Sparse Monocular and Unconstrained 3D Language Gaussian Splatting

Shivali Bhatnagar
Samsung Research Institute, Bangalore
Bengaluru, India
shivali.b@samsung.com

Raushan Joshi
Samsung Research Institute, Bangalore
Bengaluru, India
raushan.j@samsung.com

Amogha D Shanbhag
Samsung Research Institute, Bangalore
Bengaluru, India
amogha.ds@samsung.com

Ankit Dhiman
Samsung Research Institute, Bangalore
Bengaluru, India
ankit.dhiman@samsung.com

Raveendra Karu
Samsung Research Institute, Bangalore
Bengaluru, India
r.karu@samsung.com

Kaushik Das
Samsung Research Institute, Bangalore
Bengaluru, India
kaushik.d@samsung.com

***Abstract*—Human interaction with reconstructed real 3D scene from monocular/stereo/multi-view images often require language queries to locate, move and transform 3D objects. Modeling of a real-world scene requires multi-view reconstruction/ novel-view synthesis methods which require known camera intrinsic and extrinsic parameters. Recent methods like LangSplat and LeRF requires these calibration parameters for embedding language features in 3D which limits the practicality of these methods. To mitigate this, we present an end-to-end framework ¡name-of-your-method¿ to embed language features in 3D space from unconstrained and uncalibrated monocular camera sequences. This enhances the practical application of our method. We employ a Siamese based auto-encoder architecture inspired from DUSt3R to get dense point cloud along with the camera poses. We utilize this dense point cloud to train a 3D Gaussian splatting method along with language embedding features. We test our method in indoor scenarios with a limited number of camera images (5-10 images) and demonstrate accurate detection of objects in the resulting 3D model. Further, our method reduces overall training time by 50 percent compares to a current state-of-the-art method.**

***Index Terms*—Computer Vision, Computer Graphics, 3D Reconstruction, Semantics**

I. Introduction

The landscape of 3D scene reconstruction and language-guided interaction has traditionally relied on MVS techniques. They require extensive computational resources and precise camera calibration, often using complex pipelines like SfM or COLMAP [1]. These methods, while powerful, are limited by their computational intensity, and specifically, the reliance on well-calibrated images.The sequential processing requirement impose practical constraints, making them less feasible for dynamic, real-world applications, particularly where monocular images are involved. The existing state-of-the-art methods in 3D scene reconstruction and language-guided interaction such as LeRF [2] and LangSplat [3] continue to depend heavily on calibrated camera setups and intricate SfM processes, limiting their applicability in unconstrained environments. Recent groundbreaking work on 3D scene construction with uncalibrated camera [4], [5] images has opened the possibilities of more research in both academic and industrial areas. In this paper, we present an architecture for end-to-end language-aware 3D scene reconstruction using unconstrained monocular camera images, significantly expanding the applicability of 3D scene modeling in practical scenarios. Our approach diverges from prior methods like LeRF and LangSplat by eliminating the need for calibrated camera images and time-consuming COLMAP processes. Instead, we employ a Siamese-based pre-trained autoencoder inspired by the DUSt3R framework [6], which efficiently generates depth maps and point clouds from unposed images, thereby reducing the dependency on traditional MVS. Furthermore, our model integrates SAM [7] + JinaCLIP [8] block following the work of LangSplat to avoid point ambiguity and memory cost issues. The execution of the DUSt3R pipeline block and SAM + JinaCLIP in parallel reduces the overall training time by a significant margin. In our experiments, we demonstrate that our proposed architecture in Fig. 1 is capable of object detection and localization within a rendered 3D scene using prompt queries, achieving results comparable to those of traditional SfM-based models while relying solely on uncalibrated monocular view images. To summarize, our key contributions are: 1. An effective framework for language-aware 3D scene that uses uncalibrated monocular camera images. 2. Significant reduction in the number of monocular images required to achieve considerable results. 3. The architecture design enables the simultaneous execution of multiple independent components, resulting in a twofold increase in training speed.

II. Related Work

A. SfM points and Camera pose estimation

SfM has become a powerful technique for reconstructing 3D scenes from multiple images captured by a moving camera. Traditional SfM algorithms [1] estimate camera poses and the 3D structure simultaneously using feature detection, matching, and bundle adjustment [9]. However, these approaches rely on hand-crafted features and matching techniques, which can be sensitive to noise, illumination changes, and viewpoint

variations. These limitations [10] can lead to inaccurate camera pose estimates and incomplete or noisy 3D reconstructions, particularly in challenging scenarios like dynamic scenes or scenes with repetitive textures.

