UFV-Splatter: Pose-Free Feed-Forward 3D Gaussian Splatting Adapted to Unfavorable Views

Yuki Fujimura[1,a)] Takahiro Kushida[2] Kazuya Kitano[1] Takuya Funatomi[3] Yasuhiro Mukaigawa[1]

Abstract

This paper proposes a novel adaptation framework that enables pretrained pose-free feed-forward 3D Gaussian splatting (3DGS) models to handle unfavorable views, i.e., cases where the object is not centered at the world origin and the cameras are not pointing toward the origin. We leverage priors learned from favorable images by feeding recentered images into a pretrained model augmented with low-rank adaptation (LoRA) layers. We further propose a Gaussian adapter module to enhance the geometric consistency of the Gaussians derived from the recentered inputs, along with a Gaussian alignment method to render accurate target views for training. Experimental results on both synthetic and real images validate the effectiveness of our method in handling unfavorable input views.

1. Introduction

3D Gaussian splatting (3DGS) [8] is a powerful technique for representing 3D scenes using volumetric Gaussians, optimized by reconstructing multi-view input images given known camera poses. However, estimating both accurate 3D Gaussians and camera poses becomes challenging in scenarios with only sparse input views, motivating the need to address pose-free sparse-view scenarios.

One promising approach to handling sparse input views is to learn feed-forward models [1], [2], [3], [4], [13], [14]. Feed-forward 3DGS models take sparse input views and directly output pixel-aligned 3D Gaussians. Recently, pose-free feed-forward 3DGS models [10], [12], [16] have been introduced to predict pixel-aligned Gaussians without relying on extrinsic or even intrinsic camera parameters.

To train such models for object-centric scenes, large-scale synthetic datasets such as Objaverse [5] are commonly used. The multi-view images for training are rendered from millions of 3D assets, typically with the object centered at the world origin and cameras pointing toward the origin—referred to as *favorable* views in this study (red cameras in Fig. 2). While this setup facilitates stable training, it limits the generalization ability of pose-free models to real-world scenarios with varying camera poses. Generalizing to such *unfavorable* views (green and blue cameras in Fig. 2) remains a significant challenge, often resulting in inconsistent Gaussian predictions across input views.

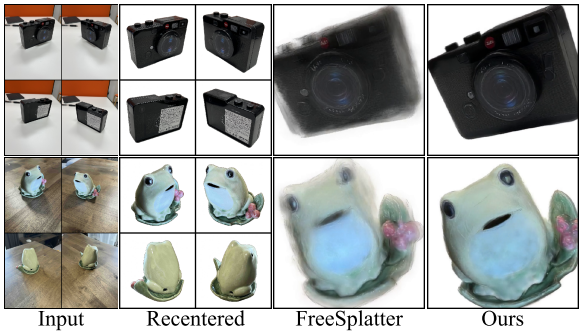


Fig. 1 Novel-view synthesis for daily captures with a smartphone. The foreground regions are masked and recentered prior to inference. These results demonstrate the practical applicability of our method in real-world scenarios with unknown and varying camera poses.

In this work, we propose a novel framework to adapt a pretrained pose-free feed-forward 3DGS model to handle unfavorable views. Our method adapts the pretrained model using low-rank adaptation (LoRA) [7]. To leverage the prior learned from favorable views, we recenter the images via simple image shifting and resizing. We further propose a Gaussian adapter module that improves geometric consistency in the Gaussians derived from the recentered images, along with a Gaussian alignment method that aligns the predictions for accurate target view rendering during training. As shown in Fig. 1, our method enhances the practical applicability of pose-free 3DGS models in real-world scenarios with unknown and varying camera poses.

