ are directly extracted from the merged per-pixel Gaussian maps by splitting channels.

Denoiser Architecture. As shown in Fig. 4 (b), the input images concatenated with the viewpoint conditions are patchified, linearly projected, and then concatenated with a positional embedding to derive the input tokens of the Transformer backbone, which consists of L blocks. Each block contains a multi-head self-attention (MSA), an MLP, and two layer normalization (LN). Eventually, the output tokens are fed into the Gaussian decoder to be linearly projected and then unpatchified into per-pixel Gaussian maps $\hat{\mathcal{H}} = \{\hat{\mathbf{H}}_{con}, \hat{\mathbf{H}}^{(1)}, \cdots, \hat{\mathbf{H}}^{(N)}\}$, where $\hat{\mathbf{H}}_{con}$ and $\hat{\mathbf{H}}^{(i)} \in \mathbb{R}^{H\times W\times 14}$. Then $N+1$ Gaussian maps are merged into the Gaussian point clouds $\mathcal{G}_\theta$ in Eq. (2). The timestep condition controls the Transformer block and Gaussian decorder through the adaptive layer normalization mechanism [53].

Gaussian Rendering. As the ground truth of $G_t^{(k)}$ is not available, we use the 2D renderings to supervise $\mathcal{G}_\theta$. To this end, we formulate DiffusionGS to a multi-view diffusion model. As aforementioned, 2D diffusion usually predicts the noise ϵ_t. Yet, noisy Gaussian point clouds do not have texture information and may degrade view consistency. To derive clean and complete 3D Gaussians, DiffusionGS is x_0-prediction instead of ϵ-prediction. The denoised multi-view images $\hat{\mathcal{X}}_{(0,t)} = \{\hat{\mathbf{x}}_{(0,t)}^{(1)}, \hat{\mathbf{x}}_{(0,t)}^{(2)}, \cdots, \hat{\mathbf{x}}_{(0,t)}^{(N)}\}$ are rendered by the differentiable rasterization function F_r as

$$\hat{\mathbf{x}}_{(0,t)}^{(i)} = F_r(\mathbf{M}_{ext}^{(i)}, \mathbf{M}_{int}^{(i)}, \mathcal{G}_\theta(\mathcal{X}_t|\mathbf{x}_{con}, \mathbf{v}_{con}, t, \mathcal{V})), \tag{4}$$

where $1 \leq i \leq N$. $\mathbf{M}_{ext}^{(i)}$ and $\mathbf{M}_{int}^{(i)}$ denote the extrinsic matrix and intrinsic matrix of the viewpoint $\mathbf{c}^{(i)}$.

For each $G_t^{(k)}$ at viewpoint $\mathbf{c}^{(i)}$, the rasterization projects its 3D covariance $\boldsymbol{\Sigma}_t^{(k)}$ from the world coordinate system to $\boldsymbol{\Sigma'}_t^{(k,i)} \in \mathbb{R}^{3\times 3}$ in the camera coordinate system as

$$\boldsymbol{\Sigma'}_t^{(k,i)} = \mathbf{J}_t^{(i)} \mathbf{W}_t^{(i)} \boldsymbol{\Sigma}_t^{(k)} {\mathbf{W}_t^{(i)}}^\top {\mathbf{J}_t^{(i)}}^\top, \tag{5}$$

where $\mathbf{J}_t^{(i)} \in \mathbb{R}^{3\times 3}$ is the Jacobian matrix of the affine approximation of the projective transformation. $\mathbf{W}_t^{(i)} \in \mathbb{R}^{3\times 3}$ is the viewing transformation. The 2D projection is divided

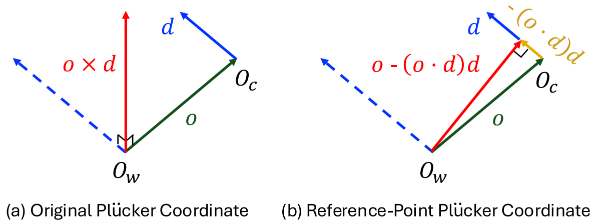


Figure 5. Plücker ray *vs.* Reference-Point Plücker Coordinate.

into non-overlapping tiles. The 3D Gaussians are assigned to the tiles where their 2D projections cover. For each tile, the assigned 3D Gaussians are sorted according to the view space depth. Then the RGB value at pixel (m, n) is derived by blending $\mathcal{N}$ ordered points overlapping the pixel as

$$\hat{\mathbf{x}}^{(i)}_{(0,t)}(m,n) = \sum_{j \in \mathcal{N}} \boldsymbol{c}^{(j)}_t \, \sigma^{(j)}_t \prod_{l=1}^{j-1} (1 - \sigma^{(l)}_t), \tag{6}$$

where $\sigma^{(l)}_t = \alpha^{(l)}_t P(\mathbf{z}^{(l)}_t | \boldsymbol{\mu}^{(l)}_t, \boldsymbol{\Sigma}^{(l)}_t)$, $\mathbf{z}^{(l)}_t$ is the l-th intersection 3D point, and $P(\mathbf{z}^{(l)}_t | \boldsymbol{\mu}^{(l)}_t, \boldsymbol{\Sigma}^{(l)}_t)$ is the possibility of the corresponding 3D Gaussian distribution at $\mathbf{z}^{(l)}_t$.

