ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10887749

Single-View Reconstruction via Decoupled 3D Gaussian Splatting

Sheng Liu†, Shiming Zhu†, Huilong Pi⋆, Yunchuan Qin, Zhuo Tang and Ruihui Li⋆
College of Computer Science and Electronic Engineering
Hunan University, Changsha, China
Email: {liusheng23, zshiming, phl880217, qinyunchuan, ztang, liruihui}@hnu.edu.cn

***Abstract*—Creating high-quality 3D object representations from a single-view image is challenging. Existing methods tend to infer the geometry and texture information simultaneously within a shared network. However, decoding geometry and texture from a unified network often leads to their entanglement, causing geometric structure collapse or floating artifacts. After revisiting this task, we propose a single-view reconstruction framework based on 3D Gaussian Splatting. The key idea is to decouple Gaussian position attribute generation from texture feature generation. Technically, our framework combines a Geometry Generator, a Texture Generator, and a Gaussian Attributes Decoder. Two parallel branches, Geometry Generator and Texture Generator, aim for point cloud prediction and texture optimization, respectively. Then the Gaussian Attributes Decoder integrates the generated position and texture attributes into a coherent Gaussian point cloud, facilitating efficient novel view synthesis. Extensive qualitative and quantitative evaluations of public datasets demonstrate that our method consistently outperforms existing methods in terms of reconstruction quality and inferring efficiency.**

***Index Terms*—3D Generation, Novel view synthesis, Cross-modal Generation, Decoupling.**

I. Introduction

Reconstructing high-quality 3D models from a single 2D image is a long-standing and challenging problem in the fields of computer vision and graphics [1]–[5]. This task is crucial in applications like video games, virtual reality (VR), and augmented reality (AR). The main challenge of single-view reconstruction lies in the limited visual information provided by a single image. A single image captures only a partial view of an object, lacking depth information and often obscuring certain features [6], [7]. These limitations hinder accurate reconstruction of the complete 3D structure.

Recently, the development of diffusion models [8], [9] and intermediate representations like NeRF [10] and 3D Gaussian Splatting (3DGS) [11] has significantly advanced single-view reconstruction. Multi-view generation diffusion models [3], [12]–[15] have expanded the potential to capture and represent the spatial characteristics of objects from a single image, while 3D Gaussian techniques have markedly improved the speed of model reconstruction and rendering. Specifically, 3DGS employs Gaussian distributions based on point clouds to explicitly represent 3D objects, and employs point-cloud-based rendering techniques (splatting) to rapidly synthesize novel views. Owing to its efficient rendering mechanism and robust representational capabilities, methods based on 3DGS have rapidly evolved and become a mainstream intermediate representation. Existing 3DGS-based methods can be broadly classified into three categories: 1) 3D object generation based on Score Distillation Sampling (SDS) optimization [16]–[19]; 2) 3D object generation based on ViT encoding and Triplane decoding [7], [20]; and 3) 3D object generation based on multi-view information fusion [21], but they still face notable limitations. Although these 3DGS-based methods adopt different architectures, they all jointly optimize the positional and texture attributes of Gaussian point clouds. In this case, the texture information of the Gaussian point clouds is inferred through 2D rendering losses, where the splatting operation is performed on the Gaussian point clouds to synthesize new views, and the loss is calculated by comparing with the ground truth (gt) images. The splatting operation requires determining all the attributes of the Gaussian point clouds. The entangled treatment of positional and texture attributes during decoding from a shared network causes interference between the prediction of geometry and texture. Adjustments in geometry can alter the supervision of texture, making it difficult for the shared network to learn the correct optimization direction. By decoupling these attributes, each component: geometry generator, texture generator, and Gaussian attributes decoder can focus on its specific sub-task, thereby enhancing training stability and reconstruction quality.