2. Method

2.1 Pose-free Feed-forward 3DGS

We first review pose-free feed-forward 3DGS. Given a set of input context images $\{\mathbf{I}_i^c\}_{i=1}^N$, where each $\mathbf{I}_i^c \in \mathbb{R}^{H\times W\times 3}$, the task is to generate 3D Gaussians for each image without access to camera poses. Here, N denotes the number of input images, and H and W represent the height and width, respectively. A pose-free feed-forward model $\mathcal{F}$ predicts pixel-aligned 3D Gaussians as follows:

[1] Nara Institute of Science and Technology
[2] Ritsumeikan University
[3] Kyoto University
[a)] fujimura.yuki@is.naist.jp

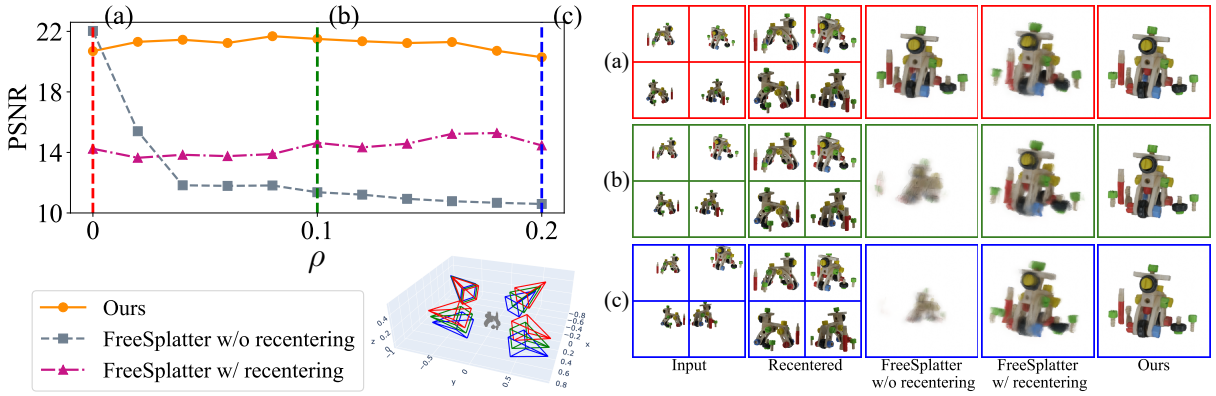


Fig. 2 *Favorable* views (red cameras, $\rho = 0$): the object is placed at the world origin, and the cameras are oriented toward the origin. *Unfavorable* views (green and blue cameras, $\rho = 0.1, 0.2$): generated by adding a translation of random direction and magnitude ρr to each favorable camera, where r denotes the distance from the object origin to the camera. FreeSplatter [10], trained only on favorable views, performs well when inputs are also favorable but struggles under unfavorable inputs (FreeSplatter w/o recentering). A naive approach, such as recentering the foreground to mimic favorable views, is insufficient to resolve these generalization issues (FreeSplatter w/ recentering). Our proposed method remains robust to unknown and varying camera poses.

$$\{\mathbf{G}_i\}_{i=1}^{N} = \mathcal{F}(\{\mathbf{I}_i^c\}_{i=1}^{N}), \tag{1}$$

where each output $\mathbf{G}_i \in \mathbb{R}^{H\times W\times(11+3(d+1)^2)}$ encodes the parameters of the predicted 3D Gaussians, including the Gaussian center $\boldsymbol{\mu}_i \in \mathbb{R}^{H\times W\times 3}$, opacity $\boldsymbol{\alpha}_i \in \mathbb{R}^{H\times W\times 1}$, rotation as quaternions $\mathbf{r}_i \in \mathbb{R}^{H\times W\times 4}$, scale $\mathbf{s}_i \in \mathbb{R}^{H\times W\times 3}$, and view-dependent color $\mathbf{c}_i \in \mathbb{R}^{H\times W\times 3(d+1)^2}$ using spherical harmonics of degree d. During training, the estimated Gaussians are used to render a target image $\mathbf{I}^t \in \mathbb{R}^{H\times W\times 3}$ for computing the training loss.

Our goal is to adapt pretrained $\mathcal{F}$ so that it generates accurate and consistent Gaussians even from unfavorable inputs, the definition of which is described in the next section.

2.2 Definition of Unfavorable View

As illustrated in Fig. 2, we define *favorable* views as camera poses oriented toward the object center (i.e., the world origin), and *unfavorable* views as those that deviate from this configuration. For synthetic data, we generate unfavorable views by applying translations to the favorable camera positions.