Then we use the weighted sum, controlled by λ, of $\mathcal{L}_2$ loss and VGG-19 [61] perceptual loss $\mathcal{L}_{\text{VGG}}$ between the multi-view predicted images $\hat{\mathcal{X}}_{(0,t)}$ and ground truth $\mathcal{X}_0$ as the denoising loss $\mathcal{L}_{de}$ to supervise the 3D Gaussians $\mathcal{G}_\theta$ as

$$\mathcal{L}_{de} = \mathcal{L}_2(\hat{\mathcal{X}}_{(0,t)}, \mathcal{X}_0) + \lambda \cdot \mathcal{L}_{\text{VGG}}(\hat{\mathcal{X}}_{(0,t)}, \mathcal{X}_0). \tag{7}$$

In the testing phase, our DiffusionGS randomly samples noise from standard normal distribution at timestep T and then gradually denoise it step by step. The predicted $\hat{\mathcal{X}}_{(0,t)}$ at each timestep t is fed into the next timestep $t-1$ to replace $\mathcal{X}_0$ in Eq. (1) for sampling $\mathcal{X}_{t-1}$ at each noisy view as

$$\mathbf{x}^{(i)}_{t-1} = \bar{\alpha}_{t-1} \hat{\mathbf{x}}^{(i)}_{(0,t)} + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon^{(i)}_{t-1}. \tag{8}$$

Then we use 30-step DDIM [63] to facilitate the sampling speed by skipping some timesteps. One clean image and relative poses are input for inference. Finally, the generated $\mathcal{G}_\theta$ at $t = 0$ in Eq. (2) can be used to render novel views.

3.2. Scene-Object Mixed Training Strategy

Existing 3D training data is relatively scarce and the cost of data annotation is expensive. Especially for the scene-level datasets, there are only ~90K training samples [36, 90] and most of them only cover small viewpoint changes, which are not sufficient to learn strong geometry representations. Besides, the majority of object-level training data is synthetic [12]. As a result, models [25, 31, 50, 69, 75] trained on object-level data often generate unrealistic textures, limiting the practice on real-camera images. To improve the capacity and generalization ability of our DiffusionGS, it is critical to make full use of both object and scene data.

Yet, directly training 3D diffusion models with both object and scene datasets may introduce artifacts or lead to non-convergence because of the large domain discrepancy. As shown in the lower part of Fig. 4 (a), object-level datasets [12] usually have an object in the central position without background and the camera rotates around this central object to capture multi-view images. The imaging range and depth are limited. In contrast, as depicted in the upper part of Fig. 4 (a), scene-level datasets have more dense image representations instead of blank background. The imaging range and depth are much wider. The distribution of viewpoints is usually a trajectory of continuous motion, such as dolly in and out [90] or panning left and right [36].

To handle these issues, we design a mixed training strategy that controls the distribution of selected views, camera condition, Gaussian point clouds, and imaging depth.

Viewpoint Selecting. The first step of our mixed training is to select viewpoints. For better convergence of training process, we impose two angle constraints on camera positions and orientations to ensure the noisy views and novel views have certain overlaps with the condition view.

The first constraint is on the angle between viewpoint positions. After normalization, this angle measures the distance of viewpoints. As the noisy views can only provide partial information, we control the angle $\theta^{(i)}_{cd}$ between the i-th noisy view position and the condition view position, and the angle $\theta^{(i,j)}_{dn}$ between the i-th noisy view position and the j-th novel view position. Then the constraints are

$$\theta^{(i)}_{cd} \leq \theta_1, \;\; \theta^{(i,j)}_{dn} \leq \theta_2, \tag{9}$$

where θ_1 and θ_2 are hyperparameters, $1 \leq i \leq N$, and $1 \leq j \leq M$. The position vector can be read from the translation of camera-to-world (c2w) matrix of the viewpoint.

