UniSHARP: Universal Sharp Monocular View Synthesis

Meixi Song
Insta360 Research
songmeixi@insta360.com

Dizhe Zhang*
Insta360 Research
zhangdizhe@insta360.com

Ruiyang Zhang
Insta360 Research
ruiyangzhang@insta360.com

Bo Du
Wuhan University
dubo@whu.edu.cn

Ming-Hsuan Yang
University of California, Merced
myang37@ucmerced.edu

Lu Qi*
Insta360 Research, Wuhan University
gavinqi@insta360.com



Figure 1: UniSHARP performs monocular novel view synthesis across diverse camera types. Given a single image from a perspective, wide-FoV, fisheye, or panoramic camera, UniSHARP predicts a 3D Gaussian point cloud and renders high-quality novel views.

Abstract

In this work, we focus on extending SHARP, the popular photorealistic view synthesis method, for universal monocular rendering across a continuum of camera systems, from conventional perspective cameras to wide-field-of-view, fisheye and omnidirectional panoramic settings. To overcome the pinhole-specific assumptions of SHARP, our key idea is to align various images in a unified omnidirectional latent space. Thus, we propose UniSHARP, which performs implicit alignment in both feature and Gaussian spaces. Specifically, Gaussian primitives are arranged along rays and radial distances in a ray-based universal representation, while 2D semantic and 3D spatial features extracted from UniK3D-inspired encoders are jointly decoded to generate the complete Gaussian cloud. To comprehensively evaluate our method, we construct a benchmark covering diverse imaging systems across various scenes. The benchmark is further stratified by field of view (FoV) to enable fine-grained assessment of the universal monocular rendering task. Extensive experiments on the proposed benchmark demonstrate the effectiveness of UniSHARP, outperforming alternative methods by a large margin. The project page can be found at: https://insta360-research-team.github.io/Unisharp-website/.

*Indicates corresponding author.

Preprint.

1 Introduction

Novel view synthesis is a fundamental capability for spatial visual intelligence, enabling captured images to support robotic navigation, AR/VR interaction, immersive telepresence, and 3D content creation [1–9]. Albeit the success achieved by neural radiance fields (NeRF) [10] and 3D Gaussian Splatting (3DGS) [11], the monocular view synthesis remains ill-posed due to the severe spatial information loss inherent in a single image.

Recent monocular 3DGS methods, such as SHARP [12] and Flash3D [13], learn feedforward Gaussian priors from perspective image collections and regress renderable primitives from a single input. However, trained primarily on narrow-FoV perspective images, these regressors fail to generalize to diverse imaging systems [14–18], including wide-FoV, fisheye, and panoramic cameras. These practical constraints motivate universal monocular novel view synthesis, where a model must infer 3D structure, visibility, and appearance from a single image while remaining applicable to heterogeneous camera models.

To address this issue, an intuitive solution is to fine-tune SHARP on wide-FoV or panoramic images. However, since SHARP maps every pixel in normalized space under the pinhole camera assumption, it inherently fails to predict geometry in non-pinhole domain. Another approach is to re-project images into multiple views, but this introduces additional computational overhead and requires extra processing to handle stitching artifacts. Thus, one question raised: how can widely used methods such as SHARP be adapted to diverse camera systems in a simple manner?

Inspired by panoramic vision [19–24], we propose UniSHARP, which extends the popular SHARP framework for universal monocular view synthesis via a unified omnidirectional representation across diverse camera systems. Specifically, UniSHARP performs implicit alignment in the latent rather than the image space along two dimensions. On one hand, a ray-based universal representation organizes Gaussian primitives along rays and radial distances, enabling initialization and refinement in a shared 3D space rather than projection-specific image coordinates. On the other hand, a unified feature space decodes both 2D semantic embeddings and 3D spatial features, providing complementary appearance and geometry priors for single-image reconstruction. With such minor modifications, the robustness of UniSHARP to diverse camera systems is enhanced.

To well evaluate our performance, we build a comprehensive benchmark that collects narrow perspective, wide-FoV, fisheye, and panoramic validation data across real-world and simulated scenes. The benchmark combines established validation datasets with OmniRooms, our AirSim-based indoor panoramic dataset, and its projected wide-FoV variant. This FoV-stratified design enables controlled analysis of how rendering quality changes as the camera FoV increases from 60° to 360°.

Our contributions are summarized as follows:

- We propose UniSHARP, a universal-camera feedforward 3DGS framework for monocular novel view synthesis across standard perspective, wide-FoV, fisheye, and panoramic inputs.
- We develop a feature-space Gaussian prediction pipeline that fuses 2D semantic encodings with 3D spatial features and allocates Gaussians at native input resolution, preserving geometric priors and high-frequency image details without camera-specific resizing.
- We introduce panoramic-specific adaptations, including spherical Gaussian initialization and distortion-aware probabilistic dropout, to regularize Gaussian distributions under severe equirectangular projection distortion.
- We introduce a FoV-stratified benchmark spanning perspective, wide-FoV, fisheye, and panoramic cameras. We validate UniSHARP on this benchmark, demonstrating state-of-the-art rendering quality and strong cross-camera generalization.

2 Related work

Multi-image novel view synthesis. Multi-image NVS reconstructs scenes from several posed or nearby observations by exploiting cross-view consistency. Neural radiance fields established continuous scene representations for photorealistic rendering [10], while later anti-aliased variants improved unbounded and high-resolution reconstruction [25, 26]. 3D Gaussian Splatting replaced expensive volume rendering with explicit primitives and real-time rasterization [11]. Feedforward