In Fig. 2, we construct unfavorable views by adding a translation of random direction and magnitude ρr to each favorable camera, where r denotes the distance from the object origin to the camera. When $\rho = 0$ (favorable views), the pretrained model $\mathcal{F}$, which is trained solely on such views, can synthesize novel views effectively. However, when $\rho > 0$ (unfavorable views), $\mathcal{F}$ often produces inconsistent Gaussians across input views, leading to degraded novel-view synthesis quality.

2.3 Model Architecture

2.3.1 Recentering images

Figure 3 provides an overview of the proposed model. Directly feeding unfavorable views into the pretrained model leads to significant geometric errors in Gaussian estimation due to domain discrepancy (see "FreeSplatter w/o recentering" in Fig. 2). To leverage the priors learned from favorable views, we recenter each input image by shifting the foreground region to the image center and resizing it ("Recentered" in Fig. 2). Formally, we define the recentered image as $\bar{\mathbf{I}}_i^c = \mathcal{R}_{\theta_i}(\mathbf{I}_i^c)$, where $\mathcal{R}$ denotes the recentering function, and $\theta_i \in \mathbb{R}^4$ represents the bounding box coordinates, specifically, the pixel positions of the top-left and bottom-right corners. The foreground is resized such that $\max(H_f, W_f) = 0.8\min(H, W)$, where H_f and W_f are the height and width of the resized foreground bounding box, respectively. This enables the model to exploit the priors from favorable views and facilitates the Gaussian alignment described in Section 2.4.

We also recenter the 2D positional embedding $\mathbf{e} \in \mathbb{R}^{H/p\times W/p\times D}$, used in subsequent modules, as $\bar{\mathbf{e}}_i = \mathcal{R}_{\theta_i/p}(\mathbf{e})$, where p is a patch size and D is a channel dimension. This recentered embedding reflects the original pixel locations of the foreground region after recentering and is used by the adapter module.

2.3.2 Adapting to unfavorable views

We adopt FreeSplatter [10], originally trained on the Objaverse dataset [5], as the pose-free feed-forward 3DGS backbone. To adapt this model to unfavorable views, we insert LoRA layers [7] into the patchifying convolution layer and all linear layers within the self-attention blocks (Fig. 3). This allows the model to learn to process recentered images without full model retraining.

2.3.3 Adapter for geometric accuracy

Although the LoRA-adapted model improves cross-view consistency, the geometry of the Gaussians derived from recentered images remains inaccurate. This is because feed-forward models predict pixel-aligned Gaussians, and recentering distorts pixel-to-ray correspondence as shown in Fig.

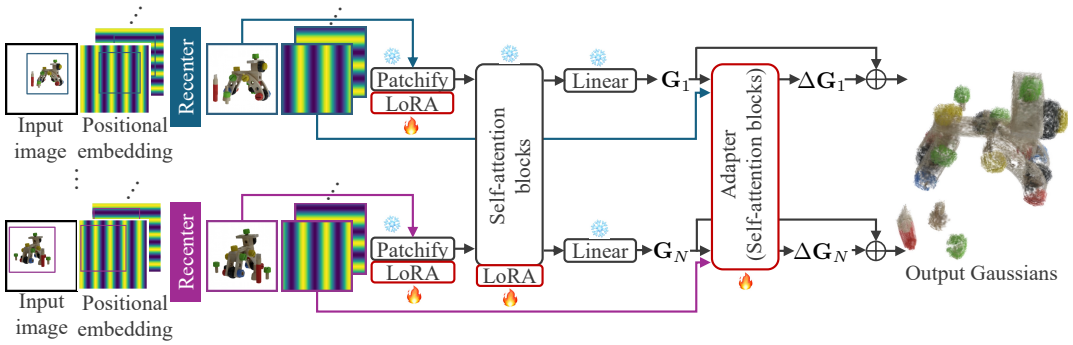


Fig. 3 Overview of the proposed model. It is built upon a pretrained pose-free feed-forward 3DGS model, augmented with LoRA layers. The model processes recentered input images to leverage the priors learned from favorable views. An additional adapter module corrects geometric errors in the estimated Gaussians caused by the recentering operation, using the recentered 2D positional embeddings.

