AffineGS: Unlocking Accurate Geometry in 3DGS via Spatially Varying Affine Rectification

Tze Ho Elden Tse[1,2*] **Jizong Peng**[1*] **Richard Chen**[1] **Angela Yao**[2]
[1]dConstruct Robotics [2]National University of Singapore
https://eldentse.github.io/AffineGS/

Abstract

3D Gaussian Splatting (3DGS) has become the leading approach for photorealistic novel view synthesis, yet its underlying geometric accuracy lags behind its visual quality. Existing depth-guided methods attempt to resolve this by regularizing the optimization with monocular depth priors, typically relying on a global linear alignment. Crucially, we reveal a fundamental flaw in this assumption: modern monocular depth networks systematically exhibit region-dependent, piecewise-linear distortions. This phenomenon invalidates standard global alignment strategies, which inevitably warp scene geometry by enforcing a single scale-and-shift transform. To address this, we introduce a spatially varying affine rectification model that dynamically corrects monocular disparity into consistent metric depth using regionally smooth, edge-aware fields. We further derive a normal-aware regularizer that uses spatial gradients to couple the rectified disparity with 3DGS depth and normals. Extensive evaluations demonstrate that our sensor-free approach prevents structural collapse under extreme view sparsity and extracts highly accurate 3D meshes, effectively bridging the gap toward sensor-supervised geometric fidelity while preserving state-of-the-art photorealism.

1 Introduction

3D Gaussian Splatting (3DGS) [12] has emerged a preferred method for fast and photorealistic rendering of novel views. However, while 3DGS excels at capturing high-frequency appearance details, its underlying geometric representation remains inherently unstructured. Compared to implicit neural surfaces [31, 15] or classical Multi-View Stereo, the geometry reconstructed by 3DGS is significantly weaker. As noted in [7, 5, 24], Gaussian primitives often overfit to the training views, leading to floating artifacts, multi-layered surfaces, and locally inconsistent geometry. This lack of structural integrity is highly problematic for downstream applications requiring reliable 3D meshes, such as autonomous driving, robotic perception, and AR/VR.

To address these geometric shortcomings, one line of work leverages physical sensor depth (*e.g.,* DN-Splatter [24]) to directly constrain and align the Gaussian primitives. However, acquiring dense, high-quality sensor depth is hardware-dependent and impractical in many real-world applications. As a more scalable alternative, a highly promising line of research [5, 26] injects monocular depth priors into the optimization process, leveraging the strong structural cues produced by modern foundation depth models. These methods typically operate on a critical assumption: that predicted monocular disparity can be aligned to metric depth using a global linear transform. While this global alignment strategy successfully provides baseline geometric guidance, our extensive experimentation reveals that this core assumption breaks down in complex real-world environments.

Specifically, we discover that modern off-the-shelf monocular estimators (*e.g.,* Depth-Anything-V2 [27], MoGe [25], Metric3D-V2 [10]) systematically exhibit region-dependent, piecewise-linear

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

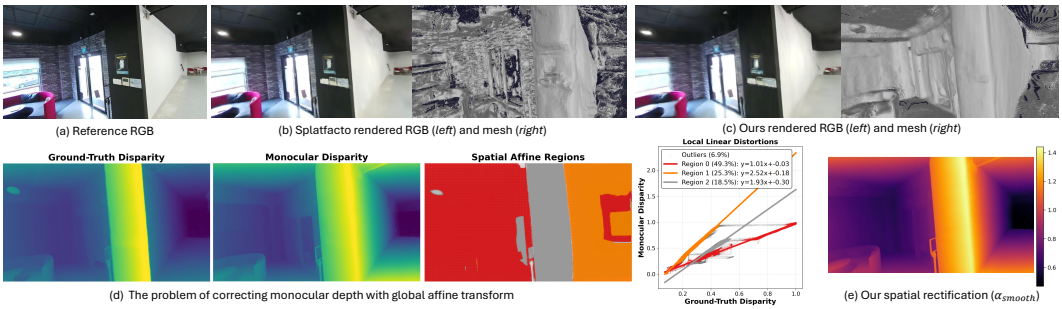


Figure 1: **Robust 3DGS geometry via spatially varying affine rectification.** (a-c) Standard 3DGS overfits appearance, producing noisy underlying geometry. In contrast, our method extracts clean, highly accurate meshes while maintaining photorealism. (d) Monocular disparity exhibits piecewise-linear distortions, proving that a single global scale-and-shift alignment is fundamentally insufficient (scatter plot). (e) We overcome this by estimating an edge-aware spatial scale field (α_{smooth}) to rectify the monocular depth prior, yielding geometrically consistent 3D representations.

distortions. As they are trained to predict relative depth, their outputs do not map consistently to metric depths. Our findings show that the models tend to fragment scenes into distinct geometric clusters. Fig. 1 shows an example where the predicted disparity fractures into multiple spatial regimes, with breaks along dominant object boundaries or large depth discontinuities. Consequently, attempting to fit a single global scale-and-shift transform will either average the error across these conflicting clusters—resulting in an entirely misaligned scene—or overfit to a single dominant region while severely warping the rest. This phenomenon fundamentally limits the efficacy of global alignment strategies, severely reducing their reliability for geometric supervision in sparse-view settings.

Motivated by this insight, we move beyond global assumptions by introducing a spatially varying affine rectification of monocular disparity. Our key observation is that monocular depth exhibits highly structured distortions: locally smooth, edge-aligned, and piecewise linear. We exploit this structure by dynamically estimating scale-and-shift fields. Crucially, we elegantly recover these spatially varying parameters without relying on any external sensor depth. Specifically, we estimate the local scale factor by comparing the finite-difference spatial gradients of the monocular disparity against the rendered 3DGS depth. We then refine these initial estimates using an edge-aware guided filter, which enforces smoothness within continuous surfaces while preserving sharp discontinuities across object boundaries. This self-contained process rectifies the monocular prior into consistent metric depth prior to geometric supervision. We further integrate this rectified depth with normal-aware constraints derived from projective camera geometry. By formulating our constraints directly on the spatial gradients, we isolate the scale estimation from local shift ambiguities and couple the structural boundaries of the disparity prior with the physical orientations of the 3DGS primitives.

In summary, our contributions are:

- We show that modern monocular depth predictions suffer from region-dependent, piecewise-linear distortions in disparity space which limits global alignment in depth-guided 3DGS, as a single transform cannot simultaneously regularize distinct geometric clusters.
- We present a novel affine rectification model that corrects monocular disparity into consistent metric depth using regionally smooth, edge-aware fields derived from spatial gradients.
- We introduce a gradient-domain regularizer derived from camera geometry that couples corrected disparity gradients with 3DGS depth and surface normals.
- Across diverse experimental settings, our method maintains the strong photometric quality of Gaussian Splatting while substantially improving geometric accuracy, bridging the gap toward sensor-supervised methods without requiring physical depth data.

2 Related Work

Novel View Synthesis (NVS). Classical multi-view geometry addressed 3D reconstruction with methods like Structure-from-Motion (SfM) and Multi-View Stereo (MVS) . Tools such as COLMAP [21] remain widely used, estimating camera poses and sparse 3D points through epipolar constraints before densifying geometry via stereo matching. While effective, these pipelines depend heavily on feature matching and struggle in low-texture or non-Lambertian regions.

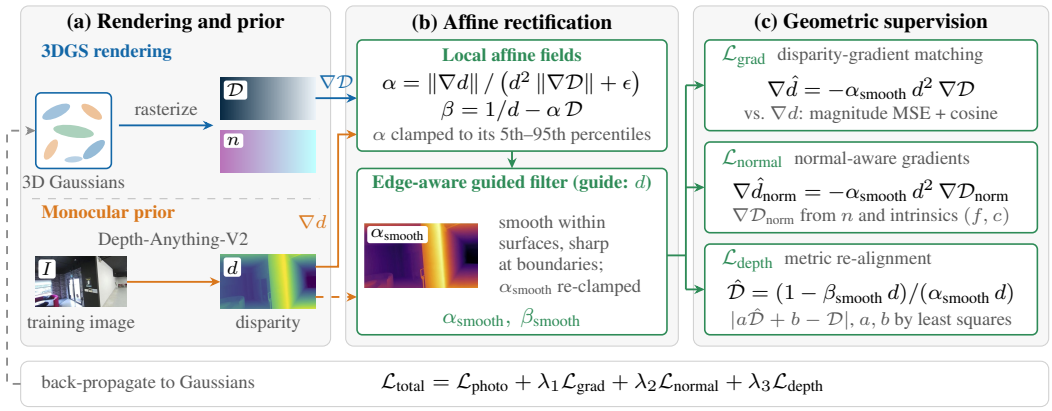


Figure 2: **Method overview.** (a) 3DGS renders depth $\mathcal{D}$ and normals n, and Depth-Anything-V2 predicts the disparity d. (b) The affine fields α and β are estimated from disparity and depth gradients and smoothed by a guided filter (Sec. 3.1). (c) They supervise the losses of Secs. 3.2 and 3.3. All fields are re-estimated at every iteration.

With the rise of deep learning, neural reconstruction methods have significantly advanced NVS. Implicit scene modeling methods like Neural Radiance Fields (NeRF) [15] represent scenes with a multilayer perceptron and synthesize images via volume rendering. NeRF delivers impressive photorealism and sparked extensive follow-up research tackling aliasing [1], efficiency [18, 29, 9], and scalability [19]. Parallel to these developments, explicit or hybrid representations, such as sparse voxel grids [13, 22], tensor decompositions [2], and polygonal primitives [4] have emerged to improve rendering speed by exploiting localized computation. Among the explicit representations, 3D Gaussian Splatting (3DGS) [12] stands out for enabling real-time rendering with high visual fidelity through alpha-blended rasterization of Gaussian primitives. Its combination of performance, simplicity, and quality has made it a leading representation for novel view synthesis. However, 3DGS's geometric accuracy remains limited and geometry improvement is an active research direction.

3DGS with Depth Priors. Recent works integrate monocular depth estimation [17] into 3DGS to strengthen geometric supervision. More advanced monocular estimators, such as Depth-Anything-V2 [27], broaden the range of depth priors available. 3DGS pipelines [5, 24] typically incorporate monocular depth by jointly optimizing a single global scale-and-shift transform during training. This dynamically aligns the predicted disparity to the rendered depth, allowing the aligned prediction to serve as a geometric regularization target. However, as we demonstrate in Section 3.1, this assumption of global linear alignment does not hold in practice due to the region-dependent, piecewise-linear distortions inherent in monocular depth predictions.

Beyond direct depth supervision, 3DGS extensions leverage geometric cues primarily by modifying the underlying representation. Recent works achieve this by structurally adjusting the primitives into 2D patches or disks [7, 11], appending surface normals as explicit attributes [6], merging Gaussian rendering with implicit signed-distance fields [3], or introducing joint normal-depth supervision pipelines [32, 24]. In contrast, our method does not modify the Gaussian representation nor rely on rigid global alignment assumptions. Instead, we extract reliable geometric supervision purely by targeting and rectifying the spatially varying distortions inherent to monocular depth priors.

3 Method

To improve the geometric fidelity of 3D Gaussian Splatting (3DGS), our method leverages monocular depth priors while explicitly accommodating their inherent spatial distortions. Modern monocular estimators exhibit region-dependent, piecewise-linear inconsistencies that cannot be resolved via standard global scale-and-shift alignment. Rather than treating these distortions as pure noise, we exploit their locally smooth and edge-aligned structure, and introduce a spatially varying affine rectification model (Section 3.1). We then couple these locally consistent metric depth signals with surface-normal and camera-geometry constraints in the gradient domain (Section 3.2), and anchor the overall scene scale using a direct metric depth re-alignment (Section 3.3). Together, these components provide robust, geometry-aware supervision without modifying the underlying 3DGS representation. Fig. 2 gives an overview of the full pipeline.

3.1 Spatially Varying Affine Rectification

Motivation. Existing depth-guided 3DGS methods typically assume that monocular disparity can be rectified into metric depth using a single global scale-and-shift transform. However, to quantify the validity of this assumption, we conducted a statistical analysis using the Depth-Anything-V2 prior on the MuSHRoom and ScanNet++ datasets. As shown in Table 1, a single global transform ($K = 1$) leaves over 40% of the scene fundamentally misaligned. Yet, when partitioned into just 3 to 5 distinct local regions, the inlier ratios jump to over 84% and 91%, halving the RMSE. This statistically proves that modern monocular estimators exhibit region-dependent, piecewise-linear distortions. Consequently, an uncorrected global prior will inevitably warp the geometry of some objects while aligning others. Rather than treating these distortions as pure noise, we exploit their locally smooth and edge-aligned structure to introduce a spatially varying affine rectification model.

	MuSHRoom		**ScanNet++**	
Alignment Strategy	**Inlier Ratio (↑)**	**RMSE (↓)**	**Inlier Ratio (↑)**	**RMSE (↓)**
Global Affine ($K = 1$)	58.9%	0.074	58.6%	0.088
Piecewise Local ($K = 3$)	84.9%	0.046	84.6%	0.056
Piecewise Local ($K = 5$)	**92.7%**	**0.030**	**91.8%**	**0.039**

Table 1: **Empirical analysis of disparity distortions.** Evaluated at a 0.02 residual threshold. A single global transform ($K = 1$) leaves a significant percentage of the scene misaligned. Partitioning the scene into local linear regions ($K \geq 3$) empirically proves the piecewise-linear nature of monocular priors. See Appendix A for a comprehensive evaluation across all thresholds and depth models.