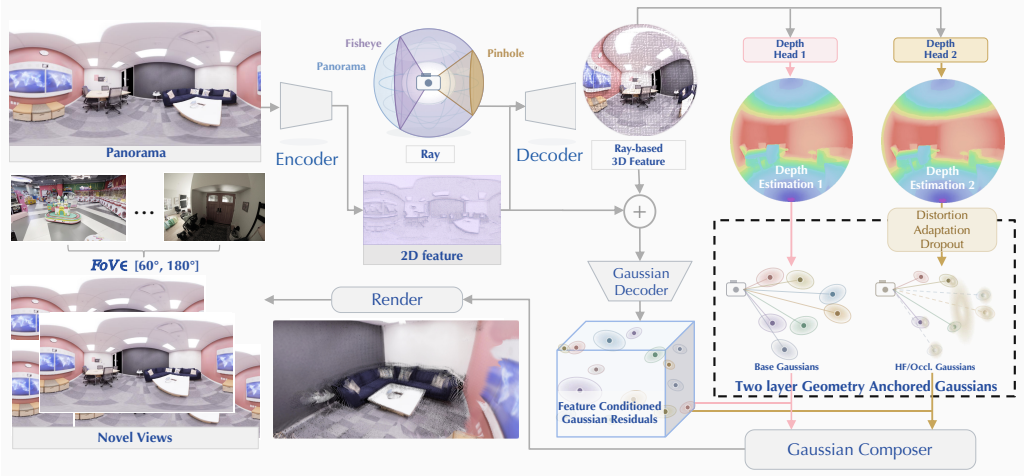

Figure 2: UniSHARP pipeline for universal-camera monocular novel view synthesis. Given a single source image, UniSHARP estimates ray-distance geometry and multi-scale features, initializes two-layer Gaussians in ray-distance space, predicts Feature Conditioned Gaussian residuals, and renders target views with the unified Gaussian representation across perspective, wide-FoV, fisheye, and panoramic cameras.

methods further reduce per-scene optimization through learned image-based rendering and multi-view stereo cost volumes [27, 28], while recent sparse-view Gaussian models predict 3DGS representations from image pairs or posed image sets [2, 3, 29]. Large reconstruction models and pose-free systems extend this trend to larger baselines or unconstrained captures [30–33]. Despite their strong quality, these methods fundamentally depend on camera poses, correspondences, or multi-view feature aggregation. UniSHARP instead targets the strictly monocular setting and predicts a renderable Gaussian representation from a single image.

Single-image novel view synthesis. Single-image NVS must infer geometry, appearance, and visibility from priors rather than direct triangulation. Early learning-based approaches predicted neural radiance fields or multiplane images from a single input [34, 35], and other methods used layered geometry, inpainting, or adaptive MPI layouts to handle disocclusion [36–39]. Larger reconstruction and synthesis models show that strong feedforward networks can infer plausible 3D structure from a single photograph [40, 41]. More recently, monocular 3DGS methods directly regress explicit Gaussian primitives, enabling efficient rendering from single images [12, 13, 42]. Generative methods improve extrapolation to larger camera motion through diffusion priors [43–45], but they often trade rendering speed or geometric explicitness for generative flexibility. These works establish single-image NVS as a compelling direction, yet they primarily assume perspective imagery. UniSHARP extends monocular 3DGS to a unified camera setting covering perspective, wide-FoV, fisheye, and panoramic inputs.

Wide-FoV novel view synthesis. Wide-FoV NVS introduces projection distortion, nonuniform angular sampling, and nontrivial boundary topology that are absent in perspective images. Universal monocular geometry estimators show that ray-based representations are important for handling arbitrary cameras [46]. In reconstruction, recent 3DGS systems adapt the rasterizer or camera model to omnidirectional and fisheye inputs [14, 47], and self-calibrating variants jointly optimize camera models, poses, and Gaussian scenes for wide-FoV captures [18, 48, 49]. In parallel, panoramic feedforward methods use spherical radiance fields, spherical cost volumes, Gaussian pyramids, or Yin-Yang grids for 360-degree synthesis [15–17, 50]. These methods address important non-perspective settings, but they commonly rely on optimized scenes, calibrated captures, or multiple panoramic observations. UniSHARP instead unifies perspective, wide-FoV, fisheye, and panoramic monocular NVS in a single feedforward 3DGS model.

3 Method

Given a single source image $I_s \in \mathbb{R}^{3\times H\times W}$, UniSHARP predicts a set of 3D Gaussian primitives to enable high-fidelity novel-view synthesis. Unlike conventional feedforward methods that

regress Gaussian attributes in the image plane, UniSHARP decouples camera projection from scene representation via a unified ray-distance space. This is achieved by: (1) introducing a ray-based universal representation for heterogeneous cameras (Sec. 3.1); (2) composing the scene using Geometry Anchored Gaussians and Feature Conditioned Gaussian residuals (Sec. 3.2); (3) employing a mixed-camera training strategy to supervise rendering across diverse projection types (Sec. 3.3).

3.1 Ray Based Universal Representation

Standard image-plane coordinates fail to generalize across heterogeneous projections because a single-pixel displacement corresponds to vastly different angular changes in perspective, fisheye, or panoramic views. To address this, UniSHARP adopts a unified ray-distance space inspired by UniK3D [46]. By decoupling viewing direction from scene range, we ensure that Gaussian primitives defined by 3D centers, covariances, and appearance are optimized in a consistent metric space instead of being tied to projection-specific image grids.

Formally, let $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$ denote the pixel domain. UniSHARP uses a predicted per-pixel unit ray field

$$\mathbf{r}_p \in \mathbb{S}^2, \qquad \|\mathbf{r}_p\|_2 = 1, \tag{1}$$

and a radial distance $d_p > 0$ from the camera center. The corresponding 3D point is then $\mathbf{x}_p = d_p \mathbf{r}_p$.

This formulation provides a universal coordinate system where Gaussian attributes (placement, scale, and color) are defined consistently across diverse camera models. By measuring spatial footprints along rays rather than in the rasterized plane, UniSHARP enables robust initialization and refinement of Gaussian scenes regardless of the input projection type.

3.2 Model Design

Building upon the unified ray system, UniSHARP first constructs Geometry Anchored Gaussians (3.2.1), providing a camera-unified gaussian space initialization. It then predicts Feature Conditioned Gaussian residuals (3.2.2) from 2D semantic and 3D ray-based features, composing them with the anchors to obtain the final renderable Gaussians. The overall pipeline is illustrated in Figure 2.