Camera pose estimation is another critical component of SfM, and traditional approaches such as the Direct Linear Transformation (DLT) algorithm and the Iterative Closest Point (ICP) [11] algorithm have been widely used. However, these approaches have limitations, including sensitivity to outliers, poor performance in low-texture regions, and high computational complexity. Traditional camera pose estimation algorithms often rely on explicit feature extraction and matching, which can be computationally expensive and error-prone in complex scenes.

B. 3D Gaussian Splatting

3D Gaussian Splatting uses anisotropic 3D Gaussians to depict radiance fields, coupled with differentiable splatting for real time rendering [14]. Method described in [15], proposes 4D Gaussians for real time rendering of dynamic scenes at high image resolutions. Recently proposed work [16], [17], on reducing memory requirement for efficiently processing 3D Gaussian splatting, where huge resource consumption were drawback. In parallel, various research is now utilising 3D Gaussians splatting with Diffusion based models [18], [19], [20] to achieve high quality text-to-3D image generation. Methods like ouroborus3D [21] utilizes 3D generative network and DreamGaussian [22] uses 3D Gaussian Splatting models for 3D content generation. Inspired from the similar work [3], we will utilize language embedding into 3D Gaussian Splatting to enable open vocabulary 3D queries.

C. 3D Language Features

Fusing language into 3D Vision has been explored in multi-modal context like 3D Vision question answering (VQA) [23], [24], [25] which extracts information from 3D space and its objects to answer a query. Few optimization based methods incorporate distillation [26], [27] of rich prior learned in the 2D space given to optimize 3D representation per-prompt. Initial efforts in 3D scene reconstruction leveraged CLIP guidance to enhance multi-view image-text alignment [28]. ConceptFusion [5] integrates CLIP features more densely into RGBD point clouds, utilizing Mask2Former [29] to predict regions of interest. However, this approach may fail to detect objects that are out of distribution relative to Mask2Former's training set. LERF [2] was the first to embed CLIP features into NeRF, enabling open-vocabulary 3D queries leveraging the powerful CLIP representation. DINO features were also used for supervising LERF to improve its performance. Liu et al. [30] also utilized CLIP and DINO [31] features to train a NeRF model for 3D open-vocabulary segmentation. Notably, the expensive computation burden of NeRFs results in long training time, which motivates concurrent works to adapt the representation of 3D Language Gaussian Splatting [3] to obtain efficient 3D language features.

III. METHODOLOGY

Given a sequence of unposed images captured by normal phone camera with unknown parameters, our aim is to recover camera poses and embed language feature to reconstruct the photo realistic scenes with open vocabulary querying feature. First step of the process shown in Fig. 1 involves extraction of only relevant features like pointcloud and depth map using framework inspired from the work of DUSt3R. The original 3D Gaussian Splatting (3DGS) method involves the parameterization of shape, position and color properties, which will be explained in following sections. However, adding language properties to each Gaussians will add extra dimensions and thus make it 3D Language Gaussians. In the proposed approach, the components SAM-JinaCLIP pair for extracting language feature, will run in parallel to the 3DGS training component. This reduces the overall training time significantly.

A. Integration of DUSt3R point cloud with Gaussian Splatting

Different from its implicit counterparts like NeRF [32] and NeRF-based approaches [2], [26] for rendering scenes, 3D Gaussian Splatting (3D-GS) [14] is a technique used for rendering high-quality 3D scenes from point clouds using a set of 3D Gaussians. A 3D Gaussian is parameterized by a mean vector $\mathbf{x} \in \mathbb{R}^3$ and a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$:

$$G(\mathbf{p}) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{p}-\mathbf{x})^T\Sigma^{-1}(\mathbf{p}-\mu)\right) \tag{1}$$

where $\Sigma = RSS^TR^T$, with R being a rotation matrix and S being a scaling matrix, ensuring that the covariance matrix remains positive semi-definite. This Gaussian splat is projected onto a 2D plane for rendering, using view-dependent appearance modeled by spherical harmonics (SH) of order 3 and direct color components. The rendered color c_i for a particular viewpoint involves α-blending the colors of individual Gaussians:

$$\mathbf{C}(\mathbf{p}) = \sum_{i=1}^{n} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad \alpha_i = o_iG(\mathbf{p}) \tag{2}$$

where o_i represents the opacity of the i-th Gaussian at pixel point p, evaluated through a 2D Gaussian with its 3D covariance projected into the camera coordinate system. Traditionally, SfM methods were employed to estimate camera poses and generate dense 3D reconstructions from multiple images.