Local affine depth modeling. Let $d(u, v)$ denote the predicted monocular disparity at pixel (u, v), normalized to the min-max range of the current rendering, and let $\mathcal{D}(u, v)$ denote the alpha-blended metric depth rendered by 3DGS. Because monocular networks predict disparity without physical scale, its inverse $(1/d)$ acts as a relative depth map. Rather than assuming a strict global alignment, we propose that the relationship between this relative monocular depth and the true metric depth is best modeled locally as a spatially varying affine transformation:

$$\frac{1}{d(u, v)} = \alpha(u, v)\mathcal{D}(u, v) + \beta(u, v), \tag{1}$$

where the spatially varying fields $\alpha(u, v)$ and $\beta(u, v)$ correct local scale and shift distortions, respectively. This fundamentally contrasts with existing monocular alignment strategies, which rigidly assume α and β are single global constants applied uniformly across the entire image. By differentiating this equation with respect to screen coordinates and applying the chain rule, we obtain:

$$-\frac{1}{d^2}\nabla d = \alpha\nabla\mathcal{D} + \mathcal{D}\nabla\alpha + \nabla\beta. \tag{2}$$

Empirically, the affine correction fields vary more slowly than depth within coherent geometric regions. By applying the local smoothness approximations $\nabla\alpha \approx 0$ and $\nabla\beta \approx 0$, the relationship simplifies to $-\frac{1}{d^2}\nabla d \approx \alpha\nabla\mathcal{D}$, *i.e.,* the spatial gradient of the relative monocular depth is locally proportional to the spatial gradient of the rendered metric depth. To robustly estimate the local scale factor α across noisy gradients, we solve for it using the magnitudes of the finite-difference spatial gradients:

$$\alpha(u, v) = \frac{\|\nabla d\|}{d^2\|\nabla\mathcal{D}\| + \epsilon}, \tag{3}$$

where ϵ is a small constant for numerical stability.

Having solved for the local scale factor α using the spatial gradients, we substitute it back into Eq. 1 to recover the shift field: $\beta(u, v) = \frac{1}{d(u,v)} - \alpha(u, v)\mathcal{D}(u, v)$. To reject extreme outliers prior to filtering, we strictly clamp α to its 5^{th} and 95^{th} percentiles.

Edge-aware smoothing via guided filtering. To ensure regional coherence while respecting sharp object boundaries, we smooth the estimated α and β fields using a Guided Filter [8]. We use the monocular disparity d as the guide image, which naturally aligns the smoothing process with the scene's structural edges. This promotes local smoothness within coherent depth regions while preventing blurring across object boundaries, effectively capturing the piecewise-linear nature of

the monocular distortions. We denote these filtered fields as α_{smooth} and β_{smooth}. We use a large filter window (a radius of 550–650 pixels on MuSHRoom; Appendix Sec. C), so the filtered fields follow the edges of the guide while their local linear relation to it varies only slowly across the image. Because the guided filter is a local linear fit, it can push the filtered scale field outside the clamping range, including to non-positive values for which the corrected depth in Eq. 9 is undefined. We therefore clamp α_{smooth} to the same percentile range after filtering.

3.2 Normal-Aware Gradient-Domain Regularization

Depth values are zero-order measurements which are highly susceptible to shift ambiguities. In contrast, surface normals encode first-order geometry. Therefore, we can use these physical surface normals as a strong cue to support our gradient estimation. Specifically, by tying our rectified disparity gradients directly to the expected depth gradients derived from the normals and camera intrinsics, we create a supervision signal that naturally penalizes geometrically implausible local slopes.

Coupling disparity and depth gradients. Using our smoothed affine model, we can analytically predict the target disparity gradient $\nabla\hat{d}$ that perfectly matches the current rendered geometry:

$$\nabla\hat{d} = -\alpha_{\text{smooth}} d^2 \nabla\mathcal{D}. \tag{4}$$

To enforce consistency between the monocular prior and the rendered geometry, we minimize the discrepancy between the target gradients and the measured disparity gradients. We use a composite loss, applying a Mean Squared Error (MSE) on gradient magnitudes and a cosine similarity to enforce angular directional alignment:

$$\mathcal{L}_{\text{grad}} = \Big(\|\nabla\hat{d}\| - \|\nabla d\|\Big)^2 + \Big(1 - \cos(\nabla\hat{d}, \nabla d)\Big). \tag{5}$$

For robustness, we clamp both gradient magnitudes to their 5^{th} and 95^{th} percentiles before computing the magnitude term.

Normal-guided projective constraints. While $\mathcal{L}_{\text{grad}}$ transfers structural edges from the monocular prior to the rendered depth map, it only constrains the positions of the Gaussian primitives. Because 3DGS is a volumetric representation, primitives can form a visually smooth alpha-blended depth map while their individual geometric orientations remain chaotic or misaligned with the true surface. This orientation ambiguity severely degrades the quality of extracted 3D meshes.

To resolve this, we explicitly constrain the first-order orientation of the primitives. We achieve this by anchoring our regularization in projective geometry rather than raw depth; this ensures that spatial depth variations are directly tied to the physical 3D rotations of the Gaussians, preventing internal misalignment. We compute analytic depth gradients directly from the rendered surface normal $n = (n_x, n_y, n_z)$. By enforcing that the local image-space tangents of the back-projected surface are orthogonal to the normal under a pinhole camera with focal lengths (f_x, f_y) and principal point (c_x, c_y), the screen-space derivatives must satisfy:

$$\frac{\partial\mathcal{D}_{\text{norm}}}{\partial u} = \frac{-n_x\mathcal{D}/f_x}{n_z + n_x(u-c_x)/f_x + n_y(v-c_y)/f_y}, \tag{6}$$

$$\frac{\partial\mathcal{D}_{\text{norm}}}{\partial v} = \frac{-n_y\mathcal{D}/f_y}{n_z + n_x(u-c_x)/f_x + n_y(v-c_y)/f_y}. \tag{7}$$

(The full derivation from a pinhole projection is provided in Appendix Section B).

We substitute these normal-derived gradients into our affine model to generate normal-aware target disparity gradients: $\nabla\hat{d}_{\text{norm}} = -\alpha_{\text{smooth}} d^2 \nabla\mathcal{D}_{\text{norm}}$. By matching these to the monocular disparity gradients, we force the physical orientation of the Gaussians to align with the local slopes of the scene. We apply the same magnitude and cosine similarity loss to this normal branch:

$$\mathcal{L}_{\text{normal}} = \Big(\|\nabla\hat{d}_{\text{norm}}\| - \|\nabla d\|\Big)^2 + \Big(1 - \cos(\nabla\hat{d}_{\text{norm}}, \nabla d)\Big). \tag{8}$$

3.3 Metric Depth Re-alignment

While the gradient-domain losses explicitly supervise local structural fidelity, we additionally provide a direct $\mathcal{L}_1$ depth penalty to prevent the scene from drifting globally. Using the smoothed affine fields,

we reconstruct a corrected depth map:

$$\hat{\mathcal{D}} = \frac{1 - \beta_{\text{smooth}} d}{\alpha_{\text{smooth}} d}. \tag{9}$$

Although $\hat{\mathcal{D}}$ captures the correct local geometry, integrating local gradient constraints can leave the reconstruction susceptible to global, low-frequency scale drift relative to the 3DGS coordinate frame. To ensure stable $\mathcal{L}_1$ supervision without inappropriately penalizing arbitrary global scene scales, we perform a global least-squares solve to find optimal scalar values a and b such that $a\hat{\mathcal{D}} + b \approx \mathcal{D}$. This anchors the locally corrected depth to the current global scale of the scene. The final metric depth loss is computed against this globally aligned reconstruction:

$$\mathcal{L}_{\text{depth}} = |(a\hat{\mathcal{D}} + b) - \mathcal{D}|. \tag{10}$$

The total geometric regularization applied to the 3DGS optimization is the weighted sum of these three components with empirically determined weighting hyperparameters:

$$\mathcal{L}_{\text{geom}} = \lambda_1 \mathcal{L}_{\text{grad}} + \lambda_2 \mathcal{L}_{\text{normal}} + \lambda_3 \mathcal{L}_{\text{depth}}. \tag{11}$$

4 Experiments

Implementation details. We train 3DGS using the following loss objective, which is a weighted combination of our proposed geometric regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}} + \lambda_1 \mathcal{L}_{\text{grad}} + \lambda_2 \mathcal{L}_{\text{normal}} + \lambda_3 \mathcal{L}_{\text{depth}}. \tag{12}$$

We empirically set $\lambda_1 = 0.1$, $\lambda_2 = 0.08$, $\lambda_3 = 0.4$ and $\mathcal{L}_{\text{photo}}$ is the original photometric loss [12]. On the MuSHRoom scenes `kokko` and `sauna`, these default weights produce floaters, so we scale all three weights by 0.5 and 0.25, respectively. On ScanNet++, we use the 0.25 scaling and train at half resolution (960×720). Appendix Sec. C lists all per-scene settings. We train all models for 30k iterations. For densification, we use the unmodified adaptive density control of Splatfacto [23] (gradient-based cloning and splitting, opacity-based pruning, and periodic opacity reset) with its default settings, identical for our method and all Gaussian-based baselines. We change no densification logic, so all reported gains come from our supervision signal alone. To obtain disparity maps, we propagate RGB images through the pre-trained Depth-Anything-V2-Large model [27]. Crucially, the spatially varying affine parameters (*e.g.,* the scale factor α) used to rectify these monocular priors are iteratively updated alongside the 3DGS scene parameters during optimization. By following [24], we initialize the Gaussian scene using 1M back-projected points from training dataset sensor depths. The Algorithm column of Tables 4 and 5 lists the meshing algorithm of each method. We mesh our method, and DN-Splatter (Monodepth) on MuSHRoom, by fusing the depth maps rendered at the training views with Open3D TSDF fusion (voxel size 0.01 and truncation 0.03, in scene units). Methods meshed with Poisson reconstruction extract a total of 2M points from back-projected depth/normal maps and use a depth level of 9; Neusfacto and MonoSDF use marching cubes, and 2DGS uses TSDF fusion. Appendix Sec. F.5 compares Poisson and TSDF meshes of the same trained models. We refer readers to [24] for more details.

Baselines. We evaluate across implicit, explicit, and classical representations. Implicit baselines include Nerfacto, Depth-Nerfacto [23], Neusfacto [30], and MonoSDF [31]. Explicit baselines feature standard Splatfacto [23], geometry-focused SuGaR [7] and 2DGS [11], alongside depth-regularized DRGS [5] and DN-Splatter [24]. For a fair sensor-free comparison, we introduce DN-Splatter (Monodepth), which replaces its physical sensor data with dynamically aligned monocular priors (Depth-Anything-V2 [27]). Finally, we include classical Poisson surface reconstruction from back-projected sensor depth to contextualize upper-bound mesh performance [24].

Datasets. We validate our proposed method across a diverse set of challenging indoor environments. For standard forward-facing captures, we utilize the NeRF-LLFF dataset [14], which comprises distinct real-world scenes. To assess generalization to complex camera movements, we evaluate on the MuSHRoom dataset [20], a real-world indoor benchmark specifically designed with disjoint camera trajectories for training and testing. Finally, to evaluate geometric accuracy, we benchmark on ScanNet++ [28], a large-scale real-world dataset providing dense RGB sequences paired with high-fidelity 3D ground-truth geometry.

