

# CoMapGS: Covisibility Map-based Gaussian Splatting for Sparse Novel View Synthesis

## Supplementary Material

### Supplementary Material Overview

This supplementary document is structured as follows:

- Section A: Validating Geometrical Correctness
- Section B: Supplementary Video Overview
- Section C: Addressing Concerns of the Proximity Loss
- Section D: Implementation and Runtime Efficiency
- Section E: Limitations

### A. Validating Geometrical Correctness

To validate the geometrical correctness of the 3D Gaussian splatting results generated by CoMapGS, we analyze the reconstructed 3D point cloud positions after training. Fig. 7 visualizes the completed set of Gaussian positions for the LLFF dataset trained