

```

1: for each  $i \in [1, N]$  do
2:    $\nabla c_i, \nabla \Sigma_i^{2D}, \nabla \mu_{p_i} = \text{Diff}(L)$ 
3:    $\nabla \Sigma_i = (W J_\theta)^\top (\nabla \Sigma_i^{2D}) (W J_\theta)$ 
    $\triangleright$  Gradient propagation to 3D
4:    $\nabla \mu_i \leftarrow \text{GradientPropagation}(\nabla \mu_{p_i}, W, K; \nabla c_i; \nabla J_\theta)$ 
    $\triangleright$  Gradient propagation to 3D & Chain rule
5:    $\nabla s_i, \nabla r_i \leftarrow \text{GradientPropagation}(\nabla \Sigma_i)$ 
    $\triangleright$  Chain rule to find derivatives w.r.t. scaling and
   rotation
6: end for
return  $\nabla \mu, \nabla c, \nabla s, \nabla r$ 

```

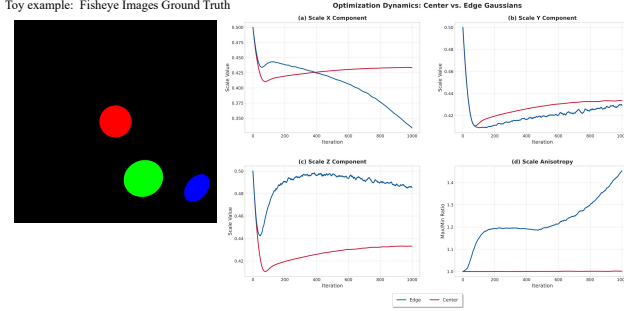


Figure 7. Toy experiment showing that strong fisheye distortion causes unstable and anisotropic Gaussian optimization near image boundaries.

Here, we provide an intuitive mathematical analysis explaining why regions with strong fisheye distortion tend to produce extremely anisotropic Gaussians during optimization. We analyze Gaussian shape optimization by examining the gradients of the covariance matrix as an example.

As shown in Eq. (7), the fisheye projection Jacobian \mathbf{J}_θ is a highly nonlinear function of the camera-space coordinates (x_c, y_c, z_c) , involving the distorted incident angle θ_d , its derivative θ'_d , and higher-order terms of the radial distance d and depth z_c . These factors jointly induce view-dependent scaling and anisotropy in the Jacobian, whose structure varies significantly with the incident angle.

During back-propagation (Algorithm 3), the 3D covariance gradient is given by $\nabla \Sigma_i = (\mathbf{W}\mathbf{J}_\theta)^\top (\nabla \Sigma_i^{2D}) (\mathbf{W}\mathbf{J}_\theta)$, indicating that the nonlinear Jacobian introduced by the fisheye model directly modulates the covariance update. In regions with large distortion (i.e., large incident angles), the Jacobian exhibits strong view-dependent anisotropy, leading to unbalanced gradients across views. In contrast, near the image center the Jacobian smoothly degenerates toward the pinhole case, resulting in more stable and nearly isotropic optimization.

We further illustrate this effect with a toy experiment (Fig. 7). Three spheres with identical radii are rendered into multiple fisheye images and fitted using three Gaussian primitives. The results show that the central Gaussian preserves isotropic scale during optimization, whereas the peripheral Gaussian (blue) is pulled by nonlinear, view-dependent gradients, leading to unstable optimization and extreme shape deformation.

These analysis helps explain the optimization instability observed near fisheye image boundaries.

D. More Quantitative and Qualitative Comparison Results & Analysis

We present test-view comparison results on the FisheyeNeRF dataset in Table 8 and additional qualitative comparisons on the Den-SOFT dataset in Fig. 9, together with further analysis of the performance differences among

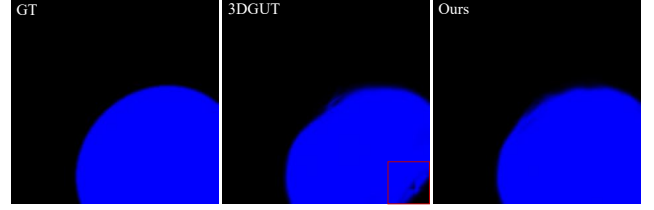


Figure 8. Toy experiment illustrating the behavior of our method and 3DGUT at fisheye image boundaries. We place a blue sphere in 3D space and render multi-view fisheye images as ground truth. Reconstruction results using 3DGUT and our method are shown above. Our method achieves more accurate fitting near the image boundaries and avoids mosaic-like artifacts, while 3DGUT exhibits degraded geometry under strong distortion.