	Scene	Methods	PSNR ↑				SSIM ↑				LPIPS ↓			
			2-view	3-view	4-view	5-view	2-view	3-view	4-view	5-view	2-view	3-view	4-view	5-view
NeRF-LLFF	Fern	3DGS	13.03	14.29	16.73	18.59	0.336	0.408	0.517	0.603	0.476	0.389	0.296	0.217
		DropGau.	14.71	17.58	18.04	19.27	0.413	0.565	0.620	0.670	0.433	0.268	0.237	0.202
		DepthSplat	17.77	18.86	19.20	19.71	0.454	0.489	0.538	0.558	0.336	0.316	0.271	0.283
		DRGS	17.59	19.13	19.91	20.55	0.516	0.588	0.616	0.642	0.286	0.232	0.203	0.167
		Ours	**18.94**$^{+1.17}$	**20.18**$^{+1.05}$	**21.30**$^{+1.39}$	**22.25**$^{+1.70}$	**0.543**$^{+0.027}$	**0.650**$^{+0.062}$	**0.694**$^{+0.074}$	**0.739**$^{+0.069}$	**0.266**$^{-0.020}$	**0.189**$^{-0.043}$	**0.154**$^{-0.049}$	**0.121**$^{-0.046}$
		Oracle	18.18	20.30	20.78	21.81	0.524	0.636	0.654	0.701	0.278	0.201	0.185	0.157
	Flower	3DGS	14.90	17.75	19.71	21.39	0.351	0.508	0.605	0.671	0.406	0.257	0.190	0.146
		DropGau.	16.28	19.16	19.92	21.32	0.433	0.589	0.633	0.686	0.320	0.194	0.171	0.138
		DepthSplat	11.83	15.29	17.97	18.17	0.154	0.317	0.425	0.460	0.623	0.432	0.332	0.327
		DRGS	15.92	17.80	19.15	20.45	0.395	0.445	0.538	0.576	0.414	0.376	0.323	0.293
		Ours	**19.35**$^{+3.07}$	**21.13**$^{+1.97}$	**22.45**$^{+2.53}$	**23.45**$^{+2.06}$	**0.567**$^{+0.134}$	**0.657**$^{+0.068}$	**0.722**$^{+0.089}$	**0.754**$^{+0.068}$	**0.188**$^{-0.132}$	**0.141**$^{-0.053}$	**0.110**$^{-0.061}$	**0.090**$^{-0.048}$
		Oracle	19.71	22.16	23.26	24.65	0.570	0.673	0.714	0.760	0.250	0.163	0.128	0.097
	Fortress	3DGS	13.87	15.98	19.26	19.98	0.363	0.492	0.609	0.631	0.389	0.283	0.201	0.191
		DropGau.	10.94	13.92	17.96	17.36	0.419	0.557	0.685	0.693	0.490	0.273	0.193	0.197
		DepthSplat	13.27	16.50	18.81	18.68	0.309	0.339	0.409	0.401	0.425	0.408	0.311	0.341
		DRGS	19.80	21.85	23.07	23.72	**0.567**	0.655	0.724	0.740	0.232	0.191	0.162	0.144
		Ours	**20.52**$^{+0.72}$	**22.52**$^{+0.67}$	**24.63**$^{+1.56}$	**25.18**$^{+1.46}$	0.547$^{-0.020}$	**0.674**$^{+0.019}$	**0.736**$^{+0.012}$	**0.774**$^{+0.034}$	**0.187**$^{-0.045}$	**0.133**$^{-0.058}$	**0.119**$^{-0.043}$	**0.111**$^{-0.033}$
		Oracle	23.07	24.51	26.39	26.73	0.654	0.728	0.787	0.797	0.159	0.130	0.100	0.093
	Horns	3DGS	11.43	12.48	13.76	14.75	0.264	0.339	0.433	0.498	0.531	0.464	0.395	0.350
		DropGau.	10.09	15.46	15.40	16.53	0.280	0.517	0.551	0.611	0.531	0.329	0.313	0.283
		DepthSplat	13.06	16.08	15.91	16.71	0.361	0.421	0.445	0.470	0.418	0.376	0.380	0.383
		DRGS	15.91	16.22	18.09	18.39	0.420	0.466	0.527	0.565	0.362	0.349	0.306	0.296
		Ours	**16.34**$^{+0.43}$	**19.95**$^{+3.73}$	**21.08**$^{+2.99}$	**22.39**$^{+4.00}$	**0.473**$^{+0.053}$	**0.671**$^{+0.154}$	**0.730**$^{+0.179}$	**0.776**$^{+0.165}$	**0.300**$^{-0.062}$	**0.191**$^{-0.138}$	**0.152**$^{-0.154}$	**0.123**$^{-0.160}$
		Oracle	18.56	20.08	20.88	22.52	0.568	0.644	0.668	0.725	0.259	0.212	0.199	0.168
	Mean	3DGS	13.31	15.13	17.37	18.68	0.329	0.437	0.541	0.601	0.451	0.348	0.271	0.226
		DropGau.	13.00	16.53	17.83	18.62	0.386	0.557	0.622	0.665	0.444	0.266	0.229	0.205
		DepthSplat	13.98	16.68	17.97	18.32	0.319	0.391	0.454	0.472	0.450	0.383	0.324	0.334
		DRGS	17.31	18.75	20.06	20.78	0.475	0.539	0.601	0.631	0.324	0.287	0.249	0.225
		Ours	**18.79**$^{+1.48}$	**20.95**$^{+2.20}$	**22.37**$^{+2.31}$	**23.32**$^{+2.54}$	**0.533**$^{+0.058}$	**0.663**$^{+0.106}$	**0.721**$^{+0.099}$	**0.761**$^{+0.096}$	**0.235**$^{-0.089}$	**0.164**$^{-0.102}$	**0.134**$^{-0.095}$	**0.111**$^{-0.094}$
		Oracle	19.88	21.76	22.83	23.93	0.579	0.670	0.706	0.746	0.237	0.177	0.153	0.129

Table 2: Quantitative comparisons on NeRF-LLFF dataset. The best performance (excluding the Oracle) is highlighted in **red**, and the second best in blue. The superscript denotes the absolute performance delta of our method against the best-performing baseline.

Evaluation metrics. We follow standard practice and report PSNR, SSIM and LPIPS metrics for color images and common depth metrics to analyze depth quality for datasets containing ground truth sensor data. For assessing mesh quality, we follow [24] and report accuracy, completion, Chamfer distance, normal consistency and F-scores with a threshold of 5cm.

4.1 Sparse-View Reconstruction - Table 2

Settings. Following the evaluation protocol of DRGS [5], we partition each scene into distinct training and testing sets. For our sparse-view experiments, we optimize the scenes using k randomly sampled training images ($k \in \{2, 3, 4, 5\}$) and evaluate novel-view synthesis on the fixed test set. To ensure statistical reliability, all reported metrics are averaged over ten random seed selections. Additionally, we reference the *Oracle* performance established by DRGS. This oracle serves as an upper bound to demonstrate the theoretical impact of perfect depth geometry; it is generated by optimizing a model on the entire dense image collection (both train and test) to create pseudo-ground-truth depth maps, which are then used in place of standard depth estimates.

Discussions. While novel view synthesis inherently degrades with view sparsity, our method demonstrates robustness and outperforms all baselines. Notably, DropGaussian's [16] performance severely collapses in the 2-view setting. While their originally reported metrics benefit from a dense, full-scene SfM initialization, our strict evaluation generates SfM points *only* from the available training views, causing their initialization-heavy strategy to fail. Conversely, our approach does not rely on a dense initial point cloud; instead, our spatial distortion correction continuously regularizes the geometry throughout optimization. The feed-forward DepthSplat [26], evaluated zero-shot, is competitive only on Fern, the scene closest to its training distribution, and trails our method by 4.3 to 5.0 dB in mean PSNR at every view count. Crucially, by rectifying these monocular priors, our method unlocks their true potential, bridging the gap to—and occasionally surpassing—the *Oracle* (which relies on dense, multi-view pseudo-GT depth). This occurs because our structural depth priors actively regularize textureless or ambiguous regions where unconstrained multi-view optimization still struggles. Qualitatively (Fig. 3), our method demonstrates a clear reduction in floating artifacts and structural collapse compared to the baselines.



(a) 3DGS (b) DRGS (c) DropGaussian (d) Ours

Figure 3: Qualitative comparisons on NeRF-LLFF dataset (more in Appendix Sec. E.1).

		Sensor Depth Loss	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑
NeRF	Nerfacto [23]	–	20.86 / 20.66	.7859 / .7633	.2321 / .2702	.0862 / .0747	.0293 / .0141	.0794 / .0667	.9335 / .9428
	Depth-Nerfacto [23]	✓	21.24 / 18.93	.7832 / .7023	.2414 / .3978	.0727 / .0563	.0155 / .0283	.0583 / .2840	.9469 / .9389
SDF	MonoSDF [31]	✓	20.68 / 20.16	.7357 / .7653	.3590 / .2261	.0555 / .0911	.0263 / .0804	.2493 / .4535	.9353 / .8825
Explicit	DN-Splatter [24]	✓	24.58 / 21.89	.8558 / .7984	.1293 / .1797	.0228 / .0354	.0089 / .0214	.1280 / .2032	.9854 / .9683
	SuGaR[7]	–	20.52 / 18.18	.7740 / .7125	.2427 / .2959	.1213 / .1174	.1258 / .1474	.5562 / .5820	.8412 / .8564
	2DGS [11]	–	22.52 / 21.73	.8185 / .7898	.1773 / .1911	.0864 / .0923	.0612 / .0583	.3799 / .3132	.8820 / .8927
	Splatfacto [23]	–	24.44 / 21.33	.8486 / .7821	.1387 / .2240	.0787 / .0817	.0364 / .0521	.2407 / .2941	.9072 / .9061
	DN-Splatter (Monodepth) [24]	–	19.26 / 17.24	.7255 / .6845	.3338 / .3851	.3542 / .3669	.6454 / .7852	.9351 / 1.103	.5682 / .5818
	Ours	–	**24.49/ 22.07**	**.8570 / .7920**	**.1288 / .1733**	**.0551 / .0490**	**.0176 / .0176**	**.1854 / .1953**	**.9674 / .9729**

Table 3: Quantitative comparison on MuSHRoom dataset. We show that our method achieves state-of-the-art geometric accuracy among sensor-free explicit representations, completely avoiding the structural collapse observed in the directly comparable DN-Splatter (Monodepth) baseline. The best results among sensor depth-free methods (indicated by –) are highlighted in **bold**.

4.2 Novel-View Synthesis and Depth Estimation - Table 3

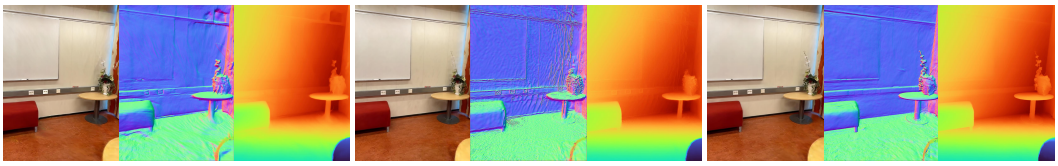
Settings. Following the protocol of [24], we report novel-view synthesis and depth metrics across two distinct evaluation sets (`left` / `right`). The `left` split consists of a test set sampled uniformly every 10 frames from the training sequence, whereas the `right` split evaluates a disjoint camera trajectory with no training overlap. Results are averaged over 6 scenes.

Discussions. Crucially, our proposed method closes the performance gap with methods relying on ground-truth sensor depth, despite strictly operating without it. For novel-view synthesis, we achieve competitive performance, closely matching the sensor-supervised DN-Splatter and outperforming other sensor-free explicit methods like 2DGS and SuGaR. The most significant advantage of our approach lies in its geometric accuracy. While DN-Splatter achieves strong geometry using physical sensor data, its performance severely degrades when forced to rely on monocular priors; the DN-Splatter (Monodepth) baseline experiences a massive structural collapse, yielding an Abs Rel error of 0.3542 / 0.3669 (see Appendix Table 18 for per-scene results). In contrast, we successfully leverage those exact same monocular priors to achieve an exceptional Abs Rel of 0.0551 / 0.0490. This proves that our method can extract high-fidelity geometric constraints from monocular networks without requiring the global sensor alignment that prior works depend on. Finally, Figs. 4 and 5 qualitatively corroborate these findings, demonstrating that our geometric regularization preserves crisp novel-view synthesis and prevents structural collapse across both the MuSHRoom and ScanNet++ datasets.

4.3 Mesh Evaluation - Tables 4 and 5

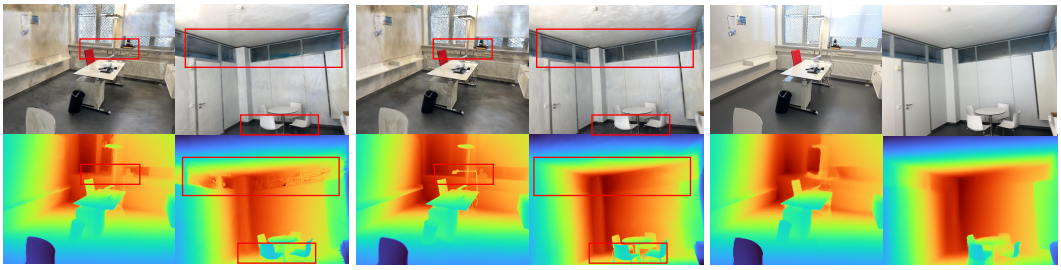
Settings. We evaluate on MuSHRoom and ScanNet++ datasets. For the MuSHRoom dataset, we report results averaged over 6 scenes: `coffee_room`, `honka`, `kokko`, `sauna`, `computer`, and `vr_room`. For the ScanNet++ dataset, the results are averaged over the `b20a261fdf` and `8b5caf3398` scenes. Where applicable, we also report our self-reproduced results for DN-Splatter in parentheses.

Discussions. The mesh evaluation results validate the structural benefits of our approach. The most critical observation across both datasets is the catastrophic failure of DN-Splatter when deprived of its ground-truth sensor depth. On the MuSHRoom dataset, the DN-Splatter (Monodepth) baseline collapses to an F-score of 0.2765. In contrast, our method maintains geometric accuracies with the same monocular priors (F-score of 0.7097). Furthermore, on the MuSHRoom dataset, our method establishes state-of-the-art performance among all evaluated explicit, sensor-free representations, outperforming dedicated meshing baselines like SuGaR and 2DGS. On ScanNet++, our method again greatly improves over DN-Splatter (Monodepth) and achieves the highest normal consistency among sensor-free methods, while SuGaR and 2DGS obtain lower Chamfer distance and higher F-score. On the `b20a261fdf` scene, our method, like plain Splatfacto, renders depth about 5% shorter than the sensor depth, which limits its F-score at the 5 cm threshold (Appendix Sec. H). Across all benchmarks, while sensor-supervised methods naturally dictate the upper bound of performance, our proposed method extracts high-quality geometries without any physical depth data (see Fig. 6).



(a) DN-Splatter (Monodepth) (b) DN-Splatter (c) Ours

Figure 4: Qualitative comparison on Mushroom dataset (more in Appendix Sec. E.2). (a) The globally aligned DN-Splatter (Monodepth) baseline suffers from degraded geometries due to structural warping. (c) By rectifying the exact same monocular prior, our method better recovers crisp planar surfaces and sharp complex object boundaries than (b) the sensor-supervised DN-Splatter.