We introduce the Decoupled Material-Position Network (DMPN), which effectively separates the generation of texture features from that of Gaussian position attributes. This strategic decoupling prevents interference between coarse geometry prediction and texture optimization, thereby mitigating Gaussian artifacts and enhancing the overall reconstruction quality. The DMPN comprises a Geometry Generator for accurate point cloud prediction and a Texture Generator for detailed triplane texture field creation. The Gaussian Attributes Decoder then integrates these attributes into a coherent Gaussian point cloud, enabling efficient novel view synthesis. Extensive qualitative and quantitative results on public datasets demonstrate that our method surpasses existing baselines in

† Equal contribution.
⋆ Corresponding author.
This work was supported by National Natural Science Foundation of China (No. 62202151).

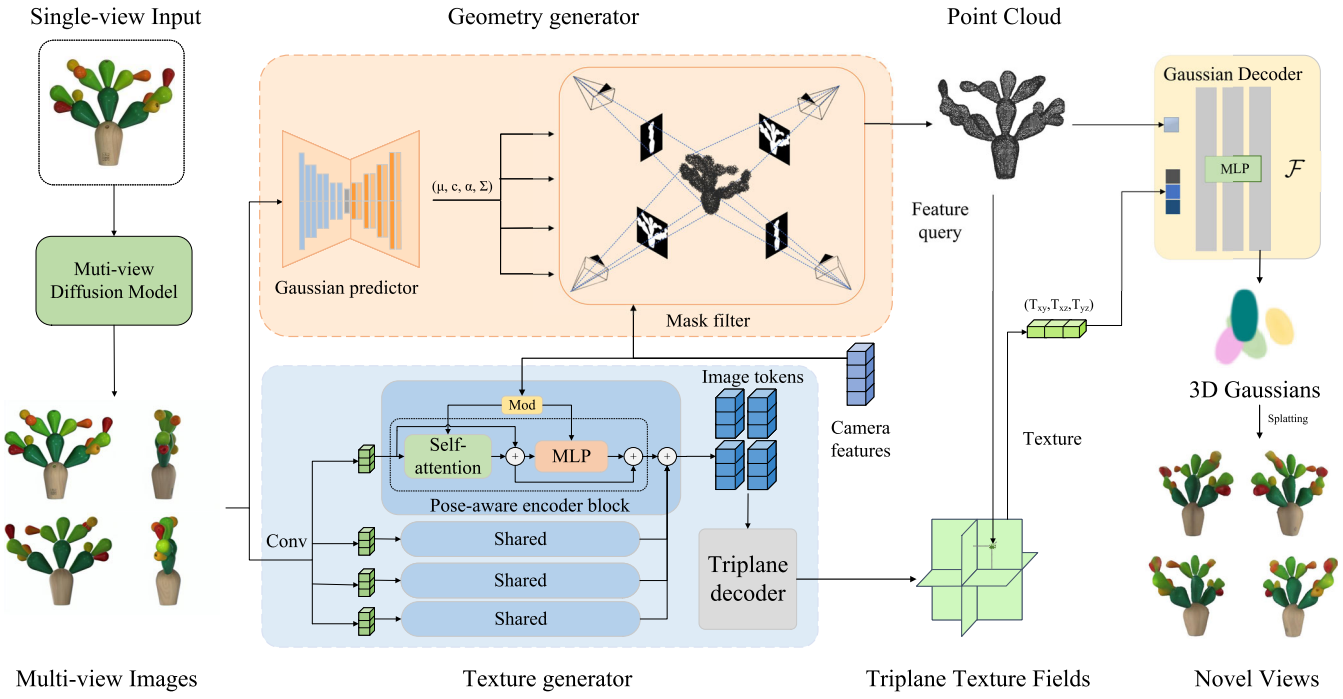

Fig. 1. Framework overview of **DMPN**. Overall, our framework consists of three main components: a multi-view generator G_M, a hybrid representation generator G_H, and an artifact-resistant Gaussian decoder G_D. The multi-view generator G_M takes a single image I as input and generates multiple views V from fixed camera positions. The hybrid representation generator G_H integrates a geometry generator and a texture generator. The geometry generator uses a gaussian predictor and a mask filter module to generate the point cloud P, while the texture generator creates the triplane representation T using camera features from fixed perspectives and the multi-view images V. Finally, the Gaussian decoder G_D decodes both the triplane T and the point cloud P to obtain 3D Gaussians G_u.

both quality and efficiency.