Fisheye-GS, 3DGUT, SC-GS, and our method.

Fisheye-GS requires preprocessing to convert raw fisheye images into ideal equidistant fisheye projections as ground truth. For datasets with more severe distortions, this preprocessing leads to greater information loss (e.g., visible black borders and reduced effective resolution, stretching-induced loss of high-frequency details). As a result, although Fisheye-GS may obtain slightly higher numerical scores in some cases (e.g., Table 1), qualitative results (Fig. 5 and the supplementary video) better reflect the reconstruction quality, showing noticeable degradation in high-frequency details.

Among the remaining methods, 3DGUT is the closest to ours in performance. For a fair comparison, we follow the same six ScanNet++ scenes used in 3DGUT (Table 2). On two compact and highly reflective indoor scenes (IDs: 0a5c013435 and d415cc449b), 3DGUT slightly outperforms our method, which is consistent with its ray-based formulation that is more robust to localized lighting and specular effects and is not the primary focus of this work. In the other scenes and datasets, however, our method consistently achieves higher reconstruction quality.

Although both DirectFisheye-GS and 3DGUT adopt similar nonlinear projection models, their approaches target fundamentally different goals. 3DGUT augments 3DGS with ray tracing via the Unscented Transform (UT), estimating projected Gaussian’s mean and covariance using seven sigma points. This sampling-based formulation improves robustness in light-intensive or reflective scenes, but under strong fisheye distortion, particularly near image boundaries or on datasets with larger distortion such as Den-SOFT, this limited sampling may fail to capture local anisotropy. As a results, the estimated covariances may vary discontinuously across neighboring pixels, leading to blurred details, and mosaic-like artifacts (see Fig. 5, Fig. 9, and Fig. 8, supp.video 2:43, 2:58...).

In contrast, our analytic Jacobian formulation propagates Gaussian covariance through the Kannala–Brandt fisheye model in a continuous manner, enabling more stable co-

Table 8. Per-scene results of our method and related state-of-art works on the FisheyeNeRF [12] Dataset (Test View). ** Fisheye-GS still need to pre-process data before training.

FisheyeNeRF	SSIM↑	Globe PSNR↑	LPIPS↓	SSIM↑	Rock PSNR↑	LPIPS↓	SSIM↑	Cube PSNR↑	LPIPS↓	SSIM↑	Flowers PSNR↑	LPIPS↓	SSIM↑	Heart PSNR↑	LPIPS↓	SSIM↑	Chairs PSNR↑	LPIPS↓	SSIM↑	Average PSNR↑	LPIPS↓
Fisheye-GS**	0.7942	23.4159	0.2667	0.7539	24.4696	0.2575	0.7826	24.4171	0.2948	0.6843	21.5505	0.2901	0.7843	23.6402	0.3108	0.6540	18.6131	0.4262	0.7422	22.6844	0.3077
3DGUT	0.7790	24.0800	0.3420	0.7320	24.1400	0.3130	0.7630	24.3140	0.3690	0.6550	21.3070	0.3690	0.7730	23.6490	0.3930	0.8520	24.6950	0.3070	0.7590	23.6975	0.3488
Self-Cali-GS	0.7443	22.1169	0.4273	0.6799	22.2834	0.4777	0.7414	22.4269	0.4573	0.5868	19.5231	0.5212	0.7907	22.3154	0.3788	0.7710	20.9594	0.3535	0.7190	21.6042	0.4360
Ours	0.7937	24.0818	0.2423	0.7480	24.5222	0.2495	0.7755	24.4438	0.2758	0.6826	21.7037	0.2776	0.7798	23.5701	0.2899	0.8620	25.0001	0.2185	0.7736	23.8870	0.2589

Table 9. Cross-view joint optimization (CVO) under pinhole inputs of Tanks & Temples [18] Dataset.