(a) DN-Splatter (b) Ours (c) Ground truth

Figure 5: Qualitative comparison on ScanNet++ dataset. We show rendered RGB and depth with red bounding boxes highlighting areas where we generate visually sharper depth, particularly along boundaries of furniture and complex geometry, compared to the DN-Splatter baseline.

		Sensor Depth	Algorithm	Accuracy ↓	Completion ↓	Chamfer-L_1 ↓	Normal Cons. ↑	F-score ↑
	Traditional Poisson	–	Poisson	.0399	.0222	.0306	.8688	.8823
NeRF	Nerfacto [23]	–	Poisson	.0430	.0578	.0504	.7822	.7212
NeRF	Depth-Nerfacto [23]	✓	Poisson	.0447	.0557	.0502	.7614	.6966
SDF	MonoSDF [31]	✓	Marching-Cubes	.0310	.0190	.0250	.8846	.9211
Explicit	DN-Splatter [24]	✓	Poisson	.0239(.0255)	.0194(.0291)	.0216(.0273)	.8822(.8685)	.9243(.9065)
Explicit	Splatfacto [23]	–	Poisson	.0749	.0555	.0652	.7727	.5835
Explicit	2DGS [11]	–	TSDF	.0731	.0642	.0687	.8008	.6039
Explicit	SuGaR [7]	–	Poisson+IBR	.0656	.0583	.0620	.8031	.6378
Explicit	DN-Splatter (Monodepth) [24]	–	TSDF	.2013	.3196	.2605	.7093	.2765
Explicit	Ours	–	TSDF	**.0535**	**.0437**	**.0486**	**.8662**	**.7097**

Table 4: Mesh evaluation on MuSHRoom dataset. While DN-Splatter achieves strong geometric accuracy using sensor depth, substituting it with monocular priors (DN-Splatter (Monodepth)) results in severe structural degradation. Our method successfully leverages those same monocular priors without global alignment and outperforms prior explicit baselines like SuGaR and 2DGS. Best results among methods not relying on sensor depth are indicated in **bold**. Appendix Sec. F.5 compares Poisson and TSDF meshes of the same trained models.

		Sensor Depth	Algorithm	Accuracy ↓	Completion ↓	Chamfer-L_1 ↓	Normal Cons. ↑	F-score ↑	Time (min)
	Traditional Poisson	–	Poisson	.0593	.0574	.0564	.4410	.7923	2.0
NeRF	Nerfacto [23]	–	Poisson	.1305	.1484	.1394	.7153	**.4698**	8.0
NeRF	Depth-Nerfacto [23]	✓	Poisson	.0731	.1647	.1189	.6848	.5018	8.1
SDF	Neusfacto [30]	–	Marching-Cubes	**.0736**	.1945	.1340	.7159	.4605	40.0
SDF	MonoSDF [31]	✓	Marching-Cubes	.0303	.0573	.0438	.8881	.8577	47.5
Explicit	DN-Splatter	✓	Poisson	.0940(.1226)	.0395(.0685)	.0667(.0956)	.8316(.7794)	.7658(.5644)	36.9
Explicit	Splatfacto [23]	–	Poisson	.1934	.1503	.1719	.6741	.1790	8.9
Explicit	2DGS [11]	–	TSDF	.1272	**.0798**	.1035	.7799	.4196	33.5
Explicit	SuGaR [7]	–	Poisson + IBR	.0940	.1011	**.0975**	.7241	.4367	70.0
Explicit	DN-Splatter (Monodepth)	–	Poisson	.1798	.2087	.1975	.7378	.1395	38.0
Explicit	Ours	–	TSDF	.1212	.0924	.1068	**.8308**	.3207	43.4

Table 5: Mesh evaluation on ScanNet++ dataset. While sensor-supervised methods dictate the upper bound performances, our approach achieves the highest normal consistency among sensor-free methods and greatly improves over DN-Splatter (Monodepth), while several baselines achieve better Chamfer distance and F-score. Best results among sensor-free methods are indicated in **bold**.



(a) DN-Splatter (b) Ours (c) Ground truth

Figure 6: Qualitative mesh comparison on ScanNet++ dataset (more in Appendix Sec. E.3). Interestingly, we are able to recover window geometries which are absent from ground-truth mesh.

4.4 Ablations

Proposed geometric regularization (Table 6). Unconstrained Splatfacto achieves high PSNR (25.293) but poor geometry (F-score 0.728) due to appearance overfitting. Adding $\mathcal{L}_{\text{depth}}$ anchors global scale (Abs Rel: $0.059 \rightarrow 0.043$), while $\mathcal{L}_{\text{normal}}$ promotes surface smoothness. Crucially, $\mathcal{L}_{\text{grad}}$ couples disparity gradients to physical structure, preventing the optimizer from "cheating" for photometric gain. Enforcing these joint constraints in our full formulation (All) trades a negligible PSNR drop (24.947 vs. 25.293) for state-of-the-art geometric fidelity. Full results and qualitative comparisons are provided in Appendix Table 13 and Fig. 12.

Method	**Photometry**		**Geometry**		
	PSNR ↑	LPIPS ↓	Abs Rel ↓	Accuracy ↓	F-Score ↑
Splatfacto (Baseline)	25.293	0.086	0.059	0.043	0.728
+ $\mathcal{L}_{\text{grad}}$	24.322	0.123	0.085	0.054	0.673
+ $\mathcal{L}_{\text{normal}}$	25.256	0.085	0.064	0.042	0.728
+ $\mathcal{L}_{\text{depth}}$	24.561	0.098	0.043	0.030	0.798
+ $\mathcal{L}_{\text{grad}}$ + $\mathcal{L}_{\text{normal}}$	24.553	0.116	0.075	0.051	0.707
+ $\mathcal{L}_{\text{grad}}$ + $\mathcal{L}_{\text{depth}}$	25.012	0.101	0.041	0.026	0.850
+ $\mathcal{L}_{\text{normal}}$ + $\mathcal{L}_{\text{depth}}$	**25.418**	**0.084**	**0.040**	0.028	0.813
All (Ours)	24.947	0.102	**0.040**	**0.025**	**0.855**

Table 6: Ablations on proposed geometric regularization components. Evaluated on the `coffee_room` scene from the MuSHRoom dataset. Compared to the Splatfacto baseline, our full formulation (*All*) trades a negligible PSNR drop for geometric accuracies. Best geometric results are **bolded**.

Additional ablations regarding the sensitivity of monocular depth models, filtering strategies, and gradient matching formulations are provided in Appendix Sections F.2, F.3, and F.4, respectively.

5 Conclusion

In this work, we identified a critical limitation in existing depth-guided 3DGS methods: the reliance on global linear alignment fails to account for the region-dependent, piecewise-linear distortions inherent to modern monocular depth priors. To overcome this, we introduced a novel spatially varying affine rectification framework that dynamically corrects these monocular distortions into geometrically consistent metric depth using edge-aware, locally smooth fields. By coupling this rectified depth with normal-aware gradient constraints, our method explicitly regularizes the physical orientation of 3D primitives without requiring any architectural modifications to 3DGS. Extensive evaluations across the NeRF-LLFF, MuSHRoom, and ScanNet++ datasets demonstrate that our approach significantly outperforms sensor-free depth-regularized baselines, as well as existing explicit baselines on NeRF-LLFF and MuSHRoom. Ultimately, our method bridges the gap toward sensor-supervised geometric fidelity.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[3] Hanlin Chen, Chen Li, Yunsong Wang, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.

[4] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *CVPR*, 2023.

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

[6] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *ECCV*, 2024.

[7] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *CVPR*, 2024.

[8] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *TPAMI*, 2012.

[9] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021.

[10] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *TPAMI*, 2024.

[11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, 2024.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM ToG*, 2023.

[13] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020.

[14] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM ToG*, 2019.

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021.

[16] Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *CVPR*, 2025.

[17] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[18] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021.

[19] Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM ToG*, 2023.

[20] Xuqian Ren, Wenjia Wang, Dingding Cai, Tuuli Tuominen, Juho Kannala, and Esa Rahtu. Mushroom: Multi-sensor hybrid room dataset for joint 3d reconstruction and novel view synthesis. In *WACV*, 2024.

[21] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[22] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.

[23] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH*, 2023.

[24] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *WACV*, 2025.

[25] 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.

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

[27] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 2024.

[28] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

[29] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

[30] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022.

[31] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022.

[32] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM ToG*, 2024.

A Comprehensive Analysis of Monocular Disparity Distortions

In Sec. 3.1 of the main paper, we demonstrated that monocular depth priors suffer from region-dependent, piecewise-linear distortions by evaluating RANSAC inlier ratios at a standard residual threshold of 0.02. To rigorously prove that this piecewise behavior is an inherent property of foundation depth models and not merely an artifact of a specific error tolerance, we expand our empirical analysis here.

A.1 The Discovery of Piecewise-Linear Disparity Fragmentation

We uncover a previously uncharacterized, systematic failure mode inherent to modern foundation depth estimators: **piecewise-linear disparity fragmentation**. While foundation models are highly effective at capturing relative depth ordering, our analysis reveals that their predictions do not map to true metric scale through a unified global transform. Instead, the networks systematically fracture scenes into distinct, spatially coherent affine regimes.

At first glance, one might assume that allocating more degrees of freedom (increasing the number of local regions, K) naturally reduces residual error, akin to fitting a higher-degree polynomial to noisy data. However, it is crucial to distinguish the structured distortions we observe from complex pixel-level noise. If the monocular priors simply suffered from random, unstructured inaccuracies, partitioning the data into K linear models would merely overfit to random, scattered subsets of pixels.

Our empirical evidence, particularly the spatial coherence masks shown in the main text (Fig. 1), proves otherwise. The K affine regimes are not random; they are highly structured and correspond directly to coherent physical surfaces. The breaks between these linear regimes align strictly with dominant object boundaries and large depth discontinuities. This proves that foundation models encode relative depth smoothly within continuous physical surfaces, but apply fundamentally different scale-and-shift parameters across different objects.

A.2 The Mathematical Impossibility of Global Alignment

This phenomenon fundamentally invalidates the core assumption of existing depth-regularized 3DGS methods. A single global affine transform possesses exactly two degrees of freedom (scale and shift). Consequently, it is mathematically impossible for a single transform to simultaneously reconcile K fundamentally incompatible linear regimes. Global alignment in this context is not merely suboptimal—it is a mathematically ill-posed operation. Attempting to fit a single line through the fragmented disparity space will inevitably force the optimizer to average the error across conflicting spatial clusters, resulting in the severe warping of physical geometry in some regions to satisfy the alignment of others (see Fig. 7).

A.3 Empirical Validation Across Foundation Models and Datasets

To demonstrate that piecewise-linear disparity fragmentation is a systemic characteristic of monocular priors and not an isolated artifact of a specific architecture or dataset, we expand our empirical evaluation across three state-of-the-art foundation models: Metric3D-V2, MoGe, and Depth-Anything-V2. As a direct extension to the visual evidence provided in our main text Fig. 1 (which utilized Depth-Anything-V2), we visually confirm in Fig. 8 that both Metric3D-V2 and MoGe exhibit the region-dependent fragmentation behavior.

For each model, we evaluate the affine alignment over the MuSHRoom dataset sequences using iteratively fitted piecewise local linear regions ($K \in \{1, \ldots, 8\}$). To rigorously test the spatial alignment, we evaluate the models across three separate residual error thresholds: a strict threshold (0.005), a moderate threshold (0.02), and a forgiving threshold (0.05). Furthermore, to ensure these findings generalize across different capturing hardware and indoor environments, we additionally evaluate Depth-Anything-V2 on the ScanNet++ dataset.