In our approach, we create a data-loader to utilize the output of DUSt3R like pointmaps, camera poses, camera inrtrinsics and depthmaps to initialize Gaussian Spaltting. The pointmap X of the observed scene is obtained from ground-truth depthmap by $D \in \mathbb{R}^{W\times H}$ as $X_{i,j} = K^{-1}\begin{bmatrix} iD_{i,j} & jD_{i,j} & D_{i,j}\end{bmatrix}^\top$ where K represents the camera intrinsics matrix and X is expressed in camera coordinate frame. The pointmap $X^{n,m}$ from camera n expressed in camera m's coordinate frame is expressed as:

$$X_{n,m} = P_mP_n^{-1} \cdot h(X_n) \tag{3}$$

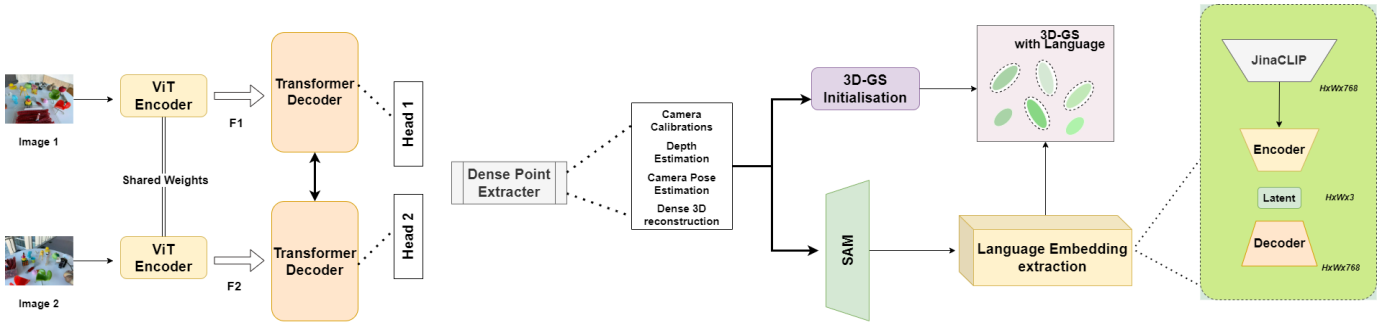

Fig. 1: The framework takes unconstrained monocular camera images and passes it through DUSt3R framework [6] to get point clouds. Further, data-loader extracts camera calibration details which is required for creating 3D gaussian splats. In parallel, Langsplat framework gets language fields through SAM [7] + JinaCLIP [8] + Autoencoder while removing point ambuiguity issue and reducing memory cost issue. In the end, language features gets integrated with each gaussian splats to form 3D language gaussian splatting.

where $P_m, P_n \in \mathbb{R}^{3\times4}$, the world-to-camera poses for images n and m, and $h : (x, y, z) \rightarrow (x, y, z, 1)$, the homogeneous mapping. DUSt3R replaces $\chi^n_{i,j} := \mathbf{P}_n^{-1} h(\mathbf{K}_n^{-1}[iD^n_{i,j}; jD^n_{i,j}; D^n_{i,j}])$ (i.e. enforcing a standard camera pinhole model as in Eq. 3), and thus all camera poses $\{\mathbf{P}_n\}$, associated intrinsics $\{\mathbf{K}_n\}$, and depthmaps $\{D^n\}$ for $n = 1, \ldots, N$ can be estimated. This allows us to generate dense 3D point clouds from monocular images, eliminating the need for extensive image matching and pose estimation steps typically required in SfM.