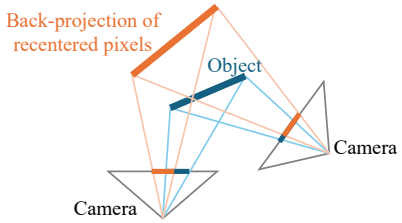


Fig. 4 Example of geometric distortion due to recentering. The rotation of the object obtained by back-projection of recentered pixels (orange region on image planes) is different from that of the original object.

4.

However, it is difficult to model such distortion directly. To address this issue, we propose an adapter module that refines the initial Gaussians by predicting residuals $\Delta\mathbf{G}_i$, which are added to the original Gaussians to produce the refined output $\widetilde{\mathbf{G}}_i$. The adapter module also takes the recentered positional embedding $\overline{\mathbf{e}}_i$ to encode original spatial context.

2.4 Gaussian alignment to render target views.

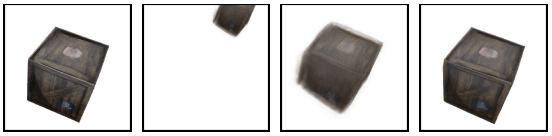
To compute loss, we render target images from the estimated Gaussians. However, due to pose-free input and recentering that breaks pixel-ray relationship, the predicted Gaussians no longer match the camera settings of the target view (Fig. 5(b)).

To address this, we propose a simple yet effective alignment of the refined 3D Gaussians $\{\widetilde{\mathbf{G}}_i\}_{i=1}^N$. Specifically, we estimate a scale factor $a \in \mathbb{R}$ and a translation vector $\mathbf{b} \in \mathbb{R}^3$ that minimize the weighted least squares distance between the refined Gaussian centers and the shifted point clouds:

$$\min_{a,\mathbf{b}} \sum_{i=1}^{N} \sum_{q\in\Omega} \mathbf{W}_i^c(q) \left\| a\widetilde{\boldsymbol{\mu}}_i(q) + \mathbf{b} - \overline{\mathbf{P}}_i^c(q) \right\|^2, \tag{2}$$

$$\Omega = [1, W] \times [1, H], \tag{3}$$

where $\widetilde{\boldsymbol{\mu}}_i(q) \in \mathbb{R}^3$ is the center of the refined Gaussian at pixel q, and $\overline{\mathbf{P}}_i^c(q)$ is the corresponding 3D point. Here, $\overline{\mathbf{P}}_i^c(q)$ is obtained by applying the recentering operations to the ground-truth point map $\mathbf{P}_i^c$ similar to the input image $\overline{\mathbf{I}}_i^c$ as $\overline{\mathbf{P}}_i^c = \mathcal{R}_{\theta_i}(\mathbf{P}_i^c)$, thereby ensuring pixel-level correspondence between $\widetilde{\boldsymbol{\mu}}_i(q)$ and $\overline{\mathbf{P}}_i^c(q)$. $\mathbf{W}_i^c$ is a weight, which is obtained as an opacity map during our dataset creation described in Section 3.1, to focus the alignment on foreground pixels. This weighted least squares problem can be efficiently solved in closed form during training. The resulting transformation is applied to both the Gaussian centers and scales, enabling accurate rendering and effective supervision, as illustrated in Figs. 5(c) and (d).



(a) (b) (c) (d)

Fig. 5 Effect of Gaussian alignment during training. (a) Ground-truth target image. (b) Rendering from unaligned 3D Gaussians at the beginning of training. (c) Rendering after applying the proposed alignment method. (d) Rendering from aligned 3D Gaussians after training.

3. Experiments

3.1 Dataset

For training our method, we use the G-Objaverse dataset [17], a high-quality subset comprising 280K assets with multi-view images, where each scene is rendered from 38 favorable views. For training our model, we apply FreeSplatter to all scenes to estimate 3D Gaussians, and render unfavorable views with the estimated Gaussians. We also render opacity maps, which are used for the foreground weights decribed in Section 2.4.

For evaluation, we use the Google Scanned Objects (GSO) dataset [6] and the OmniObject3D dataset [9]. From GSO, we select 50 scenes and render 32 unfavorable views, where four views are used as context inputs, and the remaining 28 as target views. For OmniObject3D, we follow the preprocessed subset in [11], selecting 20 scenes with provided foreground masks. The same four sparse views used in [11] are adopted as inputs, and 28 additional views are used as targets.

