

A. More Experimental Analysis

Evaluation on the DepthSplat Official Evaluation Set. Beyond the main paper’s boundary-aware setting, we also tested our PM-Loss using the DepthSplat’s [2] official evaluation set on the DL3DV dataset, with two input views. Tab. A shows the quantitative results for this setup. Applying our PM-Loss (denoted as ‘+PM’) consistently improves novel view synthesis for both DepthSplat [2] and MVSplat [1] across all metrics. This demonstrates that our distilled geometric priors are also beneficial under this evaluation setting.

Despite these positive results, our main paper primarily focuses on the boundary-aware setting. The boundary-aware setting is designed to more rigorously test the geometric integrity of 3D reconstructions, especially near the periphery of the scene. This focus helps to clearly quantify and showcase improvements in 3D Gaussian geometric quality, a key contribution of our PM-Loss.

Table A. **NVS on DL3DV with DepthSplat [2]’s official eval set.** Our PM-Loss (+PM) improves rendering for both MVSplat [1] and DepthSplat [2].

Method	View Interpolation		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
DepthSplat	19.05	0.610	0.313
DepthSplat+PM	19.86	0.641	0.298
MVSplat	17.59	0.514	0.396
MVSplat+PM	17.81	0.520	0.380

Discussion on Online Pointmap Processing The main paper presents our PM-Loss’s VRAM usage under the recommended offline preprocessing workflow, which is highly memory-efficient. For added convenience, however, pointmaps can also be generated online during training. We analyze this trade-off here. Tab. B details the VRAM footprint for both processing strategies. While the offline process adds a modest 0.96GB, the online process increases VRAM consumption by 6.09GB over the baseline (57.88GB vs. 51.79GB). This larger increase is primarily due to the significant memory required to load the pointmap generation model (e.g., VGGT-1B) onto the GPU during training. Therefore, while the online workflow offers more convenience, preprocessing pointmaps offline remains the most memory-efficient approach for using our PM-Loss.

Discussion of Existing Pointmap Based Feed-Forward 3DGS. Integrating pointmap priors into feed-forward 3DGS faces challenges from pose discrepancies (Discussed in Sec. 1). Some methods, like NoPoSplat [3], use specialized “Gaussian heads” for pointmap features. However, this approach has trade-offs, as shown in Tab. C: NoPoSplat has

Table B. **VRAM footprint of different pointmap processing strategies.** Comparison of GPU memory for baseline DepthSplat, and DepthSplat with PM-Loss (offline vs. online VGGT-1B pointmap processing) during training.

Method	Max VRAM (GB)
DepthSplat	51.79
DepthSplat+PM (offline process)	52.75
DepthSplat+PM (online process)	57.88

a significantly larger model (611M parameters vs. MVSplat’s [1] 12M), and its rendering quality without test-time pose alignment (PSNR 24.70) is notably below MVSplat’s (PSNR 26.39). While NoPoSplat achieves its best quality (PSNR 26.79) with test-time pose alignment, this step is slow, averaging 2.84 seconds per instance, thereby limiting real-time applicability.

NoPoSplat’s case illustrates that specialized heads for pointmap integration can lead to large models and require costly test-time optimizations for optimal performance. Motivated by these limitations, our PM-Loss instead distills pointmap priors via an efficient training loss, aiming for a better balance between rendering quality, model size, and inference speed.

B. Limitation and Societal Impacts

Limitation: Dependence on Pointmap Quality. As stated in the main paper, our PM-Loss’s effectiveness is limited by the quality of the pre-trained pointmap model providing geometric priors. Errors in this pointmap can propagate via our PM-Loss, impacting the final 3DGS model’s geometric quality. Fig. A illustrates this dependency. The left side shows a VGGT pointmap with inaccuracies, while the right displays DepthSplat + PM-Loss’s 3D Gaussians where these prior errors are visibly reflected. Thus, poor pointmap quality in challenging regions constrains our PM-Loss’s performance. Leveraging improved pointmap models in the future could further enhance our approach.

Broader Societal Impacts. Our work’s improvement to 3D shape quality presents both opportunities and ethical considerations. Benefits include enhanced AR/VR experiences, accelerated content creation, and cultural heritage preservation. However, the heightened realism risks misuse for creating believable fake content, which could fuel propaganda or scams and erode digital trust. Furthermore, the synthetic nature of these models warrants caution in safety-critical applications, such as autonomous vehicle training, due to potential inaccuracies.