B. The LangSplat approach for 3D GS Language field

Directly learning image features using JinaCLIP can lead to point ambiguity because JinaCLIP embeddings are image-aligned rather than pixel-aligned. To address this issue, we leverage the foundational model for image segmentation, SAM [7], which excels at grouping pixels with their surrounding counterparts belonging to the same object, resulting in object masks with clear boundaries. SAM tackles point ambiguity by generating three hierarchical masks (whole, part, subpart) for a point prompt. In this paper, we utilize SAM to capture the semantic hierarchy of objects in 3D scenes, achieving accurate and multi-scale segmentation maps. With proper filtering of segmented maps using predicted (Intersection over Union) IoU score, stability score, and overlap rate, we get three refined segmentation maps: M_s, M_p, and M_w. Next, we extract features using JinaCLIP to capture the semantic context of objects at various levels within the scene. Many previous methods [33], [34] compute image-level features rather than pixel-level features, and performs poorly as explained in this work [3]. We took the advantage of predefined semantic scales using SAM. The pixel-aligned language embeddings are mathematically represented as:

$$Ltl(v) = V(It \odot Ml(v)), \quad l \in \{s, p, w\} \tag{4}$$

where $Ml(v)$ is the mask region to which pixel v belongs at semantic level l. Given that the JinaCLIP embeddings are high-dimensional (D = $H \times W \times 768$), storing millions of 3D Gaussians with these embeddings will significantly spike the memory as well as time cost and, in some cases, may lead to out-of-memory (oom) errors. As mentioned here [3], all the segmented regions in a scene are sparsely distributed in the JinaCLIP latent space, which is a highly compact D-dimensional latent space with multistage training on **400 million (image, text) + 100K (text, text) + 1.1M (image, large text) pairs** [8]. Thus, we can further compress these JinaCLIP features using a scene-specific lightweight autoencoder. We learn an encoder E and a decoder Ψ to reconstruct the original JinaCLIP embeddings from the compressed representation.

$$H_{lt}(v) = E(L_{lt}(v)) \in \mathbb{R}^d, \text{where } d \ll D, L_{lt}(v) \in \mathbb{R}^D \tag{5}$$

We will be using $d = 3$ (as in [3]), as it yields excellent model efficiency and accuracy. Following the work [2], [3], incorporating language feature into the gaussians will enable the open-vocabulary query into the 3D scenes. Integration of language feature in i-th 3D Gaussians for given pixel v is given as:

$$F_l(v) = \sum_{i \in N} f_l(i) T(i), \quad l \in \{s, p, w\} \tag{6}$$

where $T(i) = \prod_{j=1}^{i-1}(1 - \alpha_j)$, and N is the set of sorted Gaussians overlapping with the given pixel v, and $F_l(v)$ represents the language embedding rendered at pixel v with the semantic level l. Referring to LERF [2], we computed relevancy scores for text queries in final 3D object localization and semantic segmentation tasks using JinaCLIP embeddings. For each text query, the relevancy score is calculated as:

$$\text{RS} = \min_i \frac{\exp(\phi_{\text{img}} \cdot \phi_{\text{qry}})}{\exp(\phi_{\text{img}} \cdot \phi_{\text{qry}}) + \exp(\phi_{\text{img}} \cdot \phi_{\text{i,canon}})} \tag{7}$$

where RS is relevancy score. This score measures the closeness of the rendered embedding to the query relative to canonical embeddings.

IV. Experimental Setup

Datasets and metrics: We have evaluated our method on the LERF-based dataset [3]. While other methods like LERF and LangSplat depend on SfM points and corresponding

Methods	Calibrated Images	Processing Time	Number of Images Required
LeRF [2]	Yes	100 min	100-400
LangSplat [3]	Yes	30 min	100-400
Ours	**No**	**10 min**	**3-15**

TABLE I: Characteristic comparison of Our method with other baselines. Compared to baselines, our method works with uncalibrated and less number of images, and is fast.

Test Scenes	Ours(No. of Images)				Lseg.	LeRF	LangSplat
	3	5	10	15	>200	>200	>200
ramen	65.1	63.0	64.2	63.6	14.1	62.0	73.2
figurines	60.7	71.5	73.3	76.1	8.9	75.0	80.4
teatime	73.1	77.9	77.7	78.1	33.9	84.8	88.1
waldo_kitchen	76.3	78.4	81.0	81.4	27.3	72.7	95.5
overall	**68.8**	**72.7**	**74.0**	**74.8**	21.1	73.6	**84.3**

TABLE II: Comparison of Localization Accuracy (%) on LERF Dataset. Here, other methods uses >200 images of the dataset.