As shown in Tables 7 through 10, the global alignment assumption ($K = 1$) fails significantly under strict thresholds across all models and datasets. For instance, at a strict tolerance on MuSHRoom, a global transform aligns only 25.6% of the geometry produced by Depth-Anything-V2. When tested on the entirely different ScanNet++ dataset, it similarly aligns a mere 26.9% (Table 10). However, by simply allowing the evaluation to independently align local affine regions ($K \geq 3$), the inlier ratios

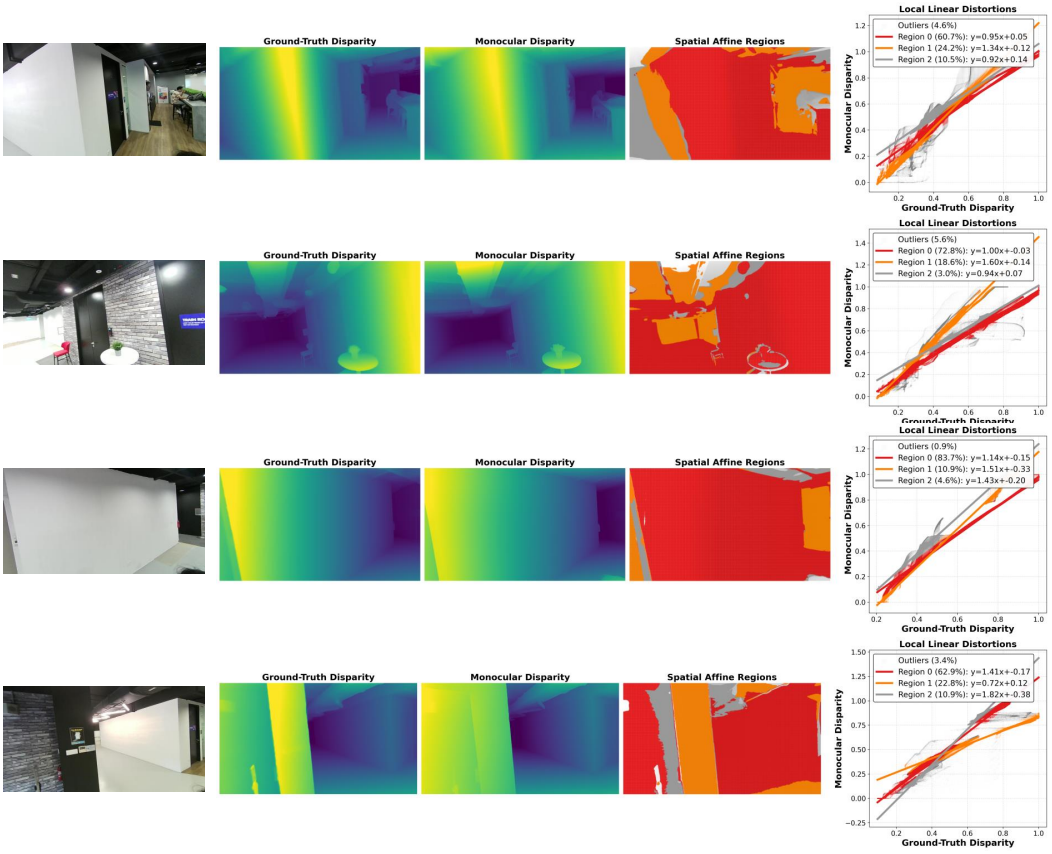


Figure 7: Piecewise-linear disparity fragmentation across diverse indoor topologies. Across various scenes, the monocular depth prior consistently fractures into distinct, spatially coherent affine regimes (visualized in red, orange, and grey). The accompanying scatter plots demonstrate that fitting a single global scale-and-shift transform ($K = 1$) is mathematically ill-posed, as the optimizer would be forced to average errors across fundamentally incompatible linear clusters.

consistently surge to near-perfect alignment across the board. This drastic improvement confirms that the massive percentage of "outliers" rejected by the global transform are not random noise, but rather highly accurate geometric surfaces operating under entirely different affine parameters.

	Strict Threshold (0.005)			Moderate Threshold (0.02)			Forgiving Threshold (0.05)		
Alignment Strategy	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**
Global Affine ($K = 1$)	69.4%	0.518	0.252	77.0%	0.517	0.253	92.2%	0.517	0.254
Piecewise Local ($K = 3$)	88.7%	0.515	0.271	92.6%	0.513	0.274	98.1%	0.373	0.452
Piecewise Local ($K = 4$)	92.1%	0.502	0.289	94.8%	0.463	0.329	98.8%	0.253	0.635
Piecewise Local ($K = 5$)	94.0%	0.454	0.338	96.1%	0.394	0.414	99.1%	0.181	0.759
Piecewise Local ($K = 6$)	95.3%	0.406	0.399	97.0%	0.326	0.509	99.3%	0.137	0.839
Piecewise Local ($K = 7$)	96.1%	0.356	0.468	97.6%	0.262	0.612	99.5%	0.101	0.906
Piecewise Local ($K = 8$)	**96.8%**	**0.291**	**0.564**	**98.0%**	**0.222**	**0.679**	**99.6%**	**0.086**	**0.931**

Table 7: **Comprehensive Statistical Analysis of Distortions (MuSHRoom dataset).** Evaluated across iPhone sequences using the **Metric3D-V2 (ViT-Large)** prior. The dramatic performance gap between global alignment ($K = 1$) and piecewise alignment ($K \geq 3$) persists regardless of the chosen error threshold, proving that foundation models fundamentally exhibit spatially varying scale and shift distortions.

B Derivation of Normal-Aware Depth Gradient Constraints

This section provides the full derivation for the normal-aware depth gradients $\frac{\partial \mathcal{D}}{\partial u}$ and $\frac{\partial \mathcal{D}}{\partial v}$ used in Section 3.2 of the main paper. These constraints are derived by enforcing that the local image-space

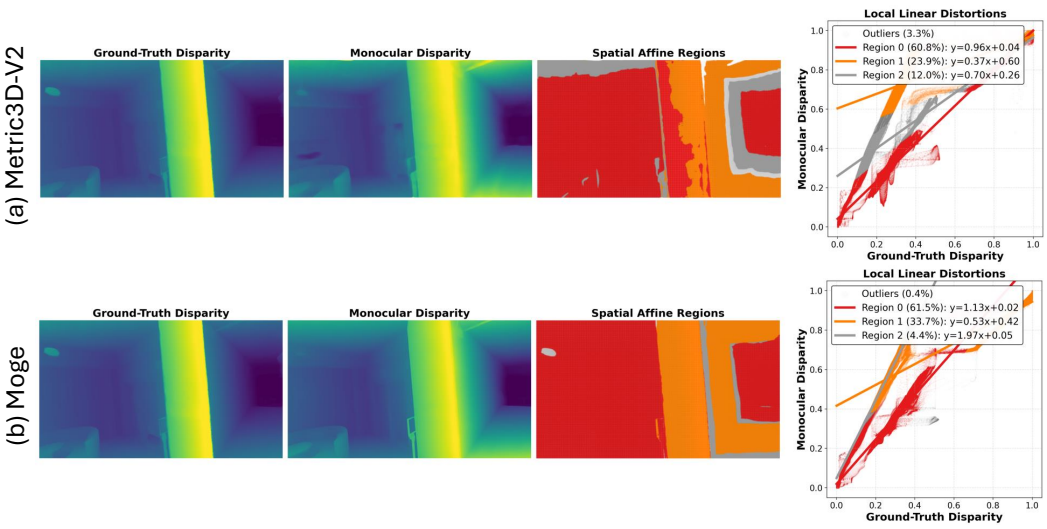


Figure 8: We compare the raw disparity predictions of (a) Metric3D-V2 and (b) MoGe against ground-truth sensor depth on the exact same scene. While the distinct network architectures allocate their local scale-and-shift parameters differently, both fundamentally exhibit the same region-dependent, piecewise-linear fragmentation. This confirms that spatial affine distortion is a systemic property of modern foundation depth models.

	Strict Threshold (0.005)			Moderate Threshold (0.02)			Forgiving Threshold (0.05)		
Alignment Strategy	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**
Global Affine ($K = 1$)	44.1%	0.042	0.982	77.4%	0.040	0.981	92.1%	0.039	0.982
Piecewise Local ($K = 3$)	70.5%	0.032	0.986	93.7%	0.024	0.991	98.6%	0.020	0.994
Piecewise Local ($K = 4$)	76.9%	0.029	0.988	96.0%	0.021	0.993	99.2%	0.017	0.996
Piecewise Local ($K = 5$)	81.4%	0.026	0.989	97.1%	0.018	0.995	99.5%	0.015	0.997
Piecewise Local ($K = 6$)	84.7%	0.024	0.991	97.9%	0.015	0.996	99.7%	0.013	0.998
Piecewise Local ($K = 7$)	87.1%	0.023	0.991	98.4%	0.013	0.997	99.8%	0.013	0.998
Piecewise Local ($K = 8$)	**89.0%**	**0.022**	**0.992**	**98.7%**	**0.012**	**0.998**	**99.9%**	**0.012**	**0.998**

Table 8: **Comprehensive Statistical Analysis of Distortions (MuSHRoom dataset).** Evaluated across iPhone sequences using the **MoGe (ViT-Large)** prior. The dramatic performance gap between global alignment ($K = 1$) and piecewise alignment ($K \geq 3$) persists regardless of the chosen error threshold, proving that foundation models fundamentally exhibit spatially varying scale and shift distortions.

	Strict Threshold (0.005)			Moderate Threshold (0.02)			Forgiving Threshold (0.05)		
Alignment Strategy	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**	**Inlier % (↑)**	**RMSE (↓)**	**Corr. (↑)**
Global Affine ($K = 1$)	25.6%	0.086	0.967	58.9%	0.074	0.967	82.4%	0.071	0.967
Piecewise Local ($K = 3$)	50.6%	0.060	0.974	84.9%	0.046	0.980	96.3%	0.033	0.990
Piecewise Local ($K = 4$)	58.4%	0.055	0.976	89.7%	0.037	0.985	98.0%	0.026	0.994
Piecewise Local ($K = 5$)	64.4%	0.050	0.978	92.7%	0.030	0.990	98.8%	0.022	0.996
Piecewise Local ($K = 6$)	69.2%	0.047	0.980	94.6%	0.025	0.994	99.2%	0.019	0.997
Piecewise Local ($K = 7$)	73.1%	0.043	0.982	95.9%	0.022	0.995	99.5%	0.017	0.998
Piecewise Local ($K = 8$)	**76.3%**	**0.039**	**0.984**	**96.8%**	**0.019**	**0.996**	**99.7%**	**0.016**	**0.998**

Table 9: **Comprehensive Statistical Analysis of Distortions (MuSHRoom dataset).** Evaluated across iPhone sequences using the **Depth-Anything-V2 (ViT-Large)** prior. The dramatic performance gap between global alignment ($K = 1$) and piecewise alignment ($K \geq 3$) persists regardless of the chosen error threshold, proving that foundation models fundamentally exhibit spatially varying scale and shift distortions.

tangents of a back-projected surface patch are strictly orthogonal to the surface normal under a pinhole camera model.

B.1 Setup and Notation

Let (u, v) be the continuous screen-space pixel coordinates. We assume a standard pinhole camera with focal lengths (f_x, f_y) and principal point (c_x, c_y).

Alignment Strategy	Strict Threshold (0.005)			Moderate Threshold (0.02)			Forgiving Threshold (0.05)		
	Inlier % (↑)	RMSE (↓)	Corr. (↑)	Inlier % (↑)	RMSE (↓)	Corr. (↑)	Inlier % (↑)	RMSE (↓)	Corr. (↑)
Global Affine ($K = 1$)	26.9%	0.092	0.952	58.6%	0.088	0.952	81.6%	0.084	0.952
Piecewise Local ($K = 3$)	51.8%	0.068	0.967	84.6%	0.056	0.974	95.2%	0.040	0.985
Piecewise Local ($K = 4$)	59.2%	0.065	0.969	89.1%	0.047	0.980	97.1%	0.031	0.991
Piecewise Local ($K = 5$)	64.8%	0.062	0.971	91.8%	0.039	0.985	98.1%	0.025	0.994
Piecewise Local ($K = 6$)	69.3%	0.058	0.974	93.7%	0.032	0.989	98.8%	0.022	0.995
Piecewise Local ($K = 7$)	73.1%	0.053	0.978	95.0%	0.027	0.992	99.1%	0.019	0.997
Piecewise Local ($K = 8$)	**76.2%**	**0.049**	**0.981**	**96.0%**	**0.024**	**0.993**	**99.4%**	**0.017**	**0.998**

Table 10: **Comprehensive Statistical Analysis of Distortions (ScanNet++ dataset).** The ScanNet++ benchmark exhibits the exact same piecewise-linear behavior as the MuSHRoom dataset when using the Depth-Anything-V2 (ViT-Large) prior, demonstrating that our findings on monocular distortions generalize across different indoor environments and capturing hardware.

A pixel at location (u, v) with metric depth $\mathcal{D}(u, v)$ back-projects into 3D camera coordinates $\mathbf{P}(u, v) = [X, Y, Z]^\top$ according to the pinhole projection equations:

$$X = \frac{u - c_x}{f_x}\mathcal{D}(u, v), \quad Y = \frac{v - c_y}{f_y}\mathcal{D}(u, v), \quad Z = \mathcal{D}(u, v). \tag{13}$$

Let $\mathbf{n} = [n_x, n_y, n_z]^\top$ denote the unit surface normal vector at this 3D point. Geometrically, the partial derivatives of the 3D position with respect to the image coordinates, $\frac{\partial \mathbf{P}}{\partial u}$ and $\frac{\partial \mathbf{P}}{\partial v}$, represent the tangent vectors of the local surface patch. By definition, these tangent vectors must be orthogonal to the surface normal $\mathbf{n}$:

$$\mathbf{n}^\top \frac{\partial \mathbf{P}}{\partial u} = 0, \quad \text{and} \quad \mathbf{n}^\top \frac{\partial \mathbf{P}}{\partial v} = 0. \tag{14}$$

B.2 Horizontal Derivative ($\partial\mathcal{D}/\partial u$)

To find the horizontal depth gradient, we first differentiate the 3D coordinate vector $\mathbf{P}$ with respect to u (applying the product rule to X and keeping v constant):

$$\frac{\partial \mathbf{P}}{\partial u} = \begin{bmatrix} \frac{\partial X}{\partial u} \\ \frac{\partial Y}{\partial u} \\ \frac{\partial Z}{\partial u} \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{D}(u,v)}{f_x} + \frac{u - c_x}{f_x}\frac{\partial \mathcal{D}}{\partial u} \\ \frac{v - c_y}{f_y}\frac{\partial \mathcal{D}}{\partial u} \\ \frac{\partial \mathcal{D}}{\partial u} \end{bmatrix}. \tag{15}$$