II. METHOD

A. Multi-view Generation

Our framework is illustrated in Fig. 1. We leverage the OpenLRM's pre-trained weights to speed up model convergence [20], [22]. Given that LRM is trained primarily on white-background images, it performs best with similar input. We adopt pre-trained weights from existing multi-view diffusion models [14], [15] and render white-background images from OmniObject3D [23] for fine-tuning, enabling consistent generation from four orthogonal views without post-processing. The original model produces grayscale background images from six perspectives. During fine-tuning, we use the query image as a constraint, stitch the images in a fixed sequence for denoising, and introduce conditional image scaling and camera pose noise to improve resolution adaptability. Fine-tuning only modifies the background color, preserving the original generative capabilities while ensuring consistent white-background outputs.

B. Hybrid generator

In existing 3D Gaussian methods, texture and geometry predictions are often entangled, leading to optimization conflicts and artifacts. To resolve this, our Hybrid Generator decouples texture and geometry generation. We extract textures from triplanes and decode gaussian attributes via a MLP. This structured approach ensures accurate texture optimization independent of geometric errors, resulting in improved reconstruction quality.

Geometry generator. Our geometry generation is inspired by the splatter-image [6], which predicts pixel-aligned Gaussians from images. We obtain the Gaussian representation $G_u = (\sigma, \mu, \Sigma, c)$ from each RGB image, and obtain the point cloud using the position attribute μ of the Gaussian. We perform fusion on multi-view point clouds. For the multiple perspective images generated by the diffusion model, we use a U-Net S that predicts pixel-aligned Gaussians to obtain the point cloud for each perspective. The point clouds P_{I_i} from other fixed camera poses $(R_i,\ T_i)$ are transformed into the perspective of the input view for fusion: $P_{I_i} = R_i P_{I_j} + T_i$, where R_i and T_i are the rotation and translation matrices for the i-th camera pose.

The rough point cloud obtained from the fusion P_{fusion} is projected onto multiple view image planes: $(u, v) = \Pi(K, T, P_{fusion})$, where K is the camera intrinsic matrix and T is the camera extrinsic matrix. Here, Π is the projection operator. The projection yields image coordinates (u, v). Noise points are removed based on visibility, mask criteria. Only the point clouds that can be projected within the mask M are retained:

$$P_{final} = \{p \in P_{fusion} \mid (u, v) \in M\}. \tag{1}$$

Texture generator. To extract the textural information of the Gaussians, we first encode multi-view images into pose-aware image tokens using a ViT model. During the image encoding phase, we design a Pose-aware encoder block that applies adaLN [24] to inject camera features into the image tokens, enabling distinction between different viewpoints. The encoded image tokens are then concatenated as input for the triplane decoder. In the triplane decoding phase, the learnable triplane tokens are concatenated with pose-related image tokens. These combined tokens are passed through self-attention layers and MLPs to obtain the triplane features. Finally, through deconvolution and reshaping operations, we derive the triplane texture fields.

C. Gaussian Decoder

A decoding MLP further converts the hybrid representation into Gaussian attributes. The hybrid representation includes a point cloud P which provides explicit geometry, and a triplane T which encodes an implicit feature field. For a given position $\mathbf{x} \in \mathbb{R}^3$ from the point cloud P, we query feature f from the triplane and adopt an MLP $\mathcal{F}$ to decode the attributes of the 3D Gaussians derived from the hybrid representation. The 3D Gaussians' attributes include opacity α, anisotropic covariance (represented by a scale s and rotation q), and spherical harmonics (sh) coefficients sh.

$$(\sigma, \mu, \Sigma, c) = \mathcal{F}(\mathbf{x}, f). \tag{2}$$

Before being used as position queries to obtain structured texture features, the positions of the Gaussian point cloud undergo an upsampling step and are then filtered again by the mask filter in the geometry generator to ensure structural reliability. After the Gaussian point clouds are obtained, we can efficiently perform novel view synthesis through the splatting operation.