3.2.1 Geometry Anchored Gaussians

For each input image, we construct two-layer Geometry Anchored Gaussians on a native ray grid. Let H_g and W_g be the Gaussian grid resolution. At pixel p and layer ℓ, each Geometry Anchored Gaussian is represented as

$$\mathcal{B}_{p,\ell} = \left\{ \mathbf{r}_p, \rho_{p,\ell}, \mathbf{s}^0_{p,\ell}, \mathbf{q}^0, \mathbf{c}^0_p, \alpha^0_\ell \right\}, \tag{2}$$

where $\mathbf{r}_p$ is the unit ray, $\rho_{p,\ell}$ is inverse radial distance, $\mathbf{s}^0_{p,\ell}$ is the base scale, $\mathbf{q}^0$ is the identity quaternion, $\mathbf{c}^0_p$ is obtained from the input color, and α^0_ℓ is opacity initialization. The first layer aligns with the visible surface, while the second layer captures disocclusions and high-frequency structures beyond a single surface hypothesis. This second radial layer is predicted by an additional depth head same as the UniK3D radial head, giving the geometry-anchored representation a separate distance hypothesis that can specialize through rendering supervision and regularization.

By allocating primitives according to native ray sampling, UniSHARP enables resolution-adaptive construction that preserves angular consistency and high-frequency details across wide-FoV and panoramic inputs without the distortions of a fixed grid.

3.2.2 Feature Conditioned Gaussian Residuals

While Geometry Anchored Gaussians provide a camera-unified layout, UniSHARP further predicts Feature Conditioned Gaussian residuals to incorporate the semantic context and geometric priors necessary for high-fidelity synthesis. Unlike conventional monocular predictors that feed RGB images and depth images directly into a decoder, UniSHARP predicts residuals within a unified space by fusing 2D semantic image features with 3D ray-based geometric features.

For each geometry-anchored location, a Gaussian decoder predicts a residual tensor $\Delta \in \mathbb{R}^{B \times 14 \times L \times H_g \times W_g}$. Here, the 14 channels comprise a 3-channel center residual (two tangent-plane

offsets and one radial inverse-distance), 3 scale channels, 4 quaternion channels, 3 color channels, and 1 opacity channel. The final Gaussian is obtained by composing the anchor and residual:

$$\mathcal{G}_{p,\ell} = \mathrm{Compose}(\mathcal{B}_{p,\ell}, \Delta_{p,\ell}) = \{\boldsymbol{\mu}_{p,\ell}, \mathbf{s}_{p,\ell}, \mathbf{q}_{p,\ell}, \mathbf{c}_{p,\ell}, \alpha_{p,\ell}\}. \tag{3}$$

3.3 Training Strategy and Objective

Mixed-camera training. UniSHARP is trained under a mixed-camera regime, jointly optimizing perspective, wide-FoV, fisheye, and panoramic data within a single model. During training, a weighted sampler draws source-target image pairs from all supported datasets and groups each mini-batch by dataset for efficient collation and rendering. Although these data differ in image formation process, FoV, and valid target regions, UniSHARP does not introduce camera-specific branches. Instead, each sample is converted into the same ray-based training interface, and all camera types share a unified network architecture. As a result, UniSHARP learns a camera-unified Gaussian representation that transfers supervision across heterogeneous cameras while avoiding separate predictors for individual camera models.

Panoramic distortion adaptation. Equirectangular panoramas oversample polar regions because pixels near the poles correspond to narrower solid angles than those at the equator. To address this, we apply a latitude-dependent probabilistic dropout on the second Gaussian layer during training:

$$p_y = 1 - \frac{\max(\cos\theta_y, 0)}{\max_{y'} \max(\cos\theta_{y'}, 0)}, \tag{4}$$

where θ_y is the latitude of row y. The first layer is always preserved to maintain visible surface coverage. While the first layer is preserved for surface coverage, the second is then selectively suppressed via a Bernoulli mask $m_{p,2} \sim \mathrm{Bernoulli}(p_y)$ that biases opacity residuals. This approach shifts panoramic distortion adaptation from a specialized prediction branch to a training-time allocation strategy.

Objective. Let s and t denote source and target views. The training objective includes appearance supervision, depth supervision, and Gaussian regularization. We use $\hat{\mathbf{I}}_v$, $\hat{\mathbf{A}}_v$, and $\hat{\mathbf{D}}_v$ for rendered color, opacity, and distance at view v, and $\mathbf{I}_v$ and $\mathbf{D}_v$ for image and depth supervision. Appearance supervision encourages the accumulated opacity to cover valid pixels, and applies a perceptual term on target views where novel view artifacts are most visible.

$$\mathcal{L}_{\mathrm{app}} = \lambda_c \sum_{v\in\{s,t\}} \left\| \hat{\mathbf{I}}_v - \mathbf{I}_v \right\|_1 + \lambda_a \sum_{v\in\{s,t\}} \mathrm{BCE}(\hat{\mathbf{A}}_v, \mathbf{1}) + \lambda_p \Phi(\hat{\mathbf{I}}_t, \mathbf{I}_t), \tag{5}$$

where Φ is a perceptual loss computed from deep features and Gram statistics.

Depth supervision anchors both the source geometry used to initialize Gaussians and the target geometry produced after splatting [51].

$$\mathcal{L}_{\mathrm{dep}} = \lambda_d \left(\left\| \tilde{\mathbf{D}}_s^{-1} - \mathbf{D}_s^{-1} \right\|_1 + \left\| \hat{\mathbf{D}}_t^{-1} - \mathbf{D}_t^{-1} \right\|_1 \right), \tag{6}$$

where $\tilde{\mathbf{D}}_s$ is the first source radial layer and $\hat{\mathbf{D}}_t$ is the rendered target distance.

Gaussian regularization stabilizes degrees of freedom that are weakly constrained by photometric loss. It includes total variation on the second radial layer, floater suppression near abrupt first-layer inverse-distance changes, and multi-scale Sobel alignment in log-distance space [52, 53]:

$$\begin{aligned}\mathcal{L}_{\mathrm{geo}} = \lambda_{\mathrm{tv}} \left\langle \left\| \nabla \hat{\mathbf{D}}_{s,2}^{-1} \right\|_1 \right\rangle + \lambda_g \left\langle \hat{\mathbf{O}} \left(1 - \exp\left(- \left[\left\| \nabla \hat{\mathbf{D}}_{s,1}^{-1} \right\|_1 - \tau \right]_+ / \sigma \right) \right) \right\rangle \\ + \lambda_{\mathrm{gi}} \frac{1}{K} \sum_{v\in\{s,t\}} \sum_{k=1}^{K} \left\langle \left\| \nabla_{\mathrm{Sobel}} \mathcal{P}_k \left(\log \hat{\mathbf{D}}_v - \log \mathbf{D}_v \right) \right\|_2 \right\rangle,\end{aligned} \tag{7}$$

where $\hat{\mathbf{O}}$ is the predicted Gaussian opacity, K is the number of depth pyramid scales, $\mathcal{P}_k$ downsamples its input to the k-th scale, and $[x]_+ = \max(x, 0)$. The three terms respectively smooth the inverse distance of the second source layer, suppress opaque floaters around sharp first-layer distance discontinuities, and align rendered and supervised distance edges with multi-scale Sobel gradients.