Next, we enforce the orthogonality constraint from Eq. 14 by taking the dot product with the normal vector $\mathbf{n}$ and setting it to zero:

$$n_x \left(\frac{\mathcal{D}(u, v)}{f_x} + \frac{u - c_x}{f_x}\frac{\partial \mathcal{D}}{\partial u} \right) + n_y \left(\frac{v - c_y}{f_y}\frac{\partial \mathcal{D}}{\partial u} \right) + n_z \left(\frac{\partial \mathcal{D}}{\partial u} \right) = 0. \tag{16}$$

We then distribute n_x and group all terms containing the depth gradient $\frac{\partial \mathcal{D}}{\partial u}$:

$$n_x \frac{\mathcal{D}(u, v)}{f_x} + \frac{\partial \mathcal{D}}{\partial u} \left(n_x \frac{u - c_x}{f_x} + n_y \frac{v - c_y}{f_y} + n_z \right) = 0. \tag{17}$$

Subtracting the isolated term to the right side yields the critical negative sign:

$$\frac{\partial \mathcal{D}}{\partial u} \left(n_z + n_x \frac{u - c_x}{f_x} + n_y \frac{v - c_y}{f_y} \right) = -n_x \frac{\mathcal{D}(u, v)}{f_x}. \tag{18}$$

Finally, dividing by the bracketed term isolates the horizontal gradient:

$$\frac{\partial \mathcal{D}}{\partial u} = \frac{-n_x \mathcal{D}(u, v)/f_x}{n_z + n_x \frac{u - c_x}{f_x} + n_y \frac{v - c_y}{f_y}}. \tag{19}$$

B.3 Vertical Derivative ($\partial\mathcal{D}/\partial v$)

The derivation for the vertical gradient follows identical symmetry. We differentiate $\mathbf{P}$ with respect to v (holding u constant):

$$\frac{\partial \mathbf{P}}{\partial v} = \begin{bmatrix} \frac{u-c_x}{f_x}\frac{\partial \mathcal{D}}{\partial v} \\ \frac{\mathcal{D}(u,v)}{f_y} + \frac{v-c_y}{f_y}\frac{\partial \mathcal{D}}{\partial v} \\ \frac{\partial \mathcal{D}}{\partial v} \end{bmatrix}. \tag{20}$$

Enforcing the orthogonality constraint $\mathbf{n}^\top \frac{\partial \mathbf{P}}{\partial v} = 0$:

$$n_x\left(\frac{u-c_x}{f_x}\frac{\partial \mathcal{D}}{\partial v}\right) + n_y\left(\frac{\mathcal{D}(u,v)}{f_y} + \frac{v-c_y}{f_y}\frac{\partial \mathcal{D}}{\partial v}\right) + n_z\left(\frac{\partial \mathcal{D}}{\partial v}\right) = 0. \tag{21}$$

Grouping the $\frac{\partial \mathcal{D}}{\partial v}$ terms and solving yields the final vertical gradient:

$$\frac{\partial \mathcal{D}}{\partial v} = \frac{-n_y \mathcal{D}(u,v)/f_y}{n_z + n_x\frac{u-c_x}{f_x} + n_y\frac{v-c_y}{f_y}}. \tag{22}$$

B.4 Summary

Equations 19 and 22 form the analytic depth gradients utilized in the main paper. By substituting these relationships into our affine rectification model, we establish a robust mapping between predicted monocular disparity and the physical geometric constraints dictated by the rendered surface normals.

C Implementation Details

C.1 Per-Scene Settings

Table 11 lists the guided filter parameters and loss weights for every scene. The filter radius is given in pixels at the training resolution, and ϵ_{gf} is the regularization parameter of the guided filter. On MuSHRoom, the default weights ($\lambda_1 = 0.1$, $\lambda_2 = 0.08$, $\lambda_3 = 0.4$) produce floaters on `kokko` and `sauna`, so we scale all three weights by 0.5 and 0.25 on these scenes, respectively. Before training, each disparity map is clipped to its 1^{st} and 99^{th} percentiles and min-max normalized to $[0, 1]$.

Dataset	Scene	Radius (px)	ϵ_{gf}	λ_1	λ_2	λ_3
MuSHRoom	`coffee_room`	600	0.08	0.1	0.08	0.4
	`computer`	600	0.5	0.1	0.08	0.4
	`honka`	600	0.1	0.1	0.08	0.4
	`kokko`	550	0.4	0.05	0.04	0.2
	`sauna`	550	0.3	0.025	0.02	0.1
	`vr_room`	650	0.2	0.1	0.08	0.4
ScanNet++	`b20a261fdf`	600	0.1	0.025	0.02	0.1
	`8b5caf3398`	600	0.1	0.025	0.02	0.1

Table 11: Per-scene guided filter parameters and loss weights.

C.2 ScanNet++ Setup

We use the iPhone sequences of ScanNet++. Following the DN-Splatter data loader [24], every tenth posed frame is held out for evaluation and every fifth remaining frame is used for training, which gives 59 training images for `b20a261fdf` and 126 for `8b5caf3398`. We train at half resolution (960×720).

C.3 Compute

Our runs use a single NVIDIA RTX 3090 GPU. Training takes about 1.5 hours per MuSHRoom scene, and about 65 minutes per ScanNet++ scene at half resolution. The guided filter runs on the CPU (OpenCV).

D Definitions for Metrics

D.1 Depth Evaluation Metrics

To rigorously assess depth estimation performance on the ScanNet++ and MuSHRoom datasets, we adopt standard evaluation protocols established by prior work [24]. Our quantitative assessment relies on the following metrics:

- **Absolute Relative Error (Abs Rel):** Computes the mean ratio of the absolute depth difference to the corresponding ground-truth depth, reflecting the average magnitude of relative deviations across the scene.
- **Squared Relative Error (Sq Rel):** Similar to Abs Rel, but squares the relative depth differences before averaging, which more severely penalizes larger, localized depth errors.
- **Root Mean Squared Error (RMSE):** Calculates the square root of the mean squared differences between the predicted and actual depth maps, providing a direct measurement of the absolute error magnitude.
- **Logarithmic RMSE (RMSE log):** Computes the RMSE in the logarithmic domain. This metric is particularly advantageous for mitigating the disproportionate influence of distant background objects, which naturally exhibit larger depth values.
- **Threshold Accuracy ($\delta < t$):** Evaluates the percentage of predicted pixels where the maximum ratio between the predicted depth and ground-truth depth (or vice versa) falls below a predefined threshold factor t.

D.2 Mesh Evaluation Metrics

For 3D geometric fidelity, we compare our extracted 3D geometry against ground-truth meshes using widely accepted reconstruction metrics, with formal definitions provided in Table 12. To quantify the structural overlap between the predicted and actual geometry, we compute Precision, Recall, and the resulting F-score. Following standard practices, we enforce a strict distance threshold of 5 cm to determine whether a predicted vertex successfully matches the ground truth. Furthermore, to ensure a fair and relevant assessment, we strictly isolate our mesh quality evaluation to regions that fall within the visibility frustums of the training camera views.

Metric	**Mathematical Formulation**
Accuracy	$\frac{1}{\lvert Q\rvert}\sum_{\mathbf{q}\in Q}\min_{\mathbf{g}\in G}\lVert\mathbf{q}-\mathbf{g}\rVert_1$
Completion	$\frac{1}{\lvert G\rvert}\sum_{\mathbf{g}\in G}\min_{\mathbf{q}\in Q}\lVert\mathbf{q}-\mathbf{g}\rVert_1$
Chamfer-L_1	$\frac{1}{2}\left(\text{Accuracy}+\text{Completion}\right)$
Normal Accuracy	$\frac{1}{\lvert Q\rvert}\sum_{\mathbf{q}\in Q}\left(\mathbf{n}_{\mathbf{q}}^{\top}\mathbf{n}_{\mathbf{g}^*}\right)$ s.t. $\mathbf{g}^*=\arg\min_{\mathbf{g}\in G}\lVert\mathbf{q}-\mathbf{g}\rVert_1$
Normal Completion	$\frac{1}{\lvert G\rvert}\sum_{\mathbf{g}\in G}\left(\mathbf{n}_{\mathbf{q}^*}^{\top}\mathbf{n}_{\mathbf{g}}\right)$ s.t. $\mathbf{q}^*=\arg\min_{\mathbf{q}\in Q}\lVert\mathbf{q}-\mathbf{g}\rVert_1$
Normal Consistency	$\frac{1}{2}\left(\text{Normal Accuracy}+\text{Normal Completion}\right)$
Precision	$\frac{1}{\lvert Q\rvert}\sum_{\mathbf{q}\in Q}\mathbb{I}\left[\min_{\mathbf{g}\in G}\lVert\mathbf{q}-\mathbf{g}\rVert_1<\tau\right]$
Recall	$\frac{1}{\lvert G\rvert}\sum_{\mathbf{g}\in G}\mathbb{I}\left[\min_{\mathbf{q}\in Q}\lVert\mathbf{q}-\mathbf{g}\rVert_1<\tau\right]$
F-score	$\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$

Table 12: **Formal Definitions of Mesh Evaluation Metrics.** We denote the point clouds sampled from the predicted mesh and the ground-truth mesh as Q and G, respectively. For any given point $\mathbf{x}$, its corresponding surface normal vector is denoted as $\mathbf{n}_{\mathbf{x}}$. For classification metrics, $\mathbb{I}[\cdot]$ represents the indicator function, and the distance threshold is set to $\tau = 5\,\text{cm}$.

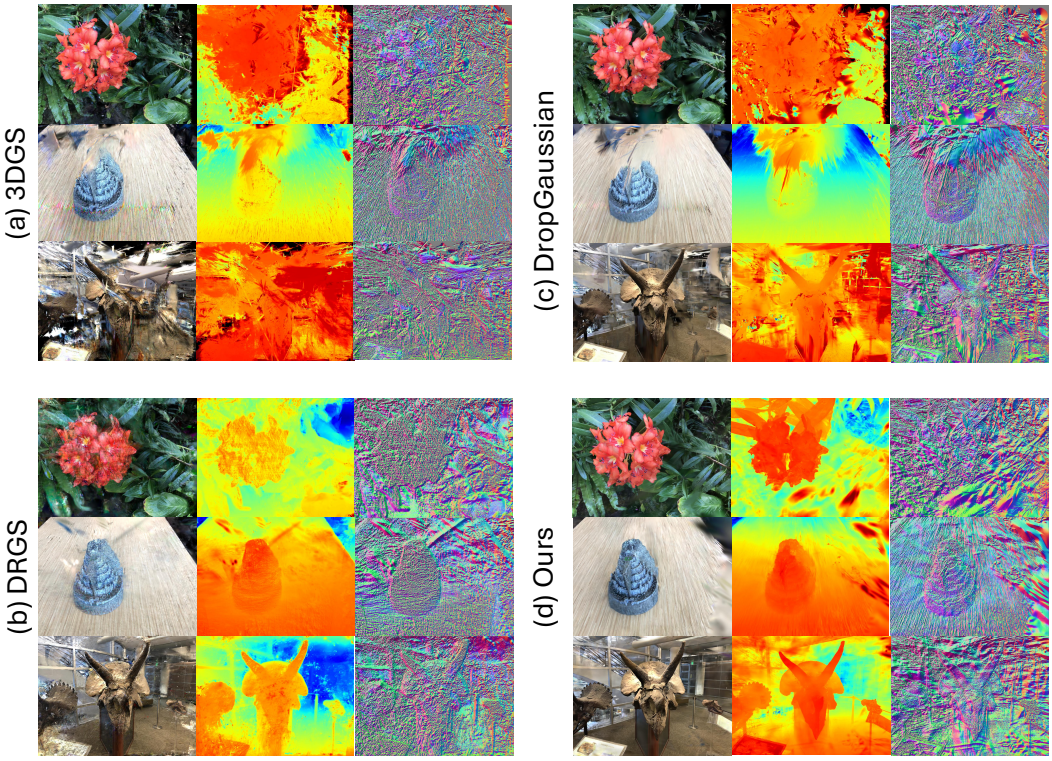


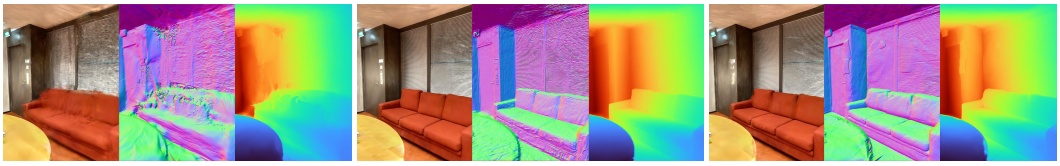
Figure 9: Additional qualitative comparison on NeRF-LLFF dataset. Visualized across the Flower, Fern, and Horns scenes under extreme view sparsity (rendered RGB, depth, and surface normal maps). (a) Unconstrained 3DGS severely overfits to the limited training views, resulting in massive floating artifacts and chaotic underlying geometry. (b) DRGS mitigates some artifacts via global depth alignment, but still suffers from noisy surface normals and blurred geometric details. (c) DropGaussian struggles to properly initialize in this sparse setting, leading to heavily corrupted backgrounds and incomplete structures. (d) In contrast, our proposed method leverages spatially rectified monocular priors to extract clean, highly accurate geometry with sharp object boundaries, smooth backgrounds, and highly consistent surface normals, all while preserving photorealistic appearance.