Table C. **Quantitative comparison of NoPoSplat’s variants and MVSplat.** Key metrics, model size, and inference time for NoPoSplat (with/without pose alignment) compared to the MVSplat baseline.

Method	Rendering Quality			Parameters (M)↓	Time (ms)↓
	PSNR↑	SSIM↑	LPIPS↓		
NoPoSplat (w/o pose align)	24.70	0.818	0.145	611	2.8
NoPoSplat (w/ pose align)	26.79	0.878	0.124	611	2840
MVSplat	26.39	0.869	0.128	12	1.4



Figure A. **Limitation: Pointmap Errors Propagate to 3DGS.** Left: VGGT pointmap with inaccuracies (e.g., for sky regions). Right: DepthSplat + PM-Loss’s 3D Gaussians from a failure case, showing propagated errors from the pointmap prior.

C. More Implementation Details

More Baseline Details. For our experiments involving DepthSplat [2], it is important to note the specific version used. The results presented in this paper are based on the initial version of DepthSplat, which was publicly released in October 2024. We acknowledge that an updated version of the DepthSplat model architecture was made available in March 2025. However, all experiments reported herein were conducted using the aforementioned October 2024 release to maintain consistency with the experimental timeline of our work.

More Loss Function Details. Our total training loss, denoted as L_{total} , consists of two primary components: a rendering loss term, L_{render} , and our proposed geometric consistency loss, L_{PM} . The detailed formulation of L_{PM} itself is provided in the main paper (Sec. 3.2). The overall loss is defined as:

$$L_{\text{total}} = L_{\text{render}} + \lambda_{\text{PM}} L_{\text{PM}} \quad (1)$$

Here, λ_{PM} is the weighting coefficient for L_{PM} . As specified in the main paper, we set $\lambda_{\text{PM}} = 0.005$.

The rendering loss, L_{render} , follows common practices established in prior works [1, 2]. It combines a Mean Squared Error (L_{MSE}) term and an LPIPS (L_{LPIPS}) perceptual loss

term:

$$L_{\text{render}} = \lambda_{\text{MSE}} L_{\text{MSE}} + \lambda_{\text{LPIPS}} L_{\text{LPIPS}} \quad (2)$$

The L_{MSE} term quantifies the pixel-wise squared differences between the rendered output and the ground-truth image. The L_{LPIPS} term assesses perceptual similarity between the rendered and ground-truth images. We set the respective weights for these components as $\lambda_{\text{MSE}} = 1.0$ and $\lambda_{\text{LPIPS}} = 0.05$.

More Training Details. We fine-tune all baseline models on the DL3DV dataset, starting from their publicly available pre-trained weights. To fairly evaluate our PM-Loss, we prepare two versions for each baseline: one fine-tuned with PM-Loss integrated, and a control version fine-tuned strictly following the baseline’s original methodology. This isolates the impact of PM-Loss. While largely adhering to original training configurations, minor adjustments were made for single GPU fine-tuning. For DepthSplat [2], we use a batch size of 1, with the number of context views randomly chosen from 2-6 and target views fixed at 4 during training. For MVsplat [1], the batch size is 12, with 2 fixed context views and 4 fixed target views. Code and pre-trained weights will be made available.



Figure B. **More comparisons on DL3DV(top two rows) and RealEstate10K(bottom four rows) under the 2-view extrapolation setting.** Adding PM-Loss leads to significant improvements in rendering quality at boundaries. Note the mitigation of black regions(row 1,3,4) and blurry artifacts(row 2,5,6) in the rendered views.

More Testing Details. Here we provide further details on our Novel View Synthesis (NVS) boundary-aware evaluation protocol, introduced in the main paper (Sec. 4.1). In this setting, we use two context views and select target views from immediately before and after them in the image

sequence. This sampling strategy is designed to position the camera in a way that makes the geometric boundaries of the reconstructed scene visible, thereby rigorously assessing the rendering integrity at the scene’s periphery. Our specific sampling strategy, based on image sequence indices, adapts