The second constraint is on the angle between viewpoint orientations. This angle also controls the overlap of different viewpoints. Denote the forward direction vectors of the condition view, the i-th noisy view, and the j-th novel view as $\vec{z}_{con}$, $\vec{z}^{\,(i)}_{noise}$, and $\vec{z}^{\,(j)}_{nv}$. Then the constraints are

$$\frac{\vec{z}_{con} \cdot \vec{z}^{\,(i)}_{noise}}{|\vec{z}_{con}| \cdot |\vec{z}^{\,(i)}_{noise}|} \geq \cos(\varphi_1), \quad \frac{\vec{z}_{con} \cdot \vec{z}^{\,(j)}_{nv}}{|\vec{z}_{con}| \cdot |\vec{z}^{\,(j)}_{nv}|} \geq \cos(\varphi_2), \tag{10}$$

Where φ_1 and φ_2 are hyperparameters. $\vec{z}$ is read from c2w.

Reference-Point Plücker Coordinate. To offer the camera conditions, previous methods [7, 17, 62, 69, 75] adopt a pixel-aligned ray embedding, plücker coordinates [54], concatenated with the image as input. As shown in Fig. 5 (a), the pixel-aligned ray embeddings are parameterized as $\boldsymbol{r} = (\boldsymbol{o} \times \boldsymbol{d}, \boldsymbol{d})$, where $\boldsymbol{o}$ and $\boldsymbol{d}$ are the position and direction of the ray landing on the pixel. Specifically, $\boldsymbol{o} \times \boldsymbol{d}$ represents the rotational effect of $\boldsymbol{o}$ relative to $\boldsymbol{d}$, showing limitations in perceiving the relative depth and geometry.

To handle this problem, we customize a Reference-Point Plücker Coordinate (RPPC) as the camera condition. As depicted in Fig. 5 (b), we use the point on the ray closest to the origin of the world coordinate system as the reference point to replace the moment vector, which can be formulated as

$$\boldsymbol{r} = (\boldsymbol{o} - (\boldsymbol{o} \cdot \boldsymbol{d})\boldsymbol{d}, \boldsymbol{d}) \tag{11}$$

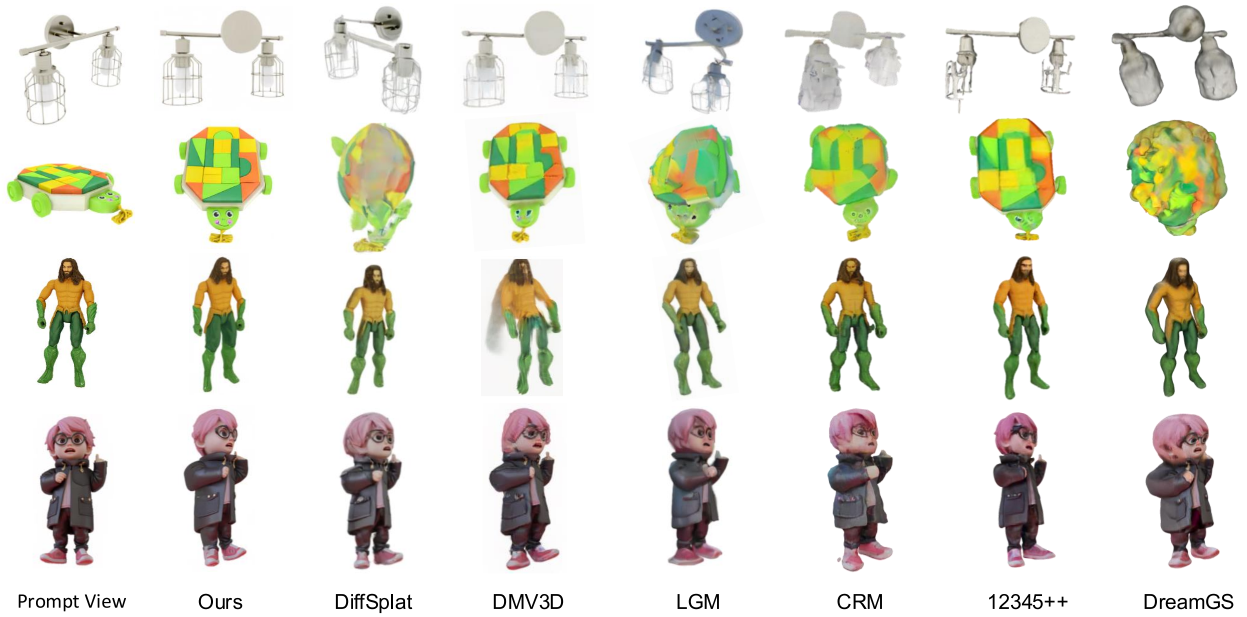


Figure 6. Visual comparison of single-view object generation on ABO, GSO, real-camera image, and text-to-image prompted by FLUX. Our method can generate more fine-grained details with accurate geometry. DiffSplat is based on SD3.5m [14] for its best performance.