E Additional Qualitative Examples

E.1 Sparse-View Reconstruction

We present additional qualitative comparisons for novel-view synthesis under highly constrained sparse-view settings on the NeRF-LLFF dataset in Fig. 9. As observed in both the `Flower` and `Horns` scenes, the unconstrained 3DGS baseline severely overfits to the limited training views, resulting in massive clouds of floating artifacts and completely unstructured background geometry. While DropGaussian attempts to regularize the scene, its heavy reliance on an initial Structure-from-Motion (SfM) point cloud causes it to struggle under extreme sparsity, leading to incomplete object boundaries and persistent background noise. DRGS successfully reduces some floating artifacts through global depth alignment, but still exhibits geometric blurring and structural inaccuracies in complex regions.

In contrast, our proposed method leverages spatially rectified monocular priors to continuously enforce strict geometric constraints throughout the optimization. This results in dramatically cleaner surface extraction, sharp object boundaries, and smooth, artifact-free backgrounds, perfectly corroborating the significant quantitative improvements reported in Table 2 of the main text.



(a) DN-Splatter (Monodepth) (b) DN-Splatter (c) Ours

Figure 10: Additional qualitative comparison of rendered RGB, normals and depths on MuSHRoom dataset. While (a) the globally aligned DN-Splatter (Monodepth) baseline suffers from severe surface rippling and rendering artifacts, (c) our spatially rectified approach successfully recovers sharp, clean geometry that closely matches (b) the ground-truth sensor-supervised DN-Splatter.



(a) DN-Splatter (b) Ours (c) Ground truth

Figure 11: Additional qualitative mesh comparison on ScanNet++ dataset. (a) The DN-Splatter baseline exhibits high-frequency noise and artifacts across large planar surfaces like walls and ceilings. (b) Our proposed method successfully suppresses this noise to reconstruct much smoother, flatter surfaces that better approximate (c) the ground-truth geometry.

E.2 Novel-View Synthesis and Depth Estimation

We present additional qualitative comparisons on the MuSHRoom dataset in Fig. 10. We show that directly substituting physical sensor depth with uncorrected monocular priors leads to severe geometric degradation. The DN-Splatter (Monodepth) baseline, which relies on a single global scale-and-shift alignment, experiences catastrophic structural collapse. This is distinctly evident in the rendered normal maps: planar surfaces such as the back wall and the structure of the couch exhibit extreme geometric distortion. These underlying topological errors subsequently degrade the novel-view RGB rendering, resulting in blurred and warped textures. In contrast, our proposed method overcomes these limitations by applying spatially varying affine rectification to the exact same monocular prior. Notably, our sensor-free method reconstructs geometric and photometric fidelity that closely matches the original DN-Splatter, which possesses the massive advantage of physical sensor depth prior.

E.3 Mesh Evaluation

We provide additional qualitative comparisons for 3D mesh reconstruction on the ScanNet++ dataset in Fig. 11. We show that the mesh extracted by the DN-Splatter baseline exhibits noticeable high-frequency noise and irregular geometric artifacts, particularly across large planar surfaces such as walls and upper room structures. In contrast, our proposed method reconstructs significantly smoother and more consistent planar surfaces, visibly reducing these noisy geometric artifacts.

F Additional Quantitative Results

F.1 Comprehensive Analysis of Geometric Regularizations

In Table 13, we provide the complete quantitative ablation covering novel-view synthesis, depth estimation, and 3D mesh reconstruction metrics. The unconstrained Splatfacto baseline exhibits severe geometric drift, yielding a depth RMSE of 0.196 and a low Normal Consistency of 0.779. Introducing our metric depth re-alignment ($\mathcal{L}_{\text{depth}}$) directly addresses global scale ambiguity, evidenced by a sharp decrease in depth RMSE (to 0.147) and an increase in the inlier ratio $\delta < 1.25$ (from 0.937 to 0.984). However, global scale correction alone is insufficient for capturing high-frequency geometric details, resulting in a modest Normal Consistency of 0.793. Conversely, applying normal-aware supervision ($\mathcal{L}_{\text{normal}}$) or gradient-domain regularization ($\mathcal{L}_{\text{grad}}$) in isolation fails to anchor the global scene geometry; in fact, using solely $\mathcal{L}_{\text{grad}}$ degrades global depth metrics (Abs Rel of 0.085) because it only supervises relative spatial slopes without an absolute metric anchor. The true efficacy of our approach emerges when these complementary constraints are jointly applied. The $\mathcal{L}_{\text{depth}}$ term rigidly anchors the global metric scale, while $\mathcal{L}_{\text{grad}}$ and $\mathcal{L}_{\text{normal}}$ enforce precise local spatial gradients and physical surface orientations. This coupled formulation (All) achieves the lowest overall depth RMSE (0.139) and the highest Normal Consistency (0.891) and mesh F-score (0.855). This extensive evaluation confirms that our joint regularization successfully resolves the fundamental ambiguity between view-dependent appearance and underlying physical structure, exchanging a marginal drop in SSIM (0.869 vs. 0.858) for a highly robust, artifact-free 3D representation. We present qualitative comparisons in Fig. 12.

Method	**Photometry**			**Depth Estimation**				**Mesh Reconstruction**				
	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	Chamfer-L_1 ↓	Norm. Cons. ↑	F-score ↑
Splatfacto	25.293	0.869	0.086	0.060	0.021	0.196	0.937	0.043	0.039	0.041	0.779	0.728
+ $\mathcal{L}_{\text{grad}}$	24.323	0.848	0.123	0.085	0.045	0.262	0.889	0.055	0.051	0.053	0.857	0.673
+ $\mathcal{L}_{\text{normal}}$	25.257	0.867	0.085	0.064	0.022	0.205	0.926	0.043	0.039	0.041	0.797	0.728
+ $\mathcal{L}_{\text{depth}}$	24.561	0.863	0.098	0.043	0.011	0.147	0.984	0.031	0.039	0.035	0.793	0.799
+ $\mathcal{L}_{\text{grad}}$ + $\mathcal{L}_{\text{normal}}$	24.553	0.851	0.117	0.075	0.034	0.238	0.902	0.051	0.047	0.049	0.868	0.707
+ $\mathcal{L}_{\text{grad}}$ + $\mathcal{L}_{\text{depth}}$	25.013	0.860	0.101	0.041	0.010	0.141	0.983	**0.026**	**0.028**	**0.027**	0.885	0.850
+ $\mathcal{L}_{\text{normal}}$ + $\mathcal{L}_{\text{depth}}$	**25.419**	**0.869**	**0.084**	**0.040**	**0.010**	0.141	**0.987**	0.029	0.032	0.031	0.819	0.814
All (Ours)	24.947	0.858	0.103	**0.040**	**0.010**	**0.139**	0.985	**0.026**	**0.028**	**0.027**	**0.891**	**0.855**

Table 13: Comprehensive ablations on proposed geometric regularization components. Evaluated on the `coffee_room` scene from the MuSHRoom dataset. Full quantitative evaluation of the individual and combined contributions of our proposed geometric loss terms. We report photometric (novel-view synthesis), depth estimation, and 3D mesh reconstruction metrics against the unconstrained Splatfacto baseline. Our full configuration (*All*) provides the best overall balance, enforcing strict structural integrity (highest mesh F-score and Normal Consistency) while maintaining competitive rendering quality. Best results in each column are indicated in **bold**.

F.2 Sensitivity to Monocular Depth Priors

In Table 14, we provide the complete scene-wise quantitative evaluation of our method across three state-of-the-art monocular depth models: Depth-Anything-V2 [27], MoGe [25], and Metric3D-V2 [10]. While the average metrics demonstrate the general robustness of our approach, the scene-by-scene breakdown reveals critical insights into prior network stability. Both Depth-Anything-V2 and MoGe demonstrate exceptional consistency across diverse indoor topologies. For instance, in the highly textured `coffee_room` scene, both models facilitate outstanding mesh F-scores (0.855 and 0.879, respectively) and near-perfect inlier ratios ($\delta < 1.25$ over 0.98).

Conversely, the performance of Metric3D-V2 exhibits high variance, largely due to an architectural constraint: Metric3D-V2 utilizes a fixed wide-aspect-ratio inference canvas. Because the MuSHRoom dataset consists of portrait-oriented captures, feeding these images into the network requires massive aspect-ratio padding, which systematically induces severe depth and disparity artifacts near the image borders. Consequently, in challenging environments like the `computer` and `vr_room` scenes—where these border artifacts intersect with thin structures and complex occlusion boundaries—Metric3D-V2 suffers catastrophic structural degradation, yielding F-scores of just 0.428 and 0.458, respectively. This scene-specific collapse heavily penalizes its average performance. These comprehensive results validate that while our spatially varying affine rectification successfully bridges the gap between

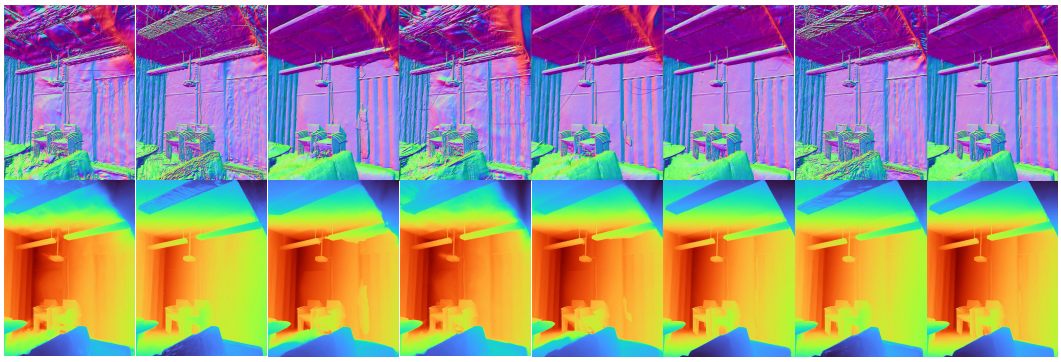


Figure 12: Qualitative comparisons of our proposed geometric regularization on MuSHRoom dataset. Visualized via rendered normals (top row) and depth maps (bottom row). **(a)** The unconstrained baseline (All off) exhibits severe geometric drift, yielding highly chaotic and noisy surface normals. **(b-d)** Applying individual loss terms in isolation yields incomplete geometry: while $\mathcal{L}_{\text{depth}}$ anchors global scale, it fails to resolve high-frequency structural noise, and while $\mathcal{L}_{\text{grad}}$ or $\mathcal{L}_{\text{normal}}$ improve local structure, they struggle without an absolute metric anchor. **(h)** Our full formulation combines these constraints. By anchoring the global metric scale while explicitly enforcing local spatial gradients and physical orientations, the complete approach reconstructs exceptionally clean planar surfaces (*e.g.,*, the walls and ceiling), sharp structural boundaries, and globally consistent depth.

relative and metric depth, the ultimate geometric fidelity of the 3DGS representation remains bounded by the structural reliability and boundary-awareness of the chosen foundation prior.

Depth Prior	Scene	Photometry			Depth Estimation				Mesh Reconstruction				
		PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	Chamfer-L_1 ↓	Norm. Cons. ↑	F-score ↑
Depth-Anything-V2	coffee_room	24.947	0.858	0.103	0.040	0.010	0.139	0.985	0.026	0.028	0.027	0.891	0.855
	computer	24.968	0.851	0.129	0.070	0.026	0.254	0.955	0.065	0.058	0.061	0.884	0.603
	honka	23.142	0.821	0.140	0.050	0.017	0.175	0.973	0.047	0.034	0.041	0.885	0.799
	kokko	22.444	0.831	0.163	0.064	0.025	0.220	0.933	0.051	0.042	0.047	0.798	0.705
	sauna	24.443	0.830	0.187	0.060	0.015	0.167	0.978	0.051	0.044	0.047	0.874	0.687
	vr_room	26.144	0.855	0.163	0.045	0.013	0.153	0.981	0.081	0.057	0.069	0.866	0.608
	Average	24.348	0.841	0.148	0.055	0.018	0.185	0.968	0.054	0.044	0.049	0.866	0.710
MoGe	coffee_room	25.088	0.863	0.097	0.039	0.010	0.139	0.984	0.026	0.026	0.026	0.894	0.879
	computer	24.569	0.848	0.137	0.091	0.038	0.293	0.915	0.074	0.083	0.079	0.860	0.626
	honka	23.171	0.820	0.144	0.051	0.016	0.166	0.973	0.035	0.030	0.033	0.886	0.822
	kokko	22.197	0.825	0.163	0.072	0.031	0.246	0.913	0.054	0.048	0.051	0.793	0.675
	sauna	24.445	0.828	0.179	0.066	0.017	0.178	0.967	0.047	0.045	0.046	0.874	0.687
	vr_room	26.301	0.861	0.150	0.046	0.014	0.164	0.977	0.062	0.050	0.056	0.883	0.649
	Average	24.295	0.841	0.145	0.061	0.021	0.198	0.955	0.050	0.047	0.048	0.865	0.723
Metric3D-V2	coffee_room	24.429	0.850	0.117	0.053	0.015	0.177	0.958	0.057	0.056	0.056	0.845	0.673
	computer	21.819	0.804	0.225	0.237	0.215	0.621	0.626	0.190	0.261	0.226	0.763	0.428
	honka	22.696	0.815	0.157	0.068	0.022	0.204	0.937	0.058	0.060	0.059	0.849	0.673
	kokko	21.797	0.812	0.187	0.095	0.053	0.320	0.877	0.106	0.074	0.090	0.749	0.508
	sauna	23.102	0.796	0.241	0.150	0.136	0.380	0.811	0.087	0.103	0.095	0.831	0.556
	vr_room	25.518	0.847	0.180	0.075	0.029	0.242	0.907	0.131	0.101	0.116	0.799	0.458
	Average	23.227	0.821	0.184	0.113	0.078	0.324	0.853	0.105	0.109	0.107	0.806	0.549

Table 14: Ablation on monocular depth models. Evaluated across all six scenes of the MuSHRoom dataset test split. We report full photometric, depth estimation, and 3D mesh reconstruction metrics for our proposed method when supervised by three different foundation models (Depth-Anything-V2, MoGe, and Metric3D-V2). While our affine rectification effectively extracts metric geometry regardless of the prior, high-quality estimators like MoGe and Depth-Anything-V2 maintain significantly higher consistency across challenging scenes.