D. Training Setup

To train the geometry generator, we adopt the LVIS subset of Objaverse [25] as the dataset $\mathcal{A}$, where I_i is a source image, I_j is a target image, and θ represents the viewpoint change between the source and target cameras. The source image I_i is fed into geometry generator, and we minimize the average reconstruction loss of the target view I_j:

$$\mathcal{L}_{\text{geo}}(T) = \frac{1}{|\mathcal{A}|} \sum_{(I_i, I_j, \theta) \in \mathcal{A}} \left\| I_j - \mathcal{M}(T(I_i), \theta) \right\|^2. \tag{3}$$

$\mathcal{M}$ represents the transformation from the source view to the target view, where the source image $\boldsymbol{I_i}$ is fed into the model, and the average reconstruction loss of the target view $\boldsymbol{I_j}$ is minimized. $\boldsymbol{T}$ denotes the inference operation.

The texture generator and gaussian decoder are jointly trained on a portion of the Cap3D subset [26] of Objaverse, independent of the geometry generator and multi-view generation model fine-tuning. During point cloud generation training, we adopt a splatter-image design for single-view generation. Each pixel only stores a single Gaussian parameter, and high-resolution training significantly improves the quality of the generated Gaussian points. We retrained this architecture so that it can take higher-resolution four-view images as input.

Our texture generator inherits pretrained weights from previous LRM-like works [22], [27] and is trained together with the Gaussian decoder after adjusting its camera pose settings. The loss function includes both image rendering and geometric structure supervision.

To supervise the Gaussian decoder, we use a differentiable renderer to generate RGB and alpha images from 16 views of the 3D Gaussians at each training step. The RGB images are optimized using mean square error loss, VGG-19 LPIPS loss, and SSIM loss. For faster shape convergence, we apply MSE loss to the alpha image:

$$\begin{aligned}\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\text{MSE}}(I_{\text{rgb}}, I_{\text{GT}}) + \lambda_{lpips}\mathcal{L}_{\text{LPIPS}}(I_{\text{rgb}}, I_{\text{GT}}) \\ + \lambda_{ssim}\mathcal{L}_{\text{SSIM}}(I_{\text{rgb}}, I_{\text{GT}}),\end{aligned} \tag{4}$$

$$\mathcal{L}_{\alpha} = \mathcal{L}_{\text{MSE}}(I_{\alpha}, I_{\text{GT}}). \tag{5}$$

The final loss function is expressed as:

$$\mathcal{L}_{\text{render}} = \lambda_{rgb}\mathcal{L}_{\text{rgb}} + \lambda_{\alpha}\mathcal{L}_{\alpha}. \tag{6}$$

III. EXPERIMENT

A. Implementation Details

Datasets. Our geometry generator was trained on the LVIS subset of Objaverse. Our Gaussian decoder is trained on the filtered Cap3D subset of Objaverse, excluding items with negative annotations or missing texture data. We rendered 512x512 images with a white background using a Blender script for training and validation.

GSO (Google Scanned Objects) [28] is a popular test set in 3D generative tasks. To benchmark our framework against existing methods, we will also evaluate the performance on this dataset, encompassing quantitative, qualitative, and ablation studies.

Training Details. The image encoder is initialized with the official DINO [29] pre-trained weights. The triplane decoder and NeRF MLP are initialized using the default initializer provided by PyTorch. We found that the pre-normalization transformer demonstrates robustness to various linear layer initializations. We train our model using the AdamW optimizer, with β_2 set to 0.95. We use a peak learning rate of 4×10^{-4} with a linear warm-up during the first 3K steps, followed by a cosine decay schedule. Non-bias and non-layernorm parameters have a weight decay of 0.05, and gradient clipping is applied at 1.