Our RPPC satisfies the translation invariance assumption of the 4D light field [1]. Plus, compared to the moment vector, our reference point can provide more information about the ray position and the relative depth, which are beneficial for the diffusion model to capture the 3D geometry of scenes and objects. By skip connections, the reference-point information can flow through every Transformer block and the Gaussian decoder to guide the GS point cloud generation.

Dual Gaussian Decoder. As the depth range varies across object- and scene-level datasets, we use two MLPs to decode the Gaussian primitives for objects and scenes in mixed training. As shown in Fig. 4 (b), for the object-level Gaussian decoder, the nearest and farthest distances [u_{near}, u_{far}] in Eq. (3) are set as [0.1, 4.2] and $\boldsymbol{\mu}_t^{(k)}$ is clipped into $[-1,1]^3$. For the scene-level Gaussian decoder, [u_{near}, u_{far}] is set to [0, 500]. The two decoders are also controlled by the timestep embedding. In the finetuning phase, we just use a single decoder while the other is removed.

Overall Training Objective. Similar to the denoising loss $\mathcal{L}_{de}$ in Eq. (7), we compute $\mathcal{L}_2$ loss and perceptual loss with the same balancing hyperparameter λ on novel views. The novel view loss is denoted as $\mathcal{L}_{nv}$. To encourage the distribution of 3D Gaussian point clouds of object-centric generation more concentrated, we design a point distribution loss $\mathcal{L}_{pd}$ for training warm-up. $\mathcal{L}_{pd}$ is formulated as

$$\mathcal{L}_{pd} = \mathbb{E}_k[l_t^{(k)} - (\frac{l_t^{(k)} - \mathbb{E}_k[l_t^{(k)}]}{\sqrt{\mathrm{Var}(l_t^{(k)})}}\sigma_0 + \mathbb{E}_k[|\boldsymbol{o}^{(k)}|])], \qquad (12)$$

where $\mathbb{E}$ represents the mean value, $l_t^{(k)} = |u_t^{(k)}\boldsymbol{d}^k|$, Var denotes the variance, and σ_0 is the target standard deviation. σ_0 is set to 0.5. Then the overall training objective $\mathcal{L}$ is

$$\mathcal{L} = (\mathcal{L}_{de} + \mathcal{L}_{nv}) \cdot \mathbf{1}_{\text{iter}>\text{iter}_0} + \mathcal{L}_{pd} \cdot \mathbf{1}_{\text{iter}\leq\text{iter}_0} \cdot \mathbf{1}_{\text{object}}, \qquad (13)$$

where $\mathbf{1}_{\text{iter}>\text{iter}_0}$ is a conditional indicator function which equals 1 if the current training iteration (iter) is greater than the threshold (iter$_0$). $\mathbf{1}_{\text{iter}\leq\text{iter}_0}$ and $\mathbf{1}_{\text{object}}$ are similar.

4. Experiment

Dataset. We use Objaverse [12] and MVImgNet [83] as the training sets for objects. We center and scale each 3D object of Objaverse into $[-1,1]^3$, and render 32 images at random viewpoints with 50° FOV. For MVImgNet, we crop the object, remove the background, normalize the cameras, and center and scale the object to $[-1,1]^3$. We preprocess 730K and 220K training samples in Objaverse and MVImgNet. We use the ABO [11] and GSO [13] datasets for evaluation. We adopt RealEstate10K [90] and DL3DV10K [36] as the scene-level training datasets. RealEstate10K includes 80K video clips of indoor and outdoor real scenes selected from YouTube videos. We follow the standard training/testing split. DL3DV10K contains 10510 videos of real-world scenarios, covering 96 complex categories. For all evaluation, each instance has 1 input view and 10 testing views.

Implementation Details. We implement DiffusionGS by Pytorch [52] and train it with Adam optimizer [29]. To save GPU memory, we adopt mixed-precision training [49] with BF16, sublinear memory training [8], and deferred GS rendering [86]. In mixed training, we use 32 A100 GPUs to train the model on Objaverse, MVImgNet, RealEstate10K, and DL3DV10K for 40K iterations at the per-GPU batch size of 16. Then we finetune the model on the object- and scene-level datasets with 64 A100 GPUs for 80K and 54K iterations at the per-GPU batch size of 8 and 16. The learning rate is linearly warmed up to 4e^{-4} with 2K iterations

Method	DreamGS [68]	LGM [69]	DMV3D [75]	CRM [72]	12345++ [38]	DiffSplat [35]	Ours
User Study Score ↑	2.93	3.41	4.07	3.19	4.56	3.96	**5.89**
Runing Time (s) ↓	120	**4.1**	31.4	10	60	4.3	5.8

(a) User preference and running time comparison on object generation

Method	Infer Time	Post-hoc GS Time	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
PhotoNVS [82]]	61s	2417s	15.31	0.5215	0.4589	28.30
Our DiffusionGS	**6s**	**0s**	**21.63**	**0.6787**	**0.2743**	**15.87**