F.3 Sensitivity to Filters

Raw outputs from monocular depth networks, while structurally strong, inevitably contain local noise, quantization artifacts, and high-frequency inconsistencies. If injected directly into the optimization pipeline without regularization, these noisy coefficients force the 3D Gaussians to overfit to erroneous local depth shifts, severely degrading both the extracted geometry and the novel view synthesis. To address this, we apply a filtering mechanism to smooth the extracted alignment coefficients. In this section, we ablate our choice of the Guided Filter against a standard Gaussian filter and a baseline with no filtering.

In Table 15, we perform this ablation on the `coffee_room` scene from the MuSHRoom dataset using the Depth-Anything-V2 prior. We keep all loss terms active (RGB + Disparity Normal Gradient + L_1 Depth) and isolate the effect of the filter type on the disparity alignment coefficients. The results are evaluated across Novel View Synthesis (`right` split), Depth estimation (`right` split), and Mesh extraction.

We show that disabling the filter entirely ("None") results in a catastrophic drop in geometric accuracy; the F-score collapses to 0.680, and the depth RMSE spikes to 0.269, proving that unregularized monocular noise actively harms the spatial reconstruction. While applying a standard Gaussian blur mitigates some of this noise and improves the F-score to 0.774, it does so indiscriminately. Because a Gaussian filter smooths across physical depth boundaries, it inadvertently blurs distinct structural edges, limiting the sharpness of the final geometry. In contrast, our chosen Guided Blur leverages the prior itself as an edge-preserving guide. This ensures that noise is smoothed across flat surfaces while sharp structural discontinuities are strictly preserved. As a result, our Guided Blur strategy firmly dominates across all categories, yielding the highest photometric fidelity, the lowest depth error, and a significantly superior geometric F-score of 0.855.

Filter Type	Novel View Synthesis			Depth Evaluation			Mesh Evaluation				
	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	RMSE ↓	δ_1 ↑	Accuracy ↓	Comp. ↓	Chamfer-L_1 ↓	Norm. Cons. ↑	F-score ↑
None	20.66	0.767	0.231	0.080	0.269	0.858	0.0530	0.0530	0.0551	0.8592	0.6804
Gaussian Blur	21.29	0.770	0.206	0.038	0.125	0.986	0.0338	0.0347	0.0343	0.8786	0.7737
Guided Blur (Ours)	**21.53**	**0.787**	**0.195**	**0.031**	**0.112**	**0.991**	**0.0258**	**0.0278**	**0.0268**	**0.8908**	**0.8551**

Table 15: Ablation of prior filtering strategies. Evaluated on the `coffee_room` scene using the Depth-Anything-V2 prior. Novel-view synthesis and depth metrics are reported on the `right` (disjoint-trajectory) split, whereas Tables 6 and 14 report the `left` split. Removing the filter results in severe geometric degradation due to raw monocular noise. While a standard Gaussian filter mitigates some noise, it indiscriminately blurs across physical boundaries. Our Guided Blur strategy uniquely preserves sharp structural edges, yielding the best performance across all photometric, depth, and geometric metrics.

F.4 Sensitivity to Gradient Matching Formulation

We evaluate the sensitivity of our approach to the specific mathematical formulation of the disparity gradient loss defined in Eq. 5. Using the `coffee_room` scene and the Depth-Anything-V2 prior, we ablate our combined magnitude and directional alignment formulation (MSE + Cosine) against using strictly Mean Squared Error (MSE) or strictly Cosine Similarity. The objective of this loss term is to ensure that the spatial gradients of the rendered depth map closely follow the structural gradients of the monocular prior.

The quantitative results in Table 16 demonstrate that relying exclusively on an MSE formulation is insufficient for complex geometries. MSE alone struggles to enforce sharp orientational changes, resulting in a significant drop in Normal Consistency (0.791) and overall F-score (0.781). While Cosine Similarity alone provides a highly effective directional cue—matching our combined approach in absolute depth accuracy—it lacks a constraint on the actual magnitude of the gradient change. By combining both MSE (magnitude) and Cosine (direction), we ensure the most robust structural regularization. This combined approach prevents local geometric collapse and yields the best overall surface extraction (F-score of 0.855) alongside optimal photometric rendering quality (21.53 dB PSNR).

F.5 Effect of the Meshing Algorithm

Mesh metrics depend strongly on the algorithm used to extract the mesh. Table 17 compares Poisson reconstruction and TSDF fusion applied to the same trained models on MuSHRoom. TSDF fusion gives better meshes for all three methods on every metric; for DN-Splatter with sensor depth, it halves the Chamfer distance and raises the F-score by 0.11. Comparisons between methods in Tables 4 and 5 should therefore be read together with their Algorithm column. The numbers in Table 17 come from our reruns with the released code, so they differ slightly from those in Table 4.

Gradient Loss	Novel View Synthesis			Depth Evaluation			Mesh Evaluation				
	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	RMSE ↓	δ_1 ↑	Accuracy ↓	Comp. ↓	Chamfer-L_1 ↓	Norm. Cons. ↑	F-score ↑
MSE	20.39	0.776	0.226	0.055	0.157	0.959	0.0328	0.0400	0.0364	0.7914	0.7815
Cosine	21.44	0.786	**0.195**	**0.030**	**0.110**	**0.991**	0.0259	**0.0278**	0.0269	**0.8911**	0.8535
MSE + Cosine (Ours)	**21.53**	**0.787**	**0.195**	0.031	0.112	**0.991**	**0.0258**	**0.0278**	**0.0268**	0.8908	**0.8551**

Table 16: Ablation on gradient matching formulation. Evaluated on the `coffee_room` scene. Novel-view synthesis and depth metrics are reported on the `right` (disjoint-trajectory) split. Relying solely on MSE fails to capture sharp orientational changes, leading to degraded Normal Consistency. While Cosine similarity provides excellent directional cues, combining both magnitude (MSE) and directional (Cosine) alignment yields the highest overall surface extraction F-score and best novel view synthesis.

Method	Algorithm	Accuracy ↓	Completion ↓	Chamfer-L_1 ↓	Normal Cons. ↑	F-score ↑
DN-Splatter	Poisson	.0405	.0421	.0413	.8322	.8299
	TSDF	.0225	.0202	.0213	.8847	.9381
DN-Splatter (Monodepth)	Poisson	.2204	.4045	.3125	.6778	.2280
	TSDF	.2162	.3397	.2780	.7095	.2769
Ours†	Poisson	.0723	.0656	.0690	.8197	.5883
	TSDF	.0607	.0445	.0526	.8795	.7117

Table 17: Poisson reconstruction versus TSDF fusion on the same trained models (MuSHRoom). DN-Splatter and DN-Splatter (Monodepth) are averaged over all six scenes. †Averaged over `coffee_room`, `computer`, `honka` and `vr_room`.

G Baselines

G.1 Comprehensive Analysis of the DN-Splatter (Monodepth) Baseline

To fully demonstrate the limitations of global alignment strategies when utilizing monocular depth priors, we provide a comprehensive scene-wise breakdown of the DN-Splatter (Monodepth) baseline on the MuSHRoom dataset (Table 18). While DN-Splatter yields exceptional results when provided with ground-truth sensor depth, its geometric integrity collapses when forced to rely on uncorrected monocular priors using a global scale-and-shift alignment.

The scene-wise metrics reveal high volatility and catastrophic failures in structurally complex environments. While simpler or more uniformly constrained scenes like `coffee_room` and `honka` manage to retain a loose semblance of geometry (F-scores of 0.516 and 0.409, respectively), environments with high depth variance and complex occlusion boundaries suffer from complete topological collapse. Specifically, in the `computer` and `kokko` scenes, the baseline yields extreme depth errors (RMSE > 1.95) and effectively zero structural reconstruction, plummeting to F-scores of 0.065 and 0.027. This erratic behavior perfectly illustrates the piecewise-linear fragmentation of foundation depth models discussed in our main text. Because monocular distortions vary spatially, forcing a single global linear alignment causes severe warping—optimizing to fit one local region inherently breaks the geometry of the rest of the scene. These results empirically validate our core motivation: to successfully harness monocular priors for 3DGS, spatially varying local rectification is strictly required.

Scene	Photometry (Left / Right)			Depth Estimation (Left / Right)				Mesh Reconstruction				
	PSNR ↑	SSIM ↑	LPIPS ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Acc. ↓	Comp. ↓	Chamfer-L_1 ↓	Norm. Cons. ↑	F-score ↑
coffee_room	22.71 / 19.74	0.788 / 0.727	0.187 / 0.288	0.088 / 0.086	0.033 / 0.033	0.245 / 0.232	0.862 / 0.903	0.076	0.084	0.080	0.792	0.516
computer	13.60 / 13.27	0.637 / 0.602	0.495 / 0.506	0.744 / 0.635	1.791 / 1.875	2.273 / 2.541	0.100 / 0.242	0.276	0.666	0.471	0.641	0.065
honka	21.38 / 19.40	0.755 / 0.705	0.218 / 0.272	0.101 / 0.091	0.047 / 0.028	0.275 / 0.244	0.877 / 0.926	0.124	0.109	0.117	0.811	0.409
kokko	12.81 / 11.38	0.617 / 0.592	0.537 / 0.585	0.823 / 0.921	1.649 / 2.256	1.958 / 2.532	0.046 / 0.009	0.445	0.689	0.567	0.524	0.027
sauna	20.95 / 18.52	0.745 / 0.718	0.329 / 0.373	0.284 / 0.356	0.319 / 0.443	0.596 / 0.708	0.619 / 0.545	0.136	0.240	0.188	0.724	0.316
vr_room	24.12 / 21.16	0.811 / 0.763	0.236 / 0.288	0.085 / 0.113	0.034 / 0.076	0.264 / 0.358	0.906 / 0.866	0.151	0.129	0.140	0.764	0.326
Average	19.26 / 17.24	0.725 / 0.684	0.334 / 0.385	0.354 / 0.367	0.645 / 0.785	0.935 / 1.103	0.568 / 0.582	0.201	0.320	0.261	0.709	0.277

Table 18: Comprehensive scene-wise evaluation of the DN-Splatter (Monodepth) baseline. Evaluated across the MuSHRoom dataset test split. Because this baseline relies on a single global scale-and-shift transform to align monocular depth, it is highly susceptible to piecewise-linear distortions. This results in severe structural collapse on complex scenes (e.g., `computer` and `kokko`), validating the necessity of our proposed spatially varying affine rectification.

H Limitations

While our proposed spatially varying affine rectification significantly improves the geometric fidelity of 3DGS, our approach has several limitations:

Dependency on Foundation Model Reliability. Because our method relies on extracting geometric cues from monocular depth estimators, our reconstruction quality is inherently bounded by the reliability of the chosen foundation model. As discussed in Appendix Sec. F.2, if the monocular prior completely fails to capture the underlying structure—such as in the presence of highly transparent surfaces, massive specular reflections (mirrors), or severe domain gaps not covered by the prior's training data—our affine rectification cannot recover the lost geometry.

Absence of Absolute Physical Scale. Our method bridges the gap toward sensor-supervised geometric fidelity by enforcing consistent Euclidean geometry across the scene. However, because it operates entirely without physical sensor data, the final reconstructed mesh still lacks an absolute real-world physical scale. The global metric anchor ($\mathcal{L}_{\text{depth}}$) aligns the geometry to the arbitrary coordinate space established by the initial SfM camera poses, rather than true physical dimensions. Moreover, because $\mathcal{L}_{\text{depth}}$ is re-aligned to the current rendering at every step, it cannot correct a global scale error that is already present in the photometric reconstruction. On the ScanNet++ scene `b20a261fdf`, both plain Splatfacto and our method render depth about 5% shorter than the sensor depth. At the 2–4 m viewing distances of this scene, this amounts to 10–20 cm, beyond the 5 cm F-score threshold, which limits our F-score on this scene to 0.18, compared with 0.46 on `8b5caf3398`.

Sensitivity to Regularization Strength. The default loss weights are too strong for some scenes. On `kokko` and `sauna`, they produce floaters, including directly in front of the camera, and we reduce all three weights by factors of 0.5 and 0.25, respectively (Appendix Sec. C).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are empirically verified in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Appendix Sec. H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We present full derivations in Appendix Sec. B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all implementation details in Sec. 4, and the per-scene settings in Appendix Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide code and instructions for the final paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide complete experimental details in Sec. 4 and Appendix Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: While we do not report explicit error bars or standard deviations to maintain table readability, we ensure the statistical reliability of our most sensitive experiments (the sparse-view evaluations) by running and averaging all reported metrics over 10 random seed selections (as detailed in Sec. 4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the GPU and training times in Appendix Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and understood the code of ethics and have made every effort to adhere to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release data or models so there is no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available dataset and pretrained models where all of them are properly cited in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release our codebase with detailed README files.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not use crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not use crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We used LLM solely for editing assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.