# Revisiting Depth Representations for Feed-Forward 3D Gaussian Splatting

**Duochao Shi**[1*], **Weijie Wang**[1, 4*], **Donny Y. Chen**[2], **Zeyu Zhang**[2, 4], **Jia-Wang Bian**[3], **Bohan Zhuang**[1]

[1]Zhejiang University, China [2]Monash University, Australia [3]MBZUAI [4]GigaAI

## Abstract

Depth maps are widely used in feed-forward 3D Gaussian Splatting (3DGS) pipelines by unprojecting them into 3D point clouds for novel view synthesis. This approach offers advantages such as efficient training, the use of known camera poses, and accurate geometry estimation. However, depth discontinuities, which are particularly problematic at the boundaries of the reconstructed geometry, often lead to fragmented or sparse point clouds, degrading rendering quality—a well-known limitation of depth-based representations. To tackle this issue, we introduce **PM-Loss**, a novel regularization loss based on a pointmap predicted by a pre-trained transformer. Although the pointmap itself may be less accurate than the depth map, it provides a powerful prior for geometric coherence and structural completeness, especially at the very edges where depth prediction falters. With the improved depth map, our method significantly improves the feed-forward 3DGS across various architectures and scenes, delivering consistently better rendering results.

## 1 Introduction

Novel view synthesis (NVS) has been significantly advanced by 3D Gaussian Splatting (3DGS) [1]. Despite its ultra-fast rendering, 3DGS requires time-consuming per-scene optimization, which has led to the development of feed-forward 3DGS methods [2, 3], the focus of our work.

The core issue with current feed-forward methods lies in their reliance on *depth maps*. Most models predict depth maps and then unproject them to form 3D Gaussians. Since depth maps often contain discontinuities, especially near boundaries [4–6], these artifacts are transferred to the 3D representation. Neural networks often fail to predict sharp depth steps, producing erroneous values at these boundaries. When unprojected, these inaccuracies manifest as fragmented floaters in space or sparse gaps along edges, degrading geometric quality.

Recently, 3D reconstruction has seen success with a representation known as the pointmap [7–13]. Unlike depth maps, pointmaps encode a set of 3D points $p \in \mathbb{R}^3$ in world space, allowing for more structurally coherent and complete modeling of geometry. This motivates us to introduce pointmaps as a strong prior to reduce artifacts in depth-based feed-forward 3DGS.

We propose a novel method to distill the geometry prior from a pointmap regression model via a simple yet effective training loss. Our PM-Loss guides the learning of point clouds unprojected from predicted depth by taking the global pointmap predicted by a large-scale 3D reconstruction model (*e.g.*, VGGT [11]) as a pseudo-ground truth. We leverage the one-to-one correspondence between depth maps and pointmaps to efficiently align the two point clouds using the Umeyama algorithm[14]. Then, the Chamfer loss is used to directly regularize them in 3D space. By distilling this geometric

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

prior, our method mitigates discontinuities from unprojected depth and significantly boosts the quality of predicted 3D point clouds and rendered novel views.

## 2 Related Work

Feed-forward 3DGS methods [2, 3, 15–23] have accelerated novel view synthesis, often improving geometry by incorporating depth priors [24]. However, these priors, typically from monocular depth estimators, can suffer from multi-view inconsistencies, leading to geometric inaccuracies. Concurrently, 3D reconstruction has seen a surge in pointmap-based methods [7–13], which excel at producing accurate and coherent 3D point clouds directly from images. While these pointmap models provide strong geometric priors, they are not optimized for direct novel view synthesis. Our work bridges this gap by distilling the robust geometric prior from pre-trained pointmap models into feed-forward 3DGS frameworks via a novel training loss, directly addressing the limitations of depth-based geometry without the high cost of retraining for rendering.

## 3 Methodology

Our goal is to train a network that directly predicts a 3DGS model from input images. We introduce PM-Loss to regularize the predicted 3D structure using a pointmap prior.