(b) Comparison with the SOTA 2D method PhotoNVS on [90]

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
LGM [69]	16.01	0.7262	0.3255	86.32
GS-LRM [87]	18.78	0.7974	0.2720	123.55
DMV3D [75]	23.69	0.8634	0.1131	32.28
DiffusionGS	**25.89**	**0.8880**	**0.0965**	**9.03**

(c) Object generation results on ABO [11]

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
LGM [69]	14.27	0.7183	0.3003	75.55
GS-LRM [87]	17.70	0.7950	0.2411	112.96
DMV3D [75]	20.82	0.8347	0.1289	33.48
DiffusionGS	**22.07**	**0.8545**	**0.1115**	**11.52**

(d) Object generation results on GSO [13]

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
PixelNeRF [80]	17.46	0.5713	0.5525	159.52
Splatter-Image [66]	18.21	0.6115	0.4839	120.35
Flash3D [87]	20.29	0.6483	0.3610	35.03
DiffusionGS	**21.63**	**0.6787**	**0.2743**	**15.87**

(e) Scene reconstruction on Realestate10K [90]

Table 1. User study and main quantitative results of single-view image-to-3D task on ABO [11], GSO [13], and Realestate10K [90].

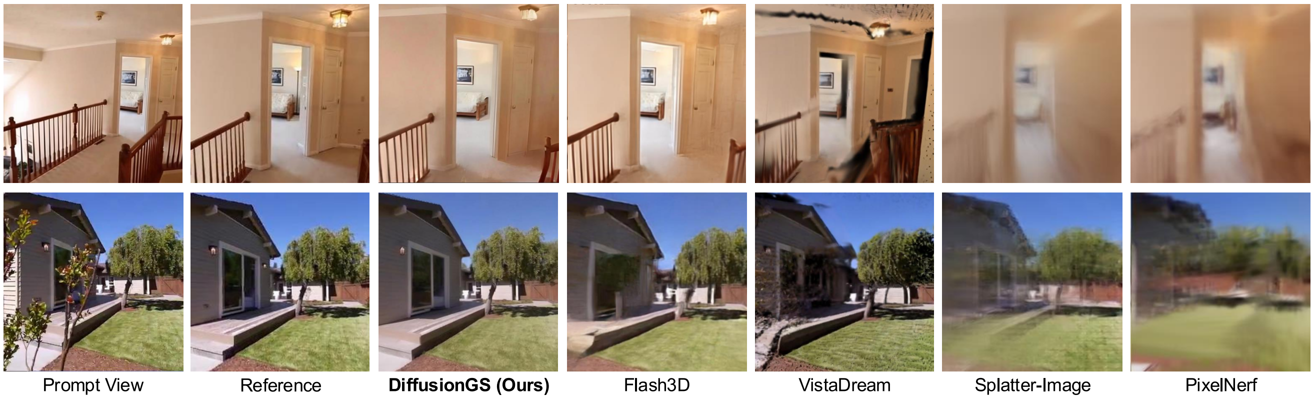


Figure 7. Visual results of single-view scene reconstruction. We train the feedforward methods with the same scene data for fairness. Previous methods yield blurry images or introduce artifacts. In contrast, our method can robustly reconstruct scenes with occlusion.

and decays to 0 using cosine annealing scheme [43]. Finally, we scale up the training resolution from 256×256 to 512×512 and finetune the model for 20K iterations.

4.1. Comparison with State-of-the-art Methods

Single-view Object Generation. Fig. 6 compares object generation on ABO, GSO, real-camera image [37], and text-to-image prompted by FLUX. We compare our method with six state-of-the-art (SOTA) methods including two 3D diffusion (DiffSplat [35] and DMV3D [75]), three 2D multi-view diffusion-based methods (LGM [69], CRM [72], and 12345++ [38]), and an SDS-based method DreamGS [68]. Previous methods render over-smoothed images or distort 3D geometry. In contrast, our method robustly generates clearer novel views and perfect 3D geometry with prompt views of any directions, while preserving fine-grained details. Even when the front view, which previous methods specialize in, is given (third and fourth row), our method still yields better view consistency by retaining the face details of the dolls. While the methods based on 2D multi-view diffusion introduce cracks, artifacts, and blur to the faces when "stitching" unaligned multi-view images.