Method	SSIM↑	drjohnson PSNR↑	LPIPS↓	SSIM↑	playroom PSNR↑	LPIPS↓	SSIM↑	train PSNR↑	LPIPS↓	SSIM↑	truck PSNR↑	LPIPS↓
3DGS	0.9012	29.4539	0.2448	0.9027	30.0726	0.2482	0.8093	22.3197	0.2171	0.8780	25.6492	0.1576
3DGS + random select	0.8975	29.4017	0.2434	0.9009	30.0162	0.2453	0.8182	22.3985	0.2031	0.8824	25.8986	0.1484
3DGS + CVO	0.9024	29.5530	0.2413	0.9083	30.3675	0.2466	0.8250	22.7337	0.2010	0.8872	25.9716	0.1466

Table 10. Performance comparison with varying batch sizes on Scene:Ruziniu in Den-SOFT dataset.

Batchsize	PSNR	LPIPS	Time (min)	Δ PSNR/ Δ Time
1	23.7717	0.2232	38.87	-
2	24.0154	0.2139	62.43	1.0344%
3	24.1583	0.2102	89.83	0.5215%
4	24.2262	0.2077	129.52	0.1711%
5	24.2896	0.2051	160.92	0.2019%

variance updates under large distortion. Combined with the CVO strategy, our method achieves sharper edges, smoother α -blending, and better cross-view geometric consistency, showing generality and effectiveness even in dataset like Den-SOFT (Table 3, Fig. 9), which exhibits more severe distortion, larger spatial coverage, and more challenging in-the-wild conditions (complex lighting, occlusions).

Finally, while numerical metrics between the two methods can be close in some scenes, qualitative results in our figures and videos show that our method preserves finer structures and sharper boundaries, further demonstrating the effectiveness of DirectFisheye-GS.

E. About Larger Batch-Size

In our experiments, the top-batchsize (defined in Line.332) is set to $N = 2$. Increasing N improves performance, but it also significantly raises the training time. For example, when N = the number of training images, the view selection strategy becomes irrelevant. We computed the trade-off between performance gain and training time for top-batchsizes $N = 1 \rightarrow 5$ over 30k iterations as shown in Table 10.

While increasing N results in better performance, the most significant improvement happens when moving from $N=1 \rightarrow 2$ —shifting from monocular to stereo supervision. This transition is conceptually important.

Table 11. Comparisons of computational costs for methods supporting native fisheye inputs. All methods are evaluated on a single NVIDIA A100 GPU.

Method	Ours	3DGUT	Self-Cali-GS
Training-Time (h)	0.395	0.404	3.333
Rendering-Speed (FPS)	77	55	31

F. Computational Cost

We compare our method with two approaches that support native fisheye input (others require preprocessing). Using the ScanNet++_8d563fc2cc scene as an example, our method achieves both the best reconstruction quality (refer to Table 2) and computational efficiency as shown in Table 11.

G. Limitations Discussion

A primary limitation is that CVO’s performance gain is not significant under complex lighting conditions. While it benefits optimization of view-dependent attributes (SH) of the same Gaussian, further modeling of complex effects like reflection and refraction is needed under the **Splat-paradigm**. More precisely model complex reflectance and anisotropy, advanced view-dependent modeling(e.g.,light reflection direction modeling) would be valuable future work.

Another promising work is that CVO uses multi-view feature correspondences (from COLMAP) to group overlapping views. It improves reconstruction in most areas and does not degrade performance in sparse-texture regions. However, truly super-challenging zones are already difficult for 3DGS, calling for improved representation and semantic constraints, which we leave for future work.

