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

Yuanhao Cai, He Zhang, Kai Zhang, Yixun Liang, Mengwei Ren, Fujun Luan, Qing Liu, Soo Ye Kim, Jianming Zhang, Zhifei Zhang, Yuqian Zhou, Yulun Zhang, Xiaokang Yang, Zhe Lin , Alan Yuille

Johns Hopkins University, Adobe Research, HKUST, Shanghai Jiao Tong University

- Introduction
- Method
- Experiment

- Introduction
- Method
- Experiment

Existing image-to-3D object-level generation methods are mainly two-stage. First generate blocked multi-view images and then lift to 3D, which easily leads to view-inconsistency.

Current single-view scene-level reconstruction algorithms usually rely on depth estimator and easily collapse when the viewpoint changes significantly.

To cope with these problems, we propose a 3D Gaussian Splatting-based diffusion model, DiffusionGS

prompt view

prompt view

LGM

VistaDream

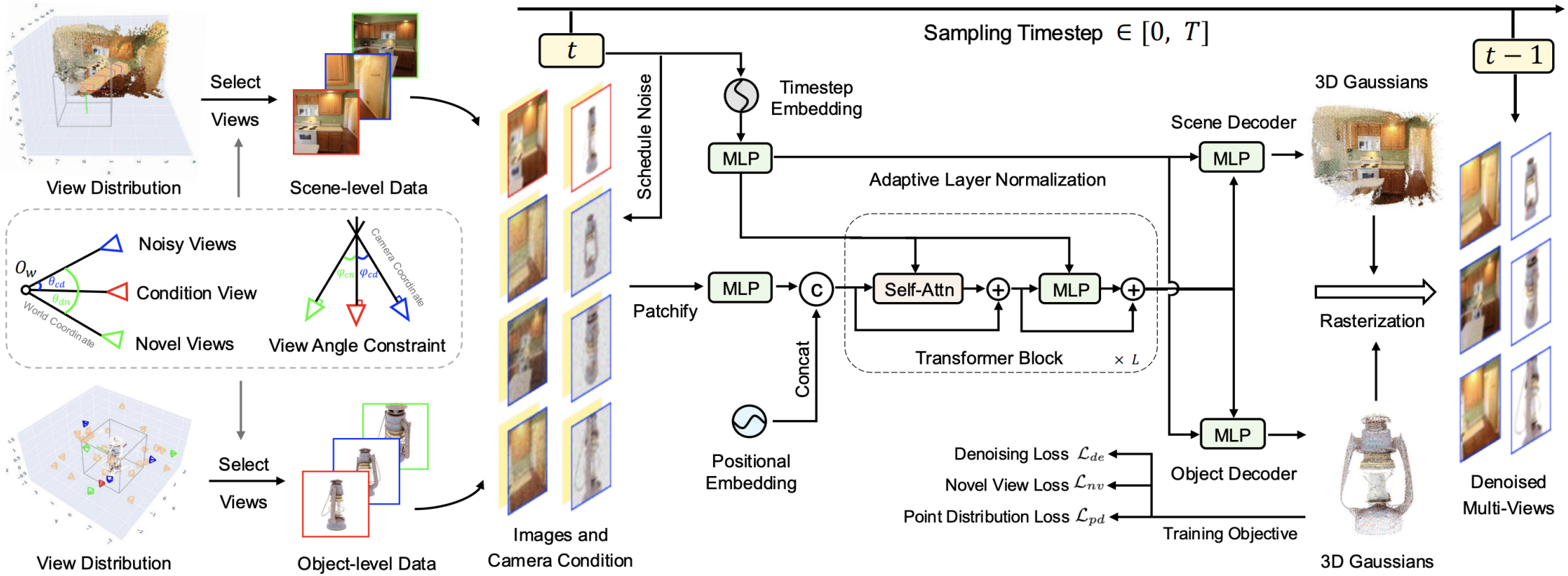
12345++

Splatter-Image

- Introduction

- Method

- Experiment



(a) Scene-Object Mixed Training

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

- Our DiffusionGS predicts 3D Gaussian point clouds at each timestep and renders x0 at each view to proceed the multi-view denoising process.