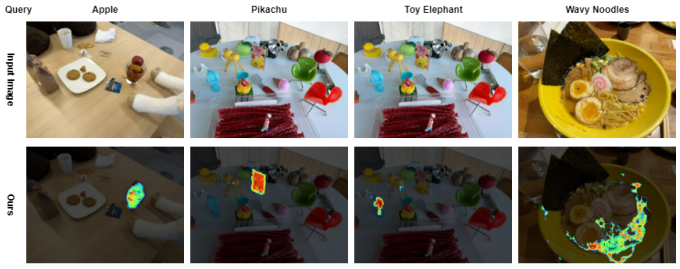

Fig. 2: Qualitative results for 3D object detection for given scenes from LeRF dataset [3]

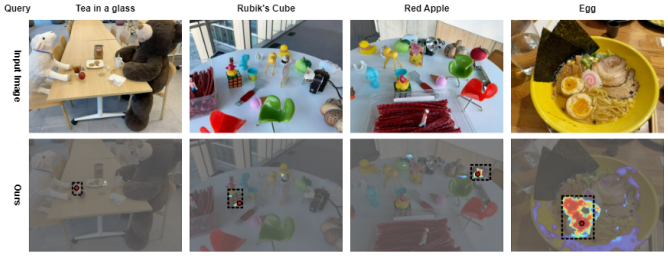

Fig. 3: Qualitative results for 3D object localization for given scenes from LeRF dataset [3]

camera properties and depth maps, our method utilizes only images from the original dataset. In order to compare the results with benchmarked methods, we keep the ground truth masks of textual queries for open-vocabulary search. The dataset contains multiple images from four different indoor scenes named as teatime, figurines, ramen, and waldo_kitchen along with camera calibrations and depth maps. For our testing setup, we have split the scenes into sets of varying numbers of images, specifically 3, 5, 10 and 15 images each. We, then, calculated accuracy for 3D object localization tasks based on the query term and also the mIoU results for 3D semantic segmentation. The metrics are kept similar for comparison with baseline methods.

Test Scenes	3 (Ours)	5 (Ours)	10 (Ours)	15 (Ours)	Lseg.	LerF	LangSplat
ramen	27.9	30.0	26.2	31.6	7.0	28.2	51.2
figurines	20.3	31.2	28.3	30.9	7.6	38.6	44.7
teatime	33.4	37.6	39.5	38.8	21.7	45.0	65.1
waldo_kitchen	28.4	34.6	33.9	40.3	29.9	37.9	44.5
overall	**27.5**	**33.3**	**31.9**	**35.4**	16.6	37.4	51.4

TABLE III: Mean mIoU scores (%) comparison of 3D semantic segmentation. Here, other methods uses >200 images of the dataset.

Implementation Details:To extract depth maps and camera calibration details for each scene set, our architecture employs known ViT-Large for the encoder [35], a ViT-Base for the decoder, and a DPT head [36]. For 2D segmentation, we use the known SAM with ViT-H model and JinaCLIP model. We train 3D Gaussian Splatting for only 7000 iterations because our scene set contains very few images compared to other methods. When training the 3D Gaussians with extracted language features, other learnable features like opacity are made non-trainable. We performed the experiment on single NVIDIA RTX A6000 GPU, which took approximately 8 minutes for end-to-end flow.

Results: Evaluation of our method on object localization and detection tasks is summarized in Tables II and III, respectively, demonstrates competitive results compared to baseline models. Additionally, Table I provides a comparison of training feasibility, highlighting that our method excels across various aspects. And qualitative analysis, as depicted in Figures 2, 3, and 4, further illustrates the robustness of our approach, particularly in unconstrained environments.

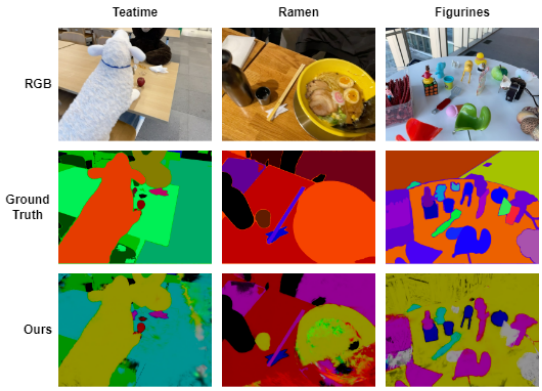

Fig. 4: Qualitative results for 3D segmentation with learned language features for given scenes

V. Conclusion

We have presented an architecture for end-to-end language-aware 3D scene reconstruction using unconstrained monocular camera images, which greatly expands the practical application of 3D scene modeling. Our method eliminates the need for calibrated camera images and delivers competitive results with significantly less number of input images. By emphasizing these advancements, our research aims to make 3D scene reconstruction more practical and versatile, paving the way for new academic research and industrial applications.