Since we cannot use the clone and split operations from the original 3DGS, training the feed-forward generator causes the Gaussians to become overly sharp and attempt to mimic fine texture details, leading to blurry or corrupted results. To address this issue, we empirically set canonical isotropic scales and rotations for all Gaussian point clouds to stabilize the training process. Specifically, the canonical isotropic scale of each 3D Gaussian is set to 0.03.

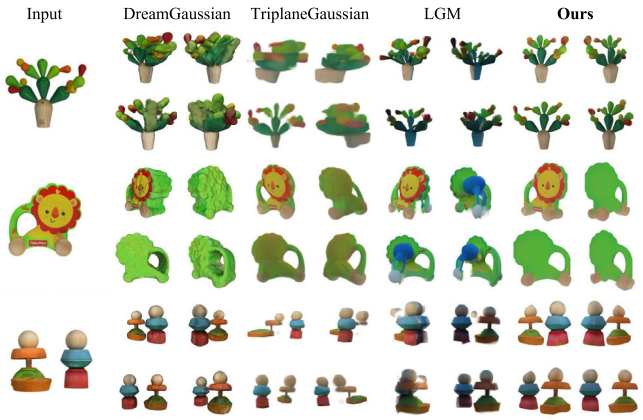

Fig. 2. Qualitative result of single view reconstruction

TABLE I
QUANTITATIVE RESULTS ON GSO ORBITING VIEWS,

Methods	PSNR↑	SSIM↑	LPIPS↓	Inference Time
DreamGaussian	16.18	0.8853	0.138246	$\sim$ 2 mins
TriplaneGaussian	20.21	0.9146	0.088134	$\sim$ **5 s**
LGM	21.33	0.9136	0.086090	$\leq$ 15 s
Ours(DMPN)	**22.47**	**0.9312**	**0.085740**	$\leq$ 15 s

B. Qualitative Results and Quantitative Comparisons

We compared our method with three other methods: DreamGaussian [18], TriplaneGaussian [7], and LGM [21] on the GSO dataset. As shown in Fig. 2, our DMPN method demonstrates superior quality, consistency, and faster performance across various novel views compared to the baseline methods. Our method preserves texture details in unseen areas, maintains accurate geometry, and reduces Gaussian artifacts around object edges.

We further validated the effectiveness of our approach by comparing the predictive outcomes of our model with the 3D ground truth data on the test set, as well as several recent baselines. Before the measurement, all the RGBA render results were filled with a uniform white background. We evaluated common metrics used in novel view synthesis tasks, including PSNR, SSIM, and LPIPS [30]. As indicated in Fig. 2 and Table I, both the qualitative and quantitative results confirm that our method consistently outperforms the existing baselines.

C. Ablation Studies

In this section, we conduct thorough experiments to validate the effectiveness of our proposed components using the available 3D ground truth from GSO. The results of ablation experiments are shown in Table II and Fig. 3. The results strongly validate the effectiveness of our modifications.

Firstly, we validated the necessity of constraining the central positions of Gaussian point clouds using filters during the geometry generation phase. ”w/o geometry filter” denotes the absence of mask filters. Although the texture information of the Gaussian point clouds is stored in structured planes, the point clouds predicted by the U-Net often contain noise points that do not conform to the actual structure. These noise points lead to white or light-colored artifacts around the objects during the Gaussian decoding phase (e.g., the shoes in Fig. 3). Secondly, we validated the necessity of the texture generator. ”w/o texture generator” refers to the removal of the texture field generator branch, compelling the geometry generator to simultaneously generate all attributes. As shown in Fig. 3, the geometry generator alone struggles to produce detailed texture information, making it essential to extract textures from the triplane texture field.