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

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

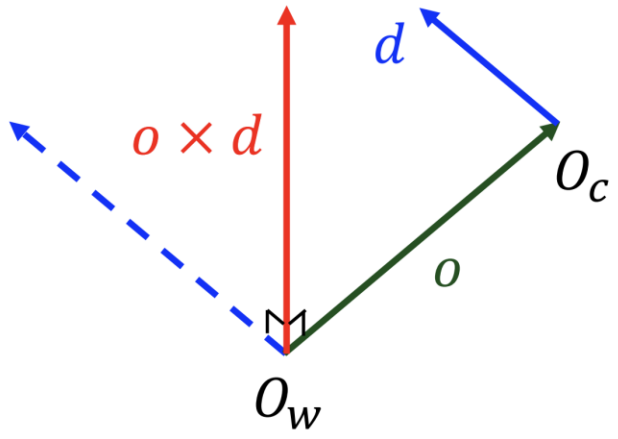
- We also design a scene-object mixed training strategy with two angle constraints

$$\theta_{cd}^{(i)} \leq \theta_1, \quad \theta_{dn}^{(i,j)} \leq \theta_2$$

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

- To better perceive the relative depth and geometry, we customize a Reference-Point Plucker Coordinate (RPPC)

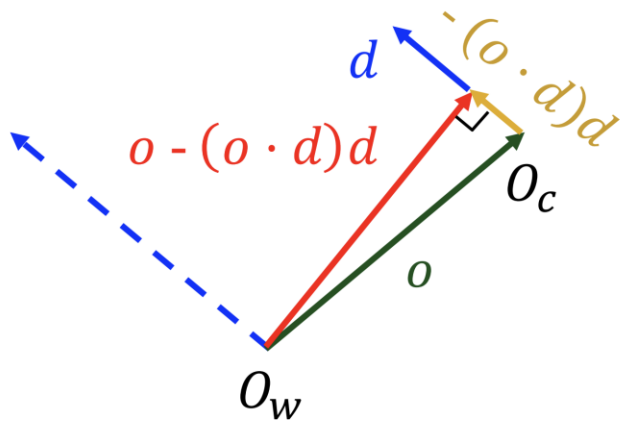
(a) $\boldsymbol{r} = (\boldsymbol{o} \times \boldsymbol{d}, \boldsymbol{d})$ (b) $\boldsymbol{r} = (\boldsymbol{o} - (\boldsymbol{o} \cdot \boldsymbol{d})\boldsymbol{d}, \boldsymbol{d})$



(a) Original Plücker Coordinate

- During the mixed training, we also impose a point distribution loss as the warm up for object-level generation:

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



(b) Reference-Point Plücker Coordinate

- Eventually, the overall training objective is

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

- Introduction
- Method
- Experiment

JOHNS HOPKINS UNIVERSITY

ICCV OCT 19-23, 2025 HONOLULU HAWAII

Method	DreamGaussian [67]	LGM [68]	DMV3D [74]	CRM [71]	12345++ [36]	DiffusionGS (Ours)
User Study Score ↑	1.94	3.04	3.16	2.69	3.81	**4.88**
Runing Time (s) ↓	120	**4.1**	31.4	10	60.0	5.8

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

Method	Infer Time	Post-hoc GS Time	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
PhotoNVS [81]]	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 [68]	16.01	0.7262	0.3255	86.32
GS-LRM [87]	18.78	0.7974	0.2720	123.55
DMV3D [74]	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 [68]	14.27	0.7183	0.3003	75.55
GS-LRM [87]	17.70	0.7950	0.2411	112.96
DMV3D [74]	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 [79]	17.46	0.5713	0.5525	159.52
Splatter-Image [65]	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]

Please visit our project page for more video and interactive results

Project Page

Code & Model

Code & Model : https://github.com/caiyuanhao1998/Open-DiffusionGS

Project page: https://caiyuanhao1998.github.io/project/DiffusionGS/