Table 1: Composition of the proposed FoV stratified benchmark for universal-camera monocular novel view synthesis. Validation pairs are grouped by effective FoV and projection type, and sample counts denote evaluated source-target pairs.

Camera group	FoV range	Validation datasets	Validation samples
Perspective	60°–90°	DL3DV [54], RealEstate10K [55], Tanks and Temples [56], WildRGB-D [57]	36,873
Wide FoV	90°–140°	OmniRooms-Wide (ours), projected from OmniRooms	10,692
Fisheye	140°–180°	ScanNet++ Fisheye [58]	14,163
Panorama	360°	Replica [59], HM3D [60], OmniRooms (ours)	42,754

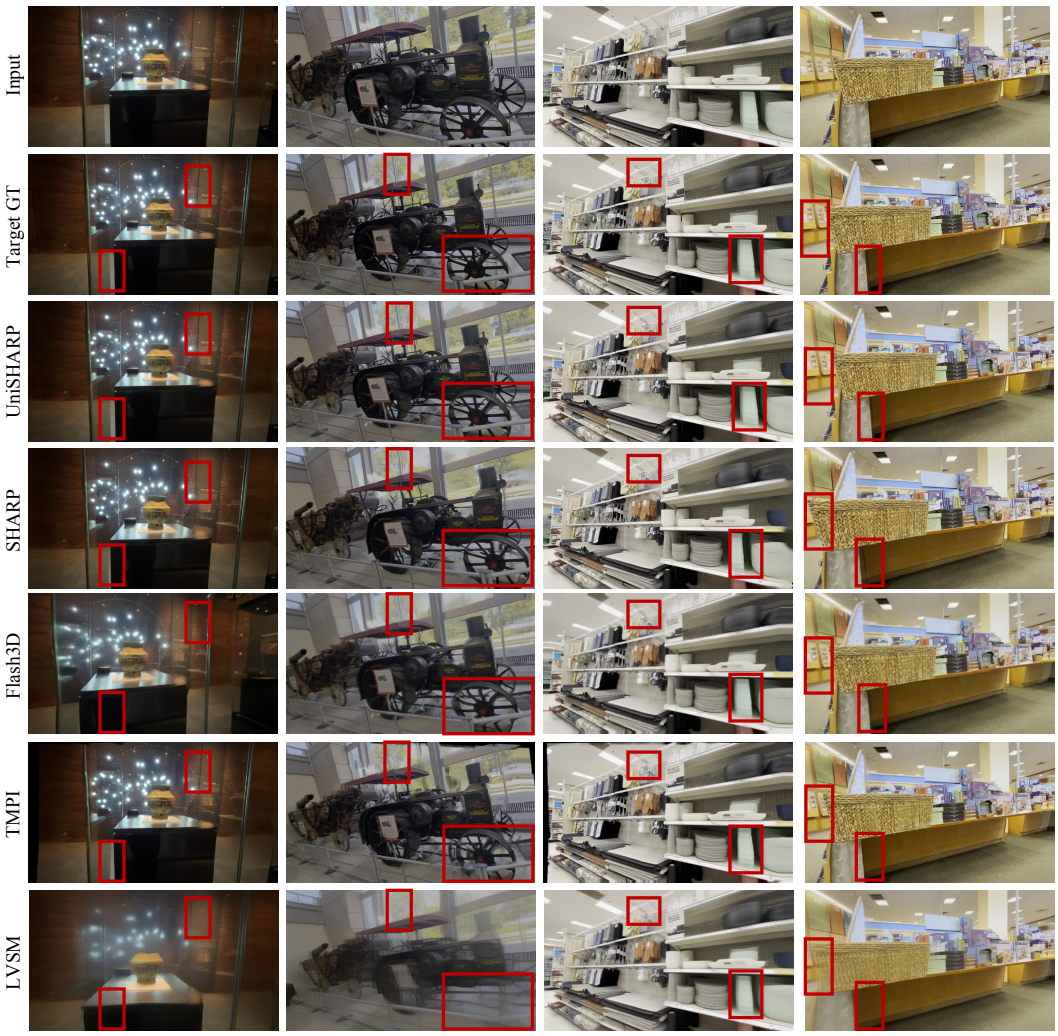

Figure 3: Qualitative comparison on perspective novel view synthesis. Given a single source image, UniSHARP produces sharper target-view geometry and fewer disocclusion artifacts than perspective monocular baselines, while preserving view-consistent scene structure.

For equirectangular panoramas, horizontal finite differences use circular boundary handling to respect the wrap-around topology. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{app}} + \mathcal{L}_{\text{dep}} + \mathcal{L}_{\text{geo}}. \tag{8}$$

Table 2: Quantitative comparison on perspective datasets included in the mixed training distribution. The best results are shown in red and the second-best results are in orange.

Method	WildRGB-D [57]			DL3DV [54]			RealEstate10K [55]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TMPI [39]	15.969	0.493	0.330	16.002	0.490	0.340	21.707	0.768	0.142
LVSM [41]	15.970	0.362	0.279	15.848	0.446	0.374	22.109	0.731	0.138
Flash3D [13]	17.378	0.553	0.395	18.418	0.574	0.179	23.573	0.785	0.172
SHARP [12]	18.300	0.568	0.212	17.368	0.545	0.286	24.215	0.789	0.101
UniSHARP	**21.556**	**0.674**	**0.143**	**19.468**	**0.594**	**0.196**	**24.495**	**0.795**	**0.087**

Table 3: Quantitative comparison on panoramic novel view synthesis. Results are reported on real and simulated ERP datasets using PSNR↑, SSIM↑, and LPIPS↓; best results are shown in red.