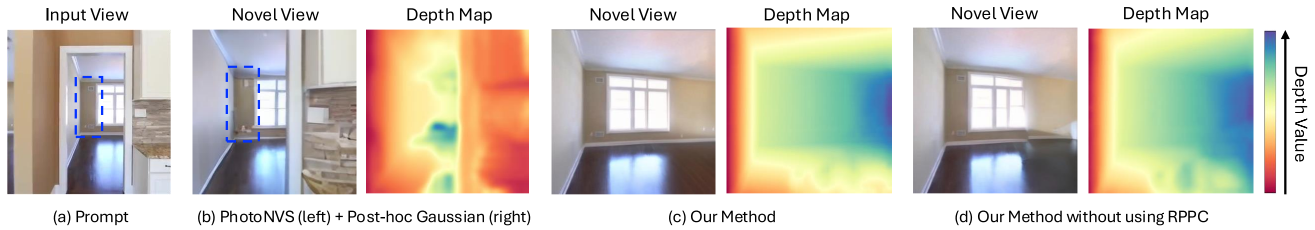
We conduct a user study by inviting 27 people to score the visual quality of the generation results of 14 objects according to the 3D geometry, texture quality, and alignment with the prompt view. The user study score ranges from 1 (worst) to 7 (best). Tab. 1a reports the results and running time at the size of 256×256. Our method achieves the highest score while enjoying over 5× and 10× inference speed compared to the SOTA 3D diffusion DMV3D and multi-view diffusion-based method 12345++. Tab. 1c and 1d list the results of object generation on the ABO and GSO datasets. DiffusionGS surpasses DMV3D by 2.2/1.25 dB in PSNR and 23.25/21.96 in FID score on ABO/GSO.

Method	Baseline	+ Our Diffusion	+ $\mathcal{L}_{pd}$	+ Mixed Training	+ RPPC
PSNR ↑	17.63	20.57	20.94	21.73	22.07
SSIM ↑	0.7928	0.8120	0.8423	0.8515	0.8545
LPIPS ↓	0.2452	0.1417	0.1218	0.1196	0.1115
FID ↓	118.31	47.86	28.41	17.79	11.52

Table 2. Ablation study. Results on the GSO [13] dataset are listed.

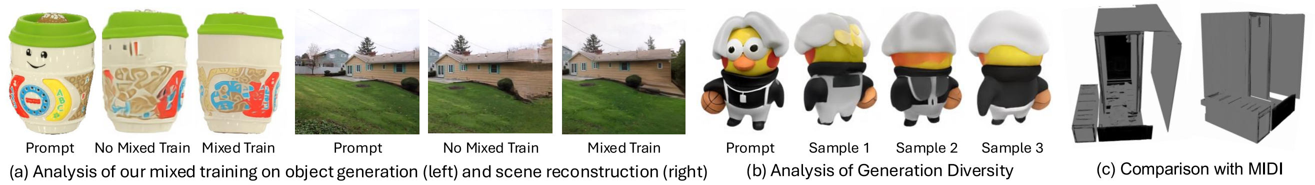
We chain stable diffusion [58] or FLUX with DiffusionGS to perform text-to-3D in Fig. 1 and 2, our method can handle hard cases with furry appearance, shadow, flat illustration, complex geometry, and even specularity.

Single-view Scene Reconstruction. We compare our method with three feedforwad methods (Flash3D [67], Splatter-Image [66], and pixelNeRF [80]) and one SDS-based method VistaDream [70]. For fair comparison, we train the feedforward methods with the same scene data as DiffusionGS. Tab. 1e reports the results on RealEstate10K. Our method outperforms the SOTA method Flash3D [87] by 1.34 dB in PSNR and 19.16 in FID score. Fig. 1, 3, and 7 depict the visual results of indoor and outdoor scene reconstruction. In Fig. 7, pixelNeRF and Splatter-Image render blurry images. Although using monocular depth estimator, Flash3D and VistaDream still produce blur, artifacts, noise,



(a) Prompt (b) PhotoNVS (left) + Post-hoc Gaussian (right) (c) Our Method (d) Our Method without using RPPC

Figure 8. Visual comparison between the SOTA 2D method PhotoNVS [82] in (b) and our method in (c) on NVS and relative depth estimation. The depth map in (b) is predicted by the post-hoc 3DGS fitting the synthesized views. (d) shows the effect of our RPPC.



(a) Analysis of our mixed training on object generation (left) and scene reconstruction (right) (b) Analysis of Generation Diversity (c) Comparison with MIDI

Figure 9. Visual analysis. (a) studies the effect of mixed training. (b) shows generation diversity. (c) shows the comparison with MIDI [22].

and black spots in the occluded region of novel views. In contrast, as our DiffusionGS can generate the views along the camera trajectory to predict more refined and structured Gaussian point clouds, it can reconstruct clearer details of scenes with occlusion, **without using depth estimator**.