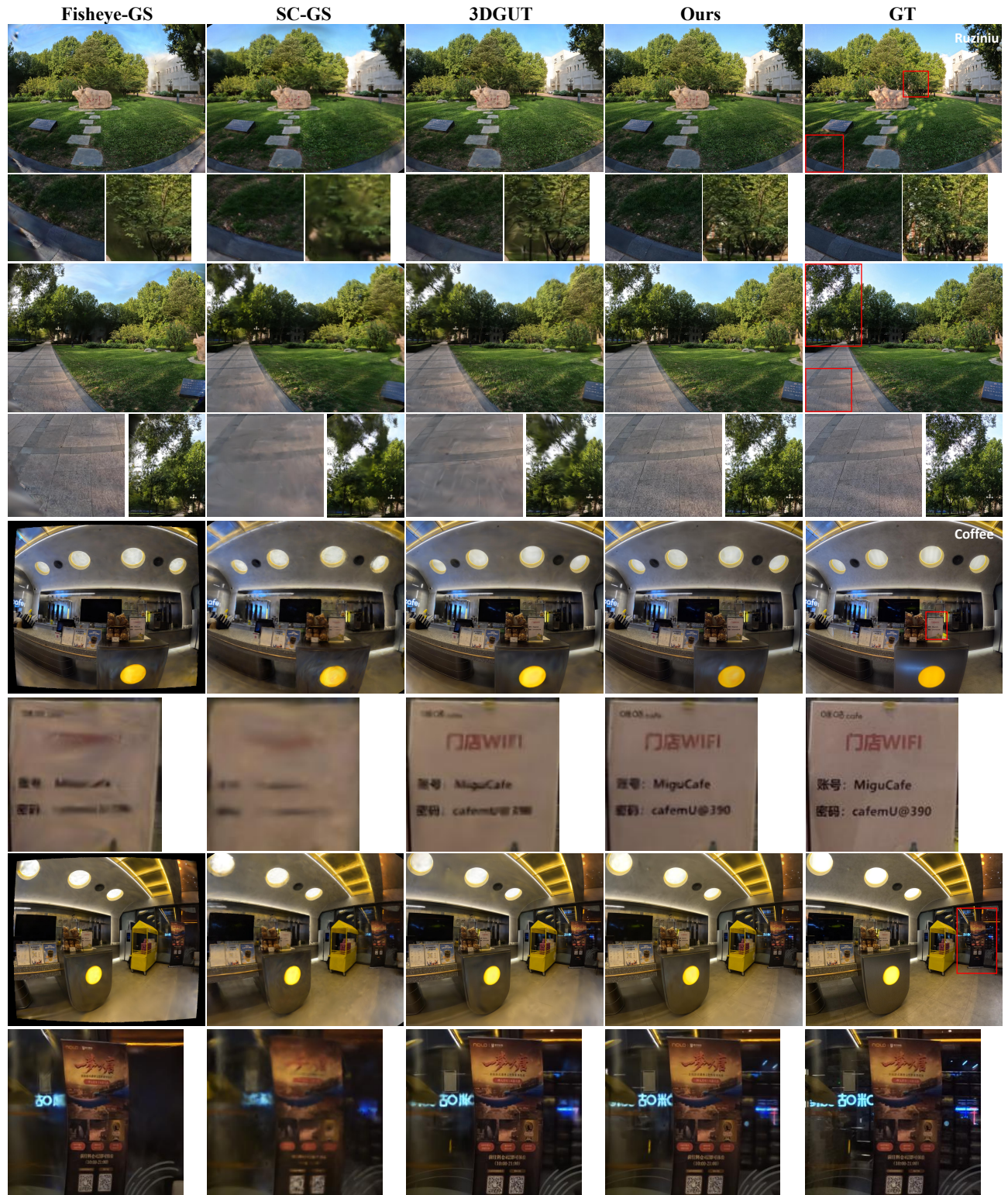


Figure 9. Qualitative comparison on Den-SOFT [47] dataset. Our method achieves the best results in both indoor and outdoor large-scale scenes.



(a) Fisheye Inputs



(b) Pinhole Inputs



(c) Different View Selection Strategy for Training

Figure 10. Qualitative comparison of ablation studies on cross-view joint optimization (CVO) with fisheye or pinhole camera inputs.