Method	HM3D [60]			OmniRooms			Replica [59]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PanoDreamer [62]	21.856	0.708	0.152	18.567	0.692	0.240	21.289	0.793	0.137
Matrix3D [63]	23.398	0.793	0.114	21.635	0.799	0.144	25.379	0.887	0.069
UniSHARP	**29.244**	**0.895**	**0.065**	**24.004**	**0.856**	**0.123**	**30.182**	**0.933**	**0.038**

4 Experiments

Training. We train one unified model on a mixture of perspective, wide-FoV, fisheye, and panoramic datasets. For perspective images, we use RealEstate10K [55], DL3DV [54], and WildRGB-D [57]. For wide-FoV images, we use OmniRooms-Wide, which is projected from our simulated indoor panoramic dataset. For fisheye images, we use ScanNet++ Fisheye [58]. For panoramic images, we use HM3D panorama datasets [60] and OmniRooms, our simulated indoor ERP dataset constructed with the AirSim platform [61] by rendering collision-free camera trajectories in synthetic indoor scenes. All datasets are converted into source-target training pairs under their native camera models. At each iteration, we first sample a dataset according to a fixed dataset-level distribution and then draw a batch from that dataset. For datasets without ground-truth depth, we use UniK3D [46] to generate pseudo depth labels. Additional implementation details are provided in Appendix A.1.

4.1 Benchmark & Metrics.

We evaluate UniSHARP on monocular NVS across perspective, wide-FoV, fisheye, and panoramic cameras. To address the limitations of evaluations tied to single projection families, we introduce a FoV stratified benchmark (Table 1). This benchmark provides a unified protocol to diagnose model behavior as camera geometry scales from narrow perspective to full 360° equirectangular inputs.

OmniRooms. OmniRooms is a simulated indoor ERP dataset collected via AirSim [**?**], with OmniRooms-Wide derived by projecting these panoramas into 130° equidistant fisheye views. For each anchor point on a 0.5m voxel grid, we render one central camera and 29 others randomly sampled within a local axis-aligned cube of edge length 30 cm around the source camera. Each frame is rendered as a 1024×2048 ERP image and all cameras share a fixed orientation.

Benchmark details. To align with single-source monocular NVS, we restrict target views to locally reachable positions: we require $> 60\%$ source-target overlap, a camera-center distance < 0.5m, and an image-index gap < 10. This design focuses evaluation on geometry inference under meaningful motion rather than unconstrained long-range hallucination. The protocol serves as a unified testbed for universal-camera NVS, enabling a direct diagnostic of how rendering quality scales across perspective, wide-FoV, fisheye, and 360-degree projections. We evaluate in a single-source, multi-target setting, using the first frame of each sequence as the source and the subsequent ten frames as targets. Results are reported using PSNR, SSIM, and LPIPS, averaged over all valid target views.

Baselines. For perspective novel view synthesis, we compare with representative single-image 3D Gaussian regressors, including SHARP [12] and Flash3D [13], as well as methods based on different scene representations, including LVSM [41], a large view synthesis model with minimal 3D inductive bias, and TMPI [39], a tiled multiplane-image representation for practical 3D photography. For wide-FoV, fisheye, and panoramic novel view synthesis, we compare with methods that cover



Figure 4: Qualitative comparison on panorama novel view synthesis. UniSHARP reconstructs coherent Gaussian geometry from a single panoramic input and renders sharper target views with fewer distortion-induced artifacts.

Table 4: Quantitative comparison on wide-FoV and fisheye novel view synthesis. Results are reported on OmniRooms-Wide and ScanNet++ Fisheye.

	ScanNet++ Fisheye [58]			OmniRooms-Wide		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PanoDreamer [62]	15.131	0.682	0.383	18.062	0.760	0.199
Matrix3D [63]	16.382	0.690	0.371	18.881	0.795	0.173
UniSHARP	**20.660**	**0.771**	**0.184**	**25.243**	**0.854**	**0.076**

different generation paradigms: PanoDreamer [62], an optimization-based single-image-to-360 scene method using diffusion and 3DGS, and Matrix3D [63], a diffusion-based video generation model.

4.2 Qualitative and Quantitative Evaluation

Perspective performance.

Table 2 reports the in-domain results on RealEstate10K, DL3DV, and WildRGB-D. UniSHARP achieves the best PSNR and SSIM across the perspective datasets and the best or second-best LPIPS, showing that universal-camera training preserves strong standard perspective rendering. The gains are consistent across real-

Table 5: Zero-shot perspective evaluation on Tanks and Temples [56].

Method	PSNR↑	SSIM↑	LPIPS↓
TMPI [39]	13.623	0.399	0.440
LVSM [41]	14.575	0.379	0.476
Flash3D [13]	15.985	**0.511**	0.312
SHARP [12]	15.964	0.502	0.301
UniSHARP	**16.315**	0.498	**0.282**

Table 6: Ablation study of model design components on WildRGB-D and HM3D. Each variant removes one component from the full model.

	WildRGB-D [57]			HM3D [60]		
Variant	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Full model	**21.56**	**0.674**	**0.143**	**29.24**	**0.895**	**0.065**
w/o native resolution allocation	21.21	0.664	0.155	28.72	0.872	0.071
w/o second Gaussian layer	20.63	0.642	0.168	28.29	0.877	0.083
w/o panoramic distortion adaptation	21.50	0.672	0.145	28.43	0.880	0.077
w/ depth-RGB images input	20.38	0.631	0.180	28.04	0.868	0.089

estate, object-centric, and in-the-wild scenes, indicating that the shared ray-distance Gaussian representation improves fidelity without overfitting to a single perspective dataset.

Table 5 evaluates out-of-domain generalization on the Tanks and Temples dataset. UniSHARP obtains the best PSNR and LPIPS compared with the strongest baselines. The SSIM remains close to Flash3D, suggesting that the unified representation preserves cross-dataset generalization while improving overall reconstruction fidelity.

Wide-FoV, fisheye, and panoramic performance.