We also compare our method and the SOTA 2D view synthesis method PhotoNVS [82] with post-hoc Gaussian fitting and the SOTA scene mesh reconstruction method MIDI [22]. In Tab. 1b, our method outperforms PhotoNVS by 6.32 dB in PSNR and 12.43 in FID. Fig. 8 (b) and (c) show the rendered views and depth maps. Due to the lack of 3D models, PhotoNVS shows limitations in perceiving 3D structure. The generated images contain many view-inconsistent contents, as depicted in the blue boxes of Fig. 8 (a) and (b). Thus, using post-hoc 3D Gaussians to fit the generated views takes a long time ($\sim$ 40 minutes), suffers from the over-fitting issue, and fails to predict the depth. In Fig. 9 (c), MIDI [22] also fails to reconstruct the 3D geometry of the room. **In contrast**, as shown in Fig. 8 (c), our method can generate more view-consistent images and predict more accurate 3D structure in only 6 seconds.

We chain Sora [4] with our method to perform text-to-scene in Fig. 1. Our method can reliably render novel views for both indoor and outdoor scenes prompted by Sora.

4.2. Ablation Study

Break-down Ablation. To study the effect of each component towards higher performance, we adopt the denoiser without timestep control as the baseline to conduct a break-down ablation. We train it on the object-level datasets with a single-view input and the same amount of supervised views as DiffusionGS. Results on GSO [13] are reported in Tab. 2. The baseline yields poor results of 17.63 dB in PSNR and 118.31 in FID. When applying our diffusion framework, loss $\mathcal{L}_{pd}$ in Eq.(12), scene-object mixed training without RPPC, and RPPC, the model gains by 2.94, 0.37, 0.79, 0.34 dB in PSNR and drops by 70.45, 19.45, 10.62, 6.27 in FID.

Analysis of Mixed Training. We conduct an analysis of our scene-object mixed training in Fig. 9 (a). For fair comparison, models are trained with the same iterations whether with or without mixed training. **(i)** The left part shows the effect on object generation. After using the mixed training, the textures of the cup become clearer and more realistic, and the artifacts on the back are reduced. **(ii)** The right part depicts the effect on scene reconstruction. When applying our mixed training, DiffusionGS can better capture the 3D geometry and reconstruct the house with less artifacts. The improvement of our mixed training for scene reconstruction on Realestate10K is 0.61 dB in PSNR and 10.53 in FID.

Analysis of RPPC. **(i)** Using RPPC leads to a quantitative improvement of 0.28 dB in PSNR and 7.09 in FID for scene reconstruction on Realestate10K. **(ii)** Fig. 8 (c) and (d) analyze the effect of RPPC. Using RPPC can render more accurate scene structures (window, floor, *etc.*) and depth.

Analysis of Generation Diversity. We change the random seed with the same prompt view in Fig. 9 (b). Our method can generate different shapes and textures for 3D assets.

5. Conclusion

In this paper, we propose a novel framework, DiffusionGS, for 3D object generation and scene reconstruction from a single view. Our DiffusionGS directly outputs 3D Gaussian point clouds at each timestep to enforce view consistency and only requires 2D renderings for supervision. In addition, we develop a scene-object mixed training strategy with a new camera conditioning method RPPC to learn a general prior capturing better 3D geometry and texture representations. Experiments show that our method outperforms SOTA methods while enjoying a faster speed of 6 seconds.

Acknowledgements: This work was supported by the office of Naval Research with award N000142412696.

References

[1] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*. The MIT press, 1991. 6

[2] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *CVPR*, 2023. 2, 3

[3] Titas Anciukevicius, Fabian Manhardt, Federico Tombari, and Paul Henderson. Denoising diffusion via image-based rendering. In *ICLR*, 2024. 3

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 8

[5] Yuanhao Cai, Yixun Liang, Jiahao Wang, Angtian Wang, Yulun Zhang, Xiaokang Yang, Zongwei Zhou, and Alan Yuille. Radiative gaussian splatting for efficient x-ray novel view synthesis. In *ECCV*, 2024. 3

[6] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *ICCV*, 2023. 3

[7] Eric Ming Chen, Sidhanth Holalkere, Ruyu Yan, Kai Zhang, and Abe Davis. Ray conditioning: Trading photo-realism for photo-consistency in multi-view image generation. In *ICCV*, 2023. 5

[8] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 6

[9] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, 2024. 3

[10] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2025. 3

[11] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 6, 7

[12] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 5, 6

[13] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022. 2, 6, 7, 8

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 6

[15] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *NeurIPS*, 2024. 3

[16] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. In *NeurIPS*, 2024. 3

[17] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv*, 2024. 5

[18] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *ICML*, 2023. 3

[19] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 3

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[21] Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin C.K. Chan, and Ziwei Liu. ReVersion: Diffusion-based relation inversion from images. In *SIGGRAPH Asia*, 2024. 3

[22] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *CVPR*, 2025. 8

[23] Xuekun Jiang, Anyi Rao, Jingbo Wang, Dahua Lin, and Bo Dai. Cinematic behavior transfer via nerf-based differentiable filming. In *CVPR*, 2024. 2