### 3.1 Preliminary

**Feed-Forward 3DGS.** In typical feed-forward 3DGS pipelines, Gaussian means $\mu_i$ are derived by unprojecting predicted depth maps. For each pixel $(u, v)$, a depth value $d(u, v)$ is predicted and used with camera parameters $(K, R_{\text{ext}}, t_{\text{ext}})$ to compute the 3D position:

$$\mu_{uv} = R_{\text{ext}} \cdot (d(u, v) \cdot K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}) + t_{\text{ext}} \tag{1}$$

This approach is efficient but suffers from geometric inaccuracies due to depth discontinuities, leading to artifacts like floaters and sparse geometry, as seen in Fig. 1.

**Pointmap Regression.** A pointmap is a structured 3D representation where each pixel $(u, v)$ of an image is associated with a 3D point $p'_{uv} \in \mathbb{R}^3$ in world coordinates. They are typically regressed from images using pretrained Vision Transformer (ViT) based models.

### 3.2 PM-Loss

We advocate for directly regularizing geometry in 3D space. Given a batch of input images, our feed-forward 3DGS model predicts a set of Gaussian centers, $X_{\text{3DGS}}$. For supervision, we use a reference point cloud, $X_{\text{PM}}$, from a pretrained pointmap model. Both point clouds share a natural one-to-one correspondence since each point pair originates from the same source pixel. While $X_{\text{PM}}$ may be less accurate in well-textured regions, it exhibits better geometric coherence and structural completeness, especially at boundaries.

**Efficient Point Cloud Alignment.** Although both point clouds are in world coordinates, they are often misaligned. The one-to-one correspondence enables the use of the highly efficient Umeyama algorithm [14] to find the optimal similarity transformation $(s^*, R^*, t^*)$ that minimizes their mean squared error. We apply this transformation to the source pointmap $X_{\text{PM}}$ to obtain the aligned pointmap $X'_{\text{PM}}$.

**Single-Directional Chamfer Loss.** Given the aligned point clouds, we define the PM-Loss $L_{\text{PM}}$ as a single-directional Chamfer distance from $X_{\text{3DGS}}$ to $X'_{\text{PM}}$:

$$L_{\text{PM}}(X_{\text{3DGS}}, X'_{\text{PM}}) = \frac{1}{N} \sum_{\mu \in X_{\text{3DGS}}} \min_{p' \in X'_{\text{PM}}} \|\mu - p'\|_2^2 \tag{2}$$

This regularizes the predicted Gaussian centers towards the geometry prior. A key insight is to re-compute the nearest neighbor in 3D space for supervision, rather than relying on the one-to-one pixel correspondence, which would degenerate to a 2D depth loss. This design is more robust to pose misalignments and prediction noise.