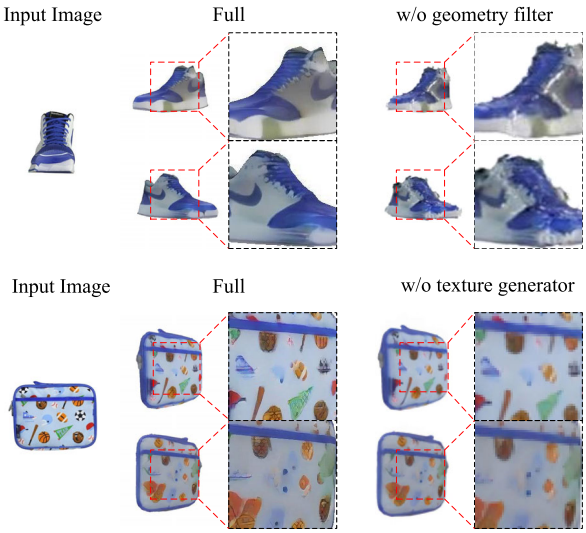

Fig. 3. Ablation Results

TABLE II
RESULT OF ABLATION STUDIES

Methods	PSNR↑	SSIM↑	LPIPS↓
w/o geometry filter	18.22	0.9048	0.137920
w/o texture generator	20.67	0.9172	0.097468
Full Model	**22.47**	**0.9312**	**0.085740**

IV. CONCLUSION

In this work, we proposed DMPN, a novel single-view 3D reconstruction pipeline that focuses on decoupling geometric and texture attributes. By strategically separating Gaussian position attributes and texture features, our approach addresses the common issue of interference between geometry prediction and texture optimization in previous methods. This decoupling not only reduces geometric artifacts but also significantly enhances the stability and quality of the reconstruction process. Extensive qualitative and quantitative evaluations on public datasets demonstrate that our decoupling strategy consistently outperforms existing 3DGS-based reconstruction methods in both reconstruction quality and efficiency.

REFERENCES

[1] H. Jun and A. Nichol, "Shap-e: Generating conditional 3d implicit functions," *ArXiv*, vol. abs/2305.02463, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258480331

[2] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *ArXiv*, vol. abs/2212.08751, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254854214

[3] Y. Liu, C.-H. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang, "Syncdreamer: Generating multiview-consistent images from a single-view image," *ArXiv*, vol. abs/2309.03453, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261582503

[4] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 9298–9309.

[5] X. Long, C. Lin, L. Liu, W. Li, C. Theobalt, R. Yang, and W. Wang, "Adaptive surface normal constraint for depth estimation," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12 829–12 838, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232404654

[6] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, "Splatter image: Ultra-fast single-view 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 208–10 217.

[7] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang, "Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 324–10 335.

[8] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Muller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *ArXiv*, vol. abs/2307.01952, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259341735

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.

[10] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[11] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[12] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt *et al.*, "Wonder3d: Single image to 3d using cross-domain diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9970–9980.

[13] Y. Shi, P. Wang, J. Ye, L. Mai, K. Li, and X. Yang, "Mvdream: Multi-view diffusion for 3d generation," *arXiv:2308.16512*, 2023.

[14] P. Wang and Y. Shi, "Imagedream: Image-prompt multi-view diffusion for 3d generation," *arXiv preprint arXiv:2312.02201*, 2023.

[15] Z. Wang, Y. Wang, Y. Chen, C. Xiang, S. Chen, D. Yu, C. Li, H. Su, and J. Zhu, "Crm: Single image to 3d textured mesh with convolutional reconstruction model," *arXiv preprint arXiv:2403.05034*, 2024.

[16] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=FjNys5c7VyY

[17] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," 2022. [Online]. Available: https://arxiv.org/abs/2112.01455

[18] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=UyNXMqnN3c

[19] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 300–309.

[20] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, "LRM: Large reconstruction model for single image to 3d," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=sllU8vvsFF

[21] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," *arXiv preprint arXiv:2402.05054*, 2024.

[22] Z. He and T. Wang, "Openlrm: Open-source large reconstruction models," https://github.com/3DTopia/OpenLRM, 2023.

[23] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian *et al.*, "Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 803–814.

[24] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1510–1519, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:6576859

[25] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.

[26] T. Luo, C. Rockwell, H. Lee, and J. Johnson, "Scalable 3d captioning with pretrained models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[27] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," *arXiv preprint arXiv:2404.07191*, 2024.

[28] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.

[29] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," 2022.

[30] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.