REFERENCES

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," 2023. [Online]. Available: https://arxiv.org/abs/2303.09553

[3] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2312.16084

[4] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang, "Colmap-free 3d gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2312.07504

[5] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. B. Tenenbaum, C. M. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," 2023. [Online]. Available: https://arxiv.org/abs/2302.07241

[6] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," 2023. [Online]. Available: https://arxiv.org/abs/2312.14132

[7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643

[8] A. Koukounas, G. Mastrapas, M. Günther, B. Wang, S. Martens, I. Mohr, S. Sturua, M. K. Akram, J. F. Martínez, S. Ognawala, S. Guzman, M. Werk, N. Wang, and H. Xiao, "Jina clip: Your clip model is also your text retriever," 2024. [Online]. Available: https://arxiv.org/abs/2405.20204

[9] S. Weber, N. Demmel, T. C. Chan, and D. Cremers, "Power bundle adjustment for large-scale 3d reconstruction," 2023. [Online]. Available: https://arxiv.org/abs/2204.12834

[10] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," 2017. [Online]. Available: https://arxiv.org/abs/1701.08493

[11] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2021.3054619

[12] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, "Deepsfm: Structure from motion via deep bundle adjustment," 2020. [Online]. Available: https://arxiv.org/abs/1912.09697

[13] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," 2016. [Online]. Available: https://arxiv.org/abs/1505.07427

[14] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[15] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," 2024. [Online]. Available: https://arxiv.org/abs/2310.08528

[16] S. S. Mallick, R. Goel, B. Kerbl, F. V. Carrasco, M. Steinberger, and F. D. L. Torre, "Taming 3dgs: High-quality radiance fields with limited resources," 2024. [Online]. Available: https://arxiv.org/abs/2406.15643

[17] S. Niedermayr, J. Stumpfegger, and R. Westermann, "Compressed 3d gaussian splatting for accelerated novel view synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2401.02436

[18] H. Wen, Z. Huang, Y. Wang, X. Chen, Y. Qiao, and L. Sheng, "Ouroboros3d: Image-to-3d generation via 3d-aware recursive diffusion," 2024. [Online]. Available: https://arxiv.org/abs/2406.03184

[19] T. Yi, J. Fang, J. Wang, G. Wu, L. Xie, X. Zhang, W. Liu, Q. Tian, and X. Wang, "Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2310.08529

[20] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," 2022. [Online]. Available: https://arxiv.org/abs/2210.06978

[21] H. Wen, Z. Huang, Y. Wang, X. Chen, Y. Qiao, and L. Sheng, "Ouroboros3d: Image-to-3d generation via 3d-aware recursive diffusion," 2024. [Online]. Available: https://arxiv.org/abs/2406.03184

[22] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," 2024. [Online]. Available: https://arxiv.org/abs/2309.16653

[23] S. Ye, D. Chen, S. Han, and J. Liao, "3d question answering," 2022. [Online]. Available: https://arxiv.org/abs/2112.08359

[24] L. Zhao, D. Cai, J. Zhang, L. Sheng, D. Xu, R. Zheng, Y. Zhao, L. Wang, and X. Fan, "Towards explainable 3d grounded visual question answering: A new benchmark and strong baseline," 2022. [Online]. Available: https://arxiv.org/abs/2209.12028

[25] Z. Chen, F. Wang, Y. Wang, and H. Liu, "Text-to-3d using gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2309.16585

[26] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," 2022. [Online]. Available: https://arxiv.org/abs/2112.05139

[27] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," 2024. [Online]. Available: https://arxiv.org/abs/2312.03203

[28] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," 2023. [Online]. Available: https://arxiv.org/abs/2305.16213

[29] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022. [Online]. Available: https://arxiv.org/abs/2112.01527

[30] K. Liu, F. Zhan, J. Zhang, M. Xu, Y. Yu, A. E. Saddik, C. Theobalt, E. Xing, and S. Lu, "Weakly supervised 3d open-vocabulary segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2305.14093

[31] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.14294

[32] 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," 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[33] W. Li, X. Huang, Z. Zhu, Y. Tang, X. Li, J. Zhou, and J. Lu, "Ordinalclip: Learning rank prompts for language-guided ordinal regression," 2022. [Online]. Available: https://arxiv.org/abs/2206.02338

[34] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, p. 2337–2348, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1007/s11263-022-01653-1

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[36] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021. [Online]. Available: https://arxiv.org/abs/2103.13413