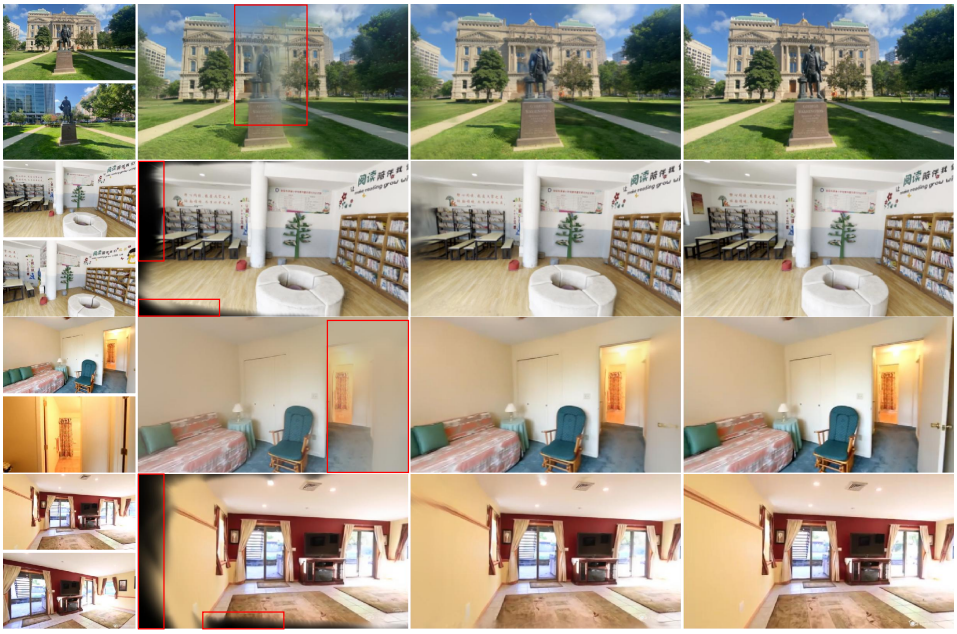



Figure 1: **Qualitative comparisons on DL3DV (top two rows) and RealEstate10K (bottom two rows).** Adding PM-Loss leads to significant improvements in rendering quality at boundaries. Note the mitigation of blurry artifacts (row 1,3) and black regions (row 2,4) in the rendered views.

# 4 Experiments

## 4.1 Experimental Settings

**Datasets and Baselines.** We evaluated on DL3DV [25], RealEstate10K [26], and DTU [27]. We apply PM-Loss to two representative feed-forward 3DGS models: MVSplat [3] and DepthSplat [24].

**Metrics.** For Novel View Synthesis (NVS), we use a boundary-aware setting where target views are selected to lie adjacent to the spatial region of context views, making geometric boundaries visible. We report PSNR, SSIM, and LPIPS. For geometric quality, we treat Gaussian centers as a point cloud and compare against DTU ground truth using Accuracy (Acc), Completeness (Comp), and Overall Chamfer Distance.

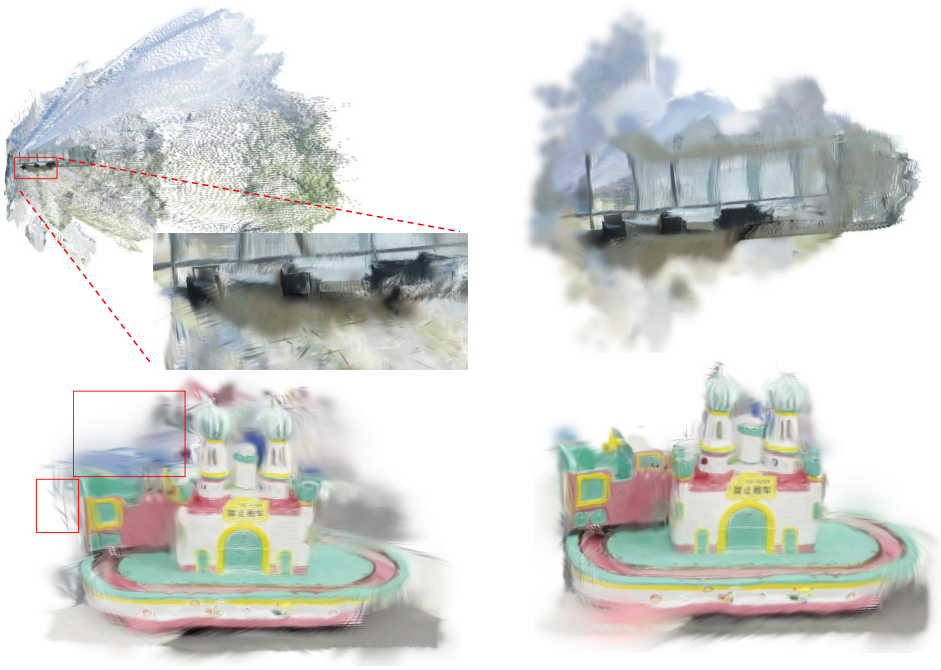
**Implementation Details.** We used PyTorch and PyTorch3D [28]. Models were fine-tuned for 100k iterations with a learning rate of $2 \times 10^{-4}$. We used the VGGT-1B [11] model to generate pointmaps. The loss weight $\lambda_{PM}$ was set to $0.005$.