On OmniRooms-Wide, UniSHARP consistently improves PSNR, SSIM, and LPIPS over both baselines. This shows that the ray-distance Gaussian representation remains effective for wide-FoV inputs, where the model must handle stronger projection distortion and larger angular coverage than standard perspective images. On ScanNet++ Fisheye, UniSHARP also outperforms PanoDreamer and Matrix3D by a clear margin, suggesting that the same geometry-aware parameterization transfers from projected wide-FoV views to native fisheye cameras. For panoramic novel view synthesis, UniSHARP achieves the best PSNR, SSIM, and LPIPS on HM3D, OmniRooms, and the out-of-domain Replica dataset. These results indicate that the proposed camera-unified design remains stable as the evaluation moves from wide-FoV and fisheye inputs to full 360° panoramas.

4.3 Ablation Studies

We conduct ablations on WildRGB-D and HM3D to analyze the main architectural and objective components of UniSHARP.

Model design. Table 6 summarizes the contribution of the main architectural components. Replacing the proposed 2D semantic and 3D geometric features with direct depth-RGB inputs causes the largest degradation on both datasets, showing that the learned feature space provides stronger context than direct RGB-depth conditioning. Removing the second Gaussian layer also hurts performance on both datasets, confirming the importance of an additional distance hypothesis for disocclusions and wide angular coverage. Native resolution allocation remains important for preserving input detail, while panoramic distortion adaptation has a clearer effect on HM3D, where regularizing equirectangular distortion is more critical than in perspective images.

5 Conclusion

We presented UniSHARP, a universal-camera feedforward 3DGS framework for monocular novel view synthesis. Starting from the observation that perspective-trained Gaussian regressors do not transfer reliably to heterogeneous camera systems, UniSHARP reformulates Gaussian prediction in a shared ray-distance space and composes Geometry Anchored Gaussians with Feature Conditioned Gaussian residuals. This design preserves the efficiency of single-image Gaussian regression while supporting perspective, wide-FoV, fisheye, and panoramic inputs within one prediction model. To evaluate this setting systematically, we further introduced a FoV stratified benchmark covering real and simulated scenes across narrow perspective to full panoramic cameras. Experiments on this benchmark show that UniSHARP maintains strong performance on perspective datasets and substantially improves novel view synthesis across diverse camera systems. We hope this work provides a practical foundation for monocular 3D Gaussian rendering in real-world imaging systems beyond the pinhole camera model.

Acknowledgements. This work was supported in part by the Shenzhen Key Industrial R&D Program under Grant No. ZDCY20250901103359008 and in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62676293.

References

[1] Xin Lin, Shi Luo, Xiaojun Shan, Xiaoyu Zhou, Chao Ren, Lu Qi, Ming-Hsuan Yang, and Nuno Vasconcelos. Hqgs: High-quality novel view synthesis with gaussian splatting in degraded scenes. In *ICLR*, 2025. 2

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

[3] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024. 3

[4] Meixi Song, Xin Lin, Dizhe Zhang, Haodong Li, Xiangtai Li, Bo Du, and Lu Qi. D^2GS: Depth-and-density guided gaussian splatting for stable and accurate sparse-view reconstruction. *arXiv preprint arXiv:2510.08566*, 2025.

[5] Hao Ren, Yiming Zeng, Zetong Bi, Zhaoliang Wan, Junlong Huang, and Hui Cheng. Prior does matter: Visual navigation via denoising diffusion bridge models. In *CVPR*, pages 12100–12110, 2025.

[6] Xiaoyuan Wang, Yizhou Zhao, Botao Ye, Xiaojun Shan, Weijie Lyu, Lu Qi, Kelvin Chan, Yinxiao Li, and Ming-Hsuan Yang. HoliGS: Holistic gaussian splatting for embodied view synthesis. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[7] Chunjiang Liu, Xiaoyuan Wang, Qingran Lin, Albert Xiao, Haoyu Chen, Shizheng Wen, Hao Zhang, Lu Qi, Ming-Hsuan Yang, Laszlo A Jeni, et al. Mosiv: Multi-object system identification from videos. *arXiv preprint arXiv:2603.06022*, 2026.

[8] Jingtong Yue, Zhiwei Lin, Xin Lin, Xiaoyu Zhou, Xiangtai Li, Lu Qi, Yongtao Wang, and Ming-Hsuan Yang. Roburcdet: Enhancing robustness of radar-camera fusion in bird's eye view for 3d object detection. *arXiv preprint arXiv:2502.13071*, 2025.

[9] Hao Ren, Zetong Bi, Yiming Zeng, Zhaoliang Wan, Lu Qi, and Hui Cheng. Strnet: Visual navigation with spatio-temporal representation through dynamic graph aggregation. In *CVPR*, pages 42464–42473, 2026. 2

[10] 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. 2

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 2

[12] Lars Mescheder, Wei Dong, Shiwei Li, Xuyang Bai, Marcel Santos, Peiyun Hu, Bruno Lecouat, Mingmin Zhen, Amael Delaunoy, Tian Fang, Yanghai Tsin, Stephan R. Richter, and Vladlen Koltun. Sharp monocular view synthesis in less than a second. In *ICLR*, 2026. 2, 3, 7, 8

[13] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 2, 3, 7, 8

[14] Longwei Li, Huajian Huang, Sai-Kit Yeung, and Hui Cheng. Omnigs: Fast radiance field reconstruction using omnidirectional gaussian splatting. *arXiv preprint arXiv:2404.03202*, 2024. 2, 3

[15] Zheng Chen, Chenming Wu, Zhelun Shen, Chen Zhao, Weicai Ye, Haocheng Feng, Errui Ding, and Song-Hai Zhang. Splatter-360: Generalizable 360 gaussian splatting for wide-baseline panoramic images. In *CVPR*, 2025. 3

[16] Cheng Zhang, Haofei Xu, Qianyi Wu, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Pansplat: 4k panorama synthesis with feed-forward gaussian splatting. In *CVPR*, 2025.

[17] Suyoung Lee, Jaeyoung Chung, Kihoon Kim, Jaeyoo Huh, Gunhee Lee, Minsoo Lee, and Kyoung Mu Lee. Omnisplat: Taming feed-forward 3d gaussian splatting for omnidirectional images with editable capabilities. In *CVPR*, 2025. 3

[18] Youming Deng, Wenqi Xian, Guandao Yang, Leonidas Guibas, Gordon Wetzstein, Steve Marschner, and Paul Debevec. Self-calibrating gaussian splatting for large field-of-view reconstruction. In *ICCV*, 2025. 2, 3

[19] Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation. *arXiv preprint arXiv:2512.16913*, 2025. 2

[20] Xiangkai Zhang, Dizhe Zhang, WenZhuo Cao, Zhaoliang Wan, Yingjie Niu, Lu Qi, Xu Yang, and Zhiyong Liu. Fly360: Omnidirectional obstacle avoidance within drone view. *arXiv preprint arXiv:2603.06573*, 2026.