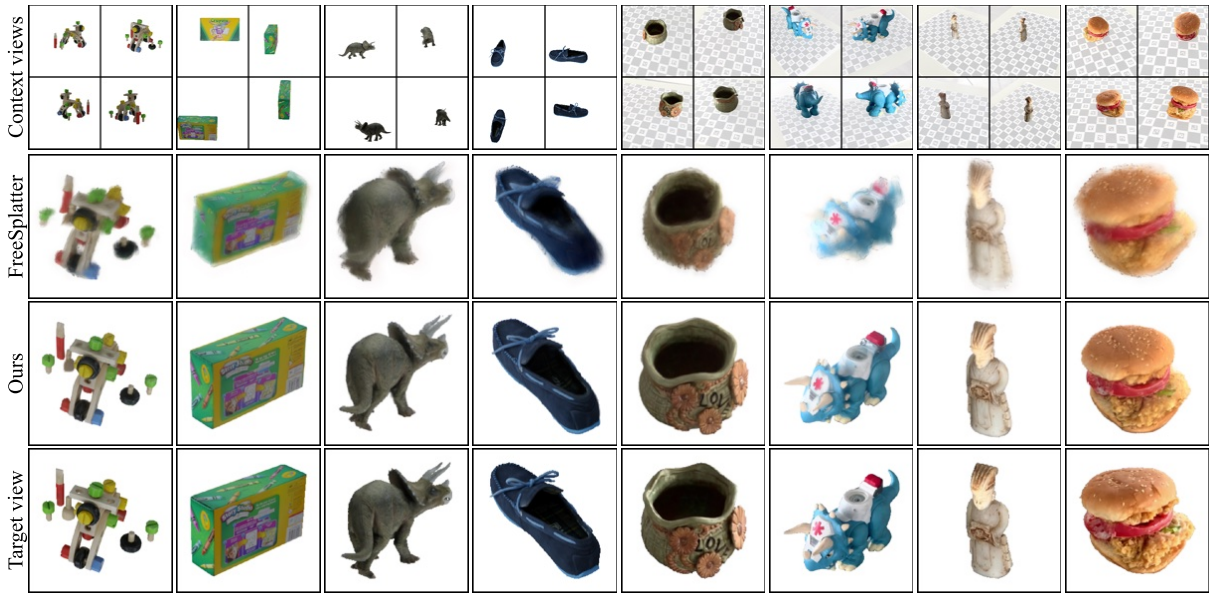


Fig. 6 Results of novel-view synthesis on the GSO and OmniObject3D dataset

Table 1 Quantitative comparison in novel-view synthesis on the GSO and OmniObject3D dataset. RC means "recentering."

	GSO		OmniObject3D	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
FreeSplatter w/o RC	12.475	0.265	12.145	0.288
FreeSplatter w/ RC	16.698	0.232	15.807	0.245
Ours	**22.910**	**0.110**	**18.861**	**0.182**

3.2 Implementation details

During training, four context views with resolution 512 × 512 are used to estimate 3D Gaussians, and four target views are rendered to compute the training loss with a pixel-wise reconstruction loss (MSE) and a perceptual loss (LPIPS [15]), weighted by 0.5 in all experiments. Training is run on four NVIDIA A100 GPUs

3.3 Experimental results

Figure 6 presents qualitative comparisons on GSO and OmniObject3D datasets. Each sample includes the context views, ground-truth target, and outputs from FreeSplatter (w/ recentering), and our method. The results show that FreeSplatter yields blurry outputs due to inconsistencies in Gaussian predictions across views. In contrast, our method effectively estimates Gaussians while maintaining multi-view consistency through adaptation, resulting in high-quality novel-view renderings.

Table 1 summarizes the quantitative results. FreeSplatter w/o recentering performs poorly under unfavorable views, and recentering alone does not fully address this issue. In contrast, our method significantly outperforms these baselines across all metrics, demonstrating effective adaptation to challenging viewpoints.

3.4 Ablation study on adaptation components

We evaluate each component in our adaptation framework (Tab. 2). The LoRA layers and the adapter each contribute to performance improvements. Adding the recentered positional embedding further improves geometric consistency, as illustrated in Fig. 7.