Table 1: **Quantitative results in the boundary-aware setting.** Both MVSplat and DepthSplat show better rendering quality with the addition of PM-Loss.

| Method | DL3DV | | | RealEstate10K | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| DepthSplat | 18.46 | 0.689 | 0.261 | 20.43 | 0.788 | 0.218 |
| DepthSplat+PM | **20.77** | **0.705** | **0.245** | **22.48** | **0.814** | **0.194** |
| MVSplat | 16.79 | 0.592 | 0.322 | 19.52 | 0.757 | 0.231 |
| MVSplat+PM | **19.25** | **0.615** | **0.291** | **22.18** | **0.787** | **0.199** |

## 4.2 Comparisons and Analysis

**Visual and Point Cloud Quality.** By regularizing the predicted point clouds, our PM-Loss improves 3D Gaussian quality and novel view rendering. Tab. 1 shows our method boosts baseline performance



DepthSplat DepthSplat + PM-Loss

Figure 2: **Qualitative comparison of 3D Gaussians on DL3DV.** Our method effectively regularizes the 3D Gaussians, reducing floating artifacts and noise.

Table 2: **Quantitative comparison on DTU with varying input numbers.** Adding PM-Loss consistently improves geometry across different numbers of input views.

| Input | Method | Acc↓ | | Comp↓ | | Overall↓ | |
|---|---|---|---|---|---|---|---|
| | | Mean | Med. | Mean | Med. | Mean | Med. |
| 2-view | DepthSplat | 0.264 | 0.200 | 0.101 | 0.051 | 0.182 | 0.125 |
| | DepthSplat+PM | **0.232** | **0.166** | **0.099** | **0.045** | **0.165** | **0.106** |
| 4-view | DepthSplat | 0.169 | 0.117 | **0.066** | 0.022 | 0.123 | 0.051 |
| | DepthSplat+PM | **0.156** | **0.076** | 0.069 | **0.022** | **0.113** | **0.049** |
| 6-view | DepthSplat | 0.162 | 0.070 | **0.048** | 0.017 | 0.105 | 0.044 |
| | DepthSplat+PM | **0.150** | **0.068** | 0.053 | **0.016** | **0.102** | **0.042** |

by over 2 dB in PSNR. This improvement stems from enhanced geometric coherence, which is particularly effective in addressing errors from depth discontinuities. As supported by Fig. 1, the baseline's failure to handle these discontinuities leads to blurry artifacts and black regions, which our method mitigates.

We qualitatively compare point cloud quality in Fig. 2, where our method produces cleaner results with fewer noisy artifacts. For quantitative analysis, we evaluate on the DTU benchmark. As shown in Tab. 2, our method improves accuracy, completeness, and overall scores, confirming our qualitative findings. These improvements are consistent across varying numbers of input views.

## 5 Conclusion

We presented PM-Loss, a simple yet effective training loss that leverages geometry priors from pointmaps to improve feed-forward 3DGS. By regularizing in 3D space, PM-Loss alleviates depth-induced discontinuities, leading to significantly improved geometry and rendering quality. Our method can be seamlessly integrated into existing training pipelines with minimal overhead and introduces no inference cost. Extensive experiments demonstrate its broad applicability and efficiency.

## References


[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[2] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, pages 19457–19467, 2024.

[3] 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*, pages 370–386. Springer, 2024.

[4] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *CVPR*, pages 14648–14657, 2020.

[5] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 46(1):497–508, 2023.

[6] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

[7] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024.

[8] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025.

[9] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *CVPR*, 2025.

[10] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[11] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

[12] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025.

[13] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024.

[14] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[15] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *NeurIPS*, 37:107326–107349, 2024.

[16] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images, 2024.

[17] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *ECCV*, 2024.

[18] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat++: Generalizable 3d gaussian splatting for efficient indoor scene reconstruction, 2025.

[19] Zhiyuan Min, Yawei Luo, Jianwen Sun, and Yi Yang. Epipolar-free 3d gaussian splatting for generalizable novel view synthesis, 2024.

[20] Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views, 2024.

[21] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *CVPR*, 2025.

[22] Weijie Wang, Donny Y. Chen, Zeyu Zhang, Duochao Shi, Akide Liu, and Bohan Zhuang. Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs, 2025.

[23] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *NeurIPS*, 37:107064–107086, 2024.

[24] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025.

[25] 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*, pages 22160–22169, 2024.

[26] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[27] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, pages 406–413, 2014.

[28] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d, 2020.