[24] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023. 3

[25] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3, 5

[26] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. *arXiv preprint arXiv:2312.02126*, 2023. 3

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 2, 3

[28] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects. In *CoRL*, 2022. 2

[29] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[30] Chaojian Li, Sixu Li, Yang Zhao, Wenbo Zhu, and Yingyan Lin. Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2022. 2

[31] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *ICLR*, 2024. 2, 5

[32] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *ECCV*, 2024. 3

[33] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. *arXiv preprint arXiv:2311.11284*, 2023. 3

[34] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473*, 2023. 3

[35] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable 3d gaussian splat generation. In *ICLR*, 2025. 3, 7

[36] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 5, 6

[37] Isabella Liu, Linghao Chen, Ziyang Fu, Liwen Wu, Haian Jin, Zhong Li, Chin Ming Ryan Wong, Yi Xu, Ravi Ramamoorthi, Zexiang Xu, and Hao Su. Openillumination: A multi-illumination dataset for inverse rendering evaluation on real objects. In *NeurIPS*, 2023. 7

[38] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *CVPR*, 2024. 2, 3, 7

[39] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 3

[40] Yang Liu, Xiang Huang, Minghan Qin, Qinwei Lin, and Haoqian Wang. Animatable 3d gaussian: Fast and high-quality reconstruction of multiple human avatars. In *ACM MM*, 2023. 2

[41] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *CVPR*, 2024. 3

[42] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, 2024. 3

[43] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 7

[44] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 3

[45] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[46] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *CVPR*, 2024. 2

[47] Yuxuan Mu, Xinxin Zuo, Chuan Guo, Yilin Wang, Juwei Lu, Xiaofeng Wu, Songcen Xu, Peng Dai, Youliang Yan, and Li Cheng. Gsd: View-guided gaussian splatting diffusion for 3d reconstruction. In *ECCV*, 2024. 3

[48] Norman Müller, Katja Schwarz, Barbara Rössle, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Multidiff: Consistent novel view synthesis from a single image. In *CVPR*, 2024. 3

[49] Sharan Narang, Gregory Diamos, Erich Elsen, Paulius Micikevicius, Jonah Alben, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *ICLR*, 2018. 6

[50] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 3, 5

[51] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc V Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. In *NeurIPS*, 2023. 3

[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[53] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 4

[54] Julius Plücker. *Neue Geometrie des Raumes gegrundet auf die Betrachtung der geraden Linie als Raumelement von Julius Pluecker*. Teubner, 1869. 3, 5

[55] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 3

[56] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *CVPR*, 2024. 2

[57] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. In *ICCV*, 2023. 3

[58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 3, 7

[59] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3

[60] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023. 3

[61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[62] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 5

[63] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3, 5

[64] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[65] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *ICCV*, 2023. 2, 3

[66] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 3, 4, 7

[67] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In *3DV*, 2025. 2, 3, 7

[68] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3, 7

[69] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, 2024. 2, 3, 5, 7

[70] Haiping Wang, Yuan Liu, Ziwei Liu, Wenping Wang, Zhen Dong, and Bisheng Yang. Vistadream: Sampling multiview consistent images for single-view scene reconstruction. *arXiv preprint arXiv:2410.16892*, 2024. 2, 3, 7

[71] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2024. 3

[72] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *ECCV*, 2024. 2, 7

[73] 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. *arXiv preprint arXiv:2310.08528*, 2023. 3

[74] Tianyi Xie, Zeshun Zong, Yuxin Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023. 3

[75] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *ICLR*, 2024. 2, 3, 5, 7

[76] Chi Yan, Delin Qu, Dong Wang, Dan Xu, Zhigang Wang, Bin Zhao, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. *arXiv preprint arXiv:2311.11700*, 2023. 3

[77] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 3

[78] Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. Dreamreward: Text-to-3d generation with human preference. In *ECCV*, 2024. 3

[79] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*, 2023. 3

[80] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 7

[81] Heng Yu, Joel Julin, Zoltán A. Milacski, Koichiro Niinuma, and László A. Jeni. Cogs: Controllable gaussian splatting. In *CVPR*, 2024. 3

[82] Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *ICCV*, 2023. 7, 8

[83] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *CVPR*, 2023. 6

[84] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 3

[85] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. In *NeurIPS*, 2024. 3

[86] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *ECCV*, 2022. 6

[87] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024. 7

[88] Allan Zhou, Moo Jin Kim, Lirui Wang, Pete Florence, and Chelsea Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *CVPR*, 2023. 2

[89] Junsheng Zhou, Weiqi Zhang, and Yu-Shen Liu. Diffgs: Functional gaussian splatting diffusion. In *NeurIPS*, 2024. 3

[90] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 5, 6, 7