[21] Haoran Feng, Dizhe Zhang, Xiangtai Li, Bo Du, and Lu Qi. Dit360: High-fidelity panoramic image generation via hybrid training. *arXiv preprint arXiv:2510.11712*, 2025.

[22] Yuheng Liu, Xin Lin, Xinke Li, Baihan Yang, Chen Wang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Hao Tan, Kai Zhang, Xiaohui Xie, Zifan Shi, and Yiwei Hu. Omniroam: World wandering via long-horizon panoramic video generation. *arXiv preprint arXiv:2603.30045*, 2026.

[23] Xin Lin, Xian Ge, Dizhe Zhang, Zhaoliang Wan, Xianshun Wang, Xiangtai Li, Wenjie Jiang, Bo Du, Dacheng Tao, Ming-Hsuan Yang, et al. One flight over the gap: A survey from perspective to panoramic vision. *arXiv preprint arXiv:2509.04444*, 2025.

[24] Changpeng Wang, Xin Lin, Junhan Liu, Yuheng Liu, Zhen Wang, Donglian Qi, Yunfeng Yan, and Xi Chen. PanoWorld: Towards spatial supersensing in 360° panorama world. *arXiv preprint arXiv:2605.13169*, 2026. 2

[25] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[26] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2

[27] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 3

[28] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 3

[29] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025. 3

[30] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025. 3

[31] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting for novel view synthesis. In *ICML*, 2025.

[32] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *ECCV*, 2024.

[33] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024. 3

[34] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

[35] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 3

[36] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 3

[37] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T. Freeman, David Salesin, Brian Curless, Noah Snavely, and Ce Liu. Slide: Single image 3d photography with soft layering and depth-aware inpainting. In *ICCV*, 2021.

[38] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. *TOG*, 41(4), 2022.

[39] Numair Khan, Eric Penner, Douglas Lanman, and Lei Xiao. Tiled multiplane images for practical 3d photography. In *ICCV*, 2023. 3, 7, 8

[40] Yuan Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 3

[41] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *ICLR*, 2025. 3, 7, 8

[42] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 3

[43] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric R. Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. In *CVPR*, 2024. 3

[44] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N. Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *CVPR*, 2025.

[45] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Mueller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025. 3

[46] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unik3d: Universal camera monocular 3d estimation. In *CVPR*, 2025. 3, 4, 7, 14

[47] Zimu Liao, Siyan Chen, Rong Fu, Yi Wang, Zhongling Su, Hao Luo, Li Ma, Linning Xu, Bo Dai, Hengjie Li, Zhilin Pei, and Xingcheng Zhang. Fisheye-gs: Lightweight and extensible gaussian splatting module for fisheye cameras. *arXiv preprint arXiv:2409.04751*, 2024. 3

[48] Huajian Huang, Yingshu Chen, Longwei Li, Hui Cheng, Tristan Braud, Yajie Zhao, and Sai-Kit Yeung. Sc-omnigs: Self-calibrating omnidirectional gaussian splatting. In *ICLR*, 2025. 3

[49] Zhengxian Yang, Fei Xie, Xutao Xue, Rui Zhang, Taicheng Huang, Yang Liu, Mengqi Ji, and Tao Yu. Directfisheye-gs: Enabling native fisheye input in gaussian splatting with cross-view joint optimization. *arXiv preprint arXiv:2604.00648*, 2026. 3

[50] Zheng Chen, Yan-Pei Cao, Yuan-Chen Guo, Chen Wang, Ying Shan, and Song-Hai Zhang. Panogrf: Generalizable spherical radiance fields for wide-baseline panoramas. In *NeurIPS*, 2023. 3

[51] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, pages 12578–12588, 2021. 5

[52] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 5

[53] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 44(3):1623–1637, 2022. 5

[54] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 6, 7

[55] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *TOG*, 37(4), 2018. 6, 7

[56] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *TOG*, 36(4), 2017. 6, 8

[57] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *CVPR*, 2024. 6, 7, 9, 14

[58] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Niessner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 6, 7, 8

[59] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6, 7

[60] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *NeurIPS Datasets and Benchmarks Track*, 2021. 6, 7, 9, 14

[61] Xian Ge, Yuling Pan, Yuhang Zhang, Xiang Li, Weijun Zhang, Dizhe Zhang, Zhaoliang Wan, Xin Lin, Xiangkai Zhang, Juntao Liang, Xiangtai Li, WenJie Jiang, Bo Du, Ming-Hsuan Yang, and Lu Qi. AirSim360: A panoramic simulation platform within drone view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26931–26940, June 2026. 7

[62] Avinash Paliwal, Xilong Zhou, Andrii Tsarov, and Nima Khademi Kalantari. Panodreamer: Optimization-based single image to 360 3d scene with diffusion. In *SIGGRAPH Asia Conference Papers*, 2025. 7, 8, 14

[63] Yuanxun Lu, Jingyang Zhang, Tian Fang, Jean-Daniel Nahmias, Yanghai Tsin, Long Quan, Xun Cao, Yao Yao, and Shiwei Li. Matrix3d: Large photogrammetry model all-in-one. In *CVPR*, 2025. 7, 8, 14

[64] Zixun Huang, Cho-Ying Wu, Yuliang Guo, Xinyu Huang, and Liu Ren. 3dgeer: 3d gaussian rendering made exact and efficient for generic cameras. In *ICLR*, 2026. 14

[65] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 14

[66] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 14

A Additional Experiments and Ablations

A.1 Implementation Details

All experiments are conducted on 8 H20 GPUs. UniSHARP uses the feature-only architecture described in Sec. 3, with a UniK3D ViT-L backbone initialized from pretrained weights [46]. For wide-FoV and fisheye rendering, we use the 3DGEER generic-camera Gaussian rasterizer [64]. We optimize the model with Adam [65] for 10^6 iterations using a 10^4-iteration warmup followed by cosine learning-rate decay. The learning rate decays from 1.0×10^{-5} to 1.0×10^{-6} for the depth head, and from 1.2×10^{-4} to 1.6×10^{-5} for the Gaussian decoder and prediction modules. The loss weights are $\lambda_c = 1.0$, $\lambda_a = 1.0$, $\lambda_p = 1.0$, $\lambda_d = 0.5$, $\lambda_{\text{tv}} = 1.0$, $\lambda_g = 0.5$, and $\lambda_{\text{gi}} = 0.5$.