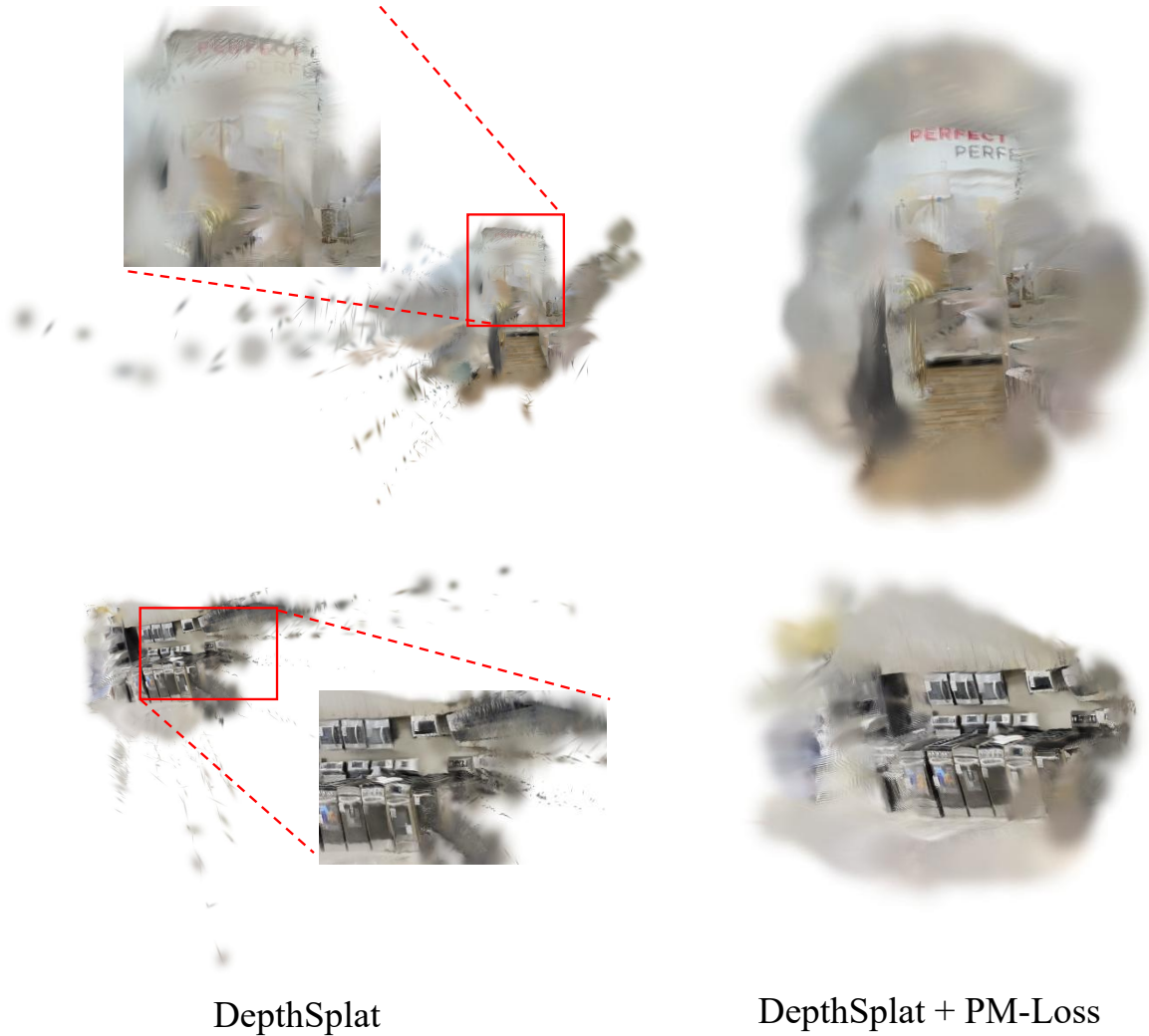


Figure C. **More comparison of 3D Gaussians on DL3DV dataset.** PM-Loss effectively regularizes the 3D Gaussians, significantly reducing floating artifacts and noise that arise from inaccurate depth predictions.

to dataset characteristics. For the RealEstate10K dataset, context views are separated by 75 image indices; we then sample 15 target views before the first context view and 15 after the second (30 total target views). For the DL3DV dataset, due to its larger inter-frame viewpoint changes, we use a context gap of 20 indices, sampling 10 target views before the first and 10 after the second context view (20 total target views).

D. More Visual Comparisons

This section presents supplementary visual comparisons to further demonstrate the impact of our PM-Loss. Fig. B showcases additional novel view synthesis results from the DL3DV and RealEstate10K datasets, highlighting the improved rendering of boundaries when our PM-Loss is ap-

plied. Furthermore, Fig. C visualizes the quality of the reconstructed 3D Gaussians on the DL3DV dataset, illustrating the reduction in floating artifacts and noise of the reconstruction achieved by our approach.

References

- [1] 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. [1](#), [2](#)
- [2] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depth-splat: Connecting gaussian splatting and depth. In *CVPR*, 2025. [1](#), [2](#)
- [3] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *ICLR*, 2025. [1](#)