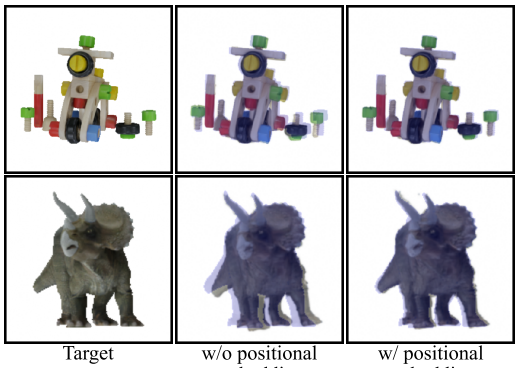


Fig. 7 Target image and rendered novel-view images without and with positional embedding. Ground-truth targets are overlaid in blue. Positional embedding improves alignment and geometry fidelity.

Table 2 Ablation study on each component for adaptation. "PE" denotes positional embedding.

LoRA	Adapter	PE	PSNR↑	SSIM↑	LPIPS↓
	✓	✓	19.749	0.853	0.165
✓			20.898	0.865	0.131
✓	✓		21.367	0.868	0.127
✓	✓	✓	**22.910**	**0.883**	**0.110**

4. Conclusion

In this paper, we proposed a method to adapt a pretrained pose-free feed-forward 3DGS model to handle unfavorable input views. Our method enhances the practical applicability of pose-free 3DGS models in real-world scenarios with unknown and varying camera poses.

Acknowledge This work was supported by JSPS KAKENHI Grant Number 26K02931.

References

[1] Charatan, D., Li, S. L., Tagliasacchi, A. and Sitzmann, V.: pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction, *CVPR*, pp. 19457–19467 (2024).

[2] Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J. and Su, H.: MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo, *ICCV*, pp. 14124–14133 (2021).

[3] Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.-J. and Cai, J.: MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images, *ECCV*, pp. 370–386 (2024).

[4] Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.-J. and Cai, J.: MVSplat360: Feed-Forward 360 Scene Synthesis from Sparse Views, *NeurIPS* (2024).

[5] Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A. and Farhadi, A.: Objaverse: A Universe of Annotated 3D Objects, *CVPR*, pp. 13142–13153 (2023).

[6] Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B. and Vanhoucke, V.: Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items, *ICRA*, p. 2553–2560 (2022).

[7] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models, *ICLR* (2022).

[8] Kerbl, B., Kopanas, G., Leimkuehler, T. and Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering, *ACM TOG*, Vol. 42, No. 4 (2023).

[9] Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., Lin, D. and Liu, Z.: OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation, *CVPR*, pp. 803–814 (2023).

[10] Xu, J., Gao, S. and Shan, Y.: FreeSplatter: Pose-free Gaussian Splatting for Sparse-view 3D Reconstruction, *ICCV*, pp. 25442–25452 (2025).

[11] Yang, C., Li, S., Fang, J., Liang, R., Xie, L., Zhang, X., Shen, W. and Tian, Q.: GaussianObject: High-Quality 3D Object Reconstruction from Four Views with Gaussian Splatting, *ACM TOG*, Vol. 43, No. 6 (2024).

[12] Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.-H. and Peng, S.: No Pose, No Problem: Surprisingly Simple 3D Gaussian Splats from Sparse Unposed Images, *ICLR* (2025).

[13] Yu, A., Ye, V., Tancik, M. and Kanazawa, A.: pixelNeRF: Neural Radiance Fields From One or Few Images, *CVPR*, pp. 4578–4587 (2021).

[14] Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K. and Xu, Z.: GS-LRM: Large Reconstruction Model for 3D Gaussian Splatting, *ECCV*, pp. 1–19 (2024).

[15] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. and Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, *CVPR* (2018).

[16] Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y. and Wetzstein, G.: FLARE: Feed-forward Geometry, Appearance and Camera Estimation from Uncalibrated Sparse Views, *CVPR*, pp. 21936–21947 (2025).

[17] Zuo, Q., Gu, X., Dong, Y., Zhao, Z., Yuan, W., Qiu, L., Bo, L. and Dong, Z.: High-Fidelity 3D Textured Shapes Generation by Sparse Encoding and Adversarial Decoding, *ECCV*, pp. 52–69 (2024).