Rendering. For perspective inputs, we use standard gsplat rasterization [66]. For equirectangular panoramas, we rasterize into cube-map faces and resample the rendered faces to equirectangular format. For wide-FoV and fisheye inputs, we use the 3DGEER generic-camera Gaussian renderer [64], which evaluates each Gaussian along per-pixel rays.

A.2 Training Objective Ablation

Table 7: Ablation study of the main training losses on WildRGB-D and HM3D. Each variant removes one loss term from the full objective to measure its contribution to rendering quality.

	WildRGB-D [57]			HM3D [60]		
Variant	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Full objective	**21.56**	**0.674**	**0.143**	**29.24**	**0.895**	**0.065**
w/o perceptual appearance loss	21.18	0.663	0.158	28.69	0.884	0.078
w/o target rendered depth loss	20.42	0.628	0.206	27.12	0.842	0.136
w/o second-layer TV regularization	21.30	0.667	0.151	28.56	0.879	0.084
w/o floater suppression	21.11	0.659	0.162	27.94	0.866	0.153

Training objective. Table 7 analyzes the main loss terms. The full objective achieves the best overall performance, reaching 21.56 PSNR and 0.143 LPIPS on WildRGB-D, and 29.24 PSNR and 0.065 LPIPS on HM3D. Removing target rendered depth supervision causes the largest PSNR drop among the loss ablations, reducing PSNR to 20.42 on WildRGB-D and 27.12 on HM3D, while also increasing LPIPS to 0.206 and 0.136. This shows that source-side depth supervision alone cannot constrain the Gaussian scene after view transformation; supervising the rendered target depth is critical for maintaining cross-view geometry and suppressing view-dependent distortions. The perceptual appearance loss improves visual fidelity, while second-layer TV regularization and floater suppression stabilize the Gaussian field. Floater suppression is particularly important for panoramic scenes, where removing it increases HM3D LPIPS from 0.065 to 0.153 due to unstable second-layer Gaussians near depth discontinuities.

A.3 Fisheye Dataset Visualization

A.4 Inference Time Comparison

Table 8: Inference time comparison for single-image novel view synthesis. Runtime is measured under the same evaluation setting; relative runtime is reported with respect to UniSHARP.

Method	Runtime ↓	Relative runtime ↓
UniSHARP	**3.1s**	–
PanoDreamer [62]	8.6s	2.8×
Matrix3D [63]	38.8s	12.5×

Inference efficiency. Table 8 compares the inference time of UniSHARP with panoramic baselines. UniSHARP completes inference in 3.1 seconds, while PanoDreamer and Matrix3D require 8.6 seconds and 38.8 seconds, respectively. PanoDreamer and Matrix3D are therefore 2.8× and 12.5×



Figure 5: Visualization of the fisheye validation data used in our benchmark. The samples illustrate the strong radial distortion and wide angular coverage that distinguish native fisheye novel view synthesis from standard perspective evaluation.

slower than UniSHARP. The speed advantage comes from the model design: UniSHARP predicts the complete Gaussian representation with a single feedforward pass and directly renders it, avoiding per-scene optimization, diffusion sampling, or iterative video generation at inference time.

B Limitations

UniSHARP is a feedforward Gaussian prediction model rather than a generative scene completion method. With mixed-camera training, the model can acquire a degree of extrapolation ability and can handle moderate disocclusions around the input view. However, when target views expose large regions that are completely outside the source image, the model has limited evidence for hallucinating unseen content. In such cases, holes or weakly supported structures may appear near outer boundary regions. Improving long-range extrapolation while preserving the efficiency and geometric consistency of feedforward Gaussian prediction remains an interesting direction for future work.

C Societal Impact

Universal-camera monocular novel view synthesis can broaden the use of 3D perception and rendering systems beyond standard perspective imagery. By supporting perspective, wide-FoV, fisheye, and panoramic inputs in a unified model, UniSHARP may benefit applications such as embodied AI, robotics, AR/VR content creation, immersive telepresence, and spatial documentation. The ability to infer renderable 3D structure from a single image can reduce capture requirements and make 3D content generation more accessible for devices with diverse camera systems. The proposed benchmark and dataset can also support more systematic evaluation of camera-general view synthesis methods, encouraging future research on robust and geometry-aware spatial intelligence.

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 main claims made in the abstract and introduction sections already reflect the paper's contributions.

   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 have been discussed in the Appendix B.

   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: [N/A]

Justification: This paper does not include theoretical results.

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 have tried to include all the details and referenced work for reproduction. We will also release the code of our method.

   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: [No]

Justification: The code will be published. For review, we submit an anonymized source-code zip with the supplementary materials. However, some datasets used in this work are not publicly redistributable, so we are unable to provide full open access to all data required for reproduction.

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: These details are described in the experiment section and appendix.

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: The reported PSNR, SSIM, and LPIPS metrics are deterministically computed on fixed benchmark test sets using a fixed evaluation protocol. Since the evaluation does not involve stochastic sampling or repeated trials, we do not report error bars or statistical significance measures.

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: This is described in the appendix section.

   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: This work conforms to the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: It is discussed in the Appendix C.

    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: This paper poses 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: The used assets are properly cited in the experiments section.

    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: This paper introduces a new indoor panoramic dataset collected by the authors. Detailed descriptions of the dataset are provided in the experiments section and appendix.

    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 paper does not involve crowdsourcing nor research with 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 paper does not involve crowdsourcing nor research with 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: The use of an LLM was limited strictly to language polishing, editing, and formatting assistance. It did not contribute to the conception of the study, the development of the methodology, the analysis or interpretation of results, or the generation of scientific conclusions. Therefore, the use of the LLM does not affect the methodological rigor, scientific validity, or originality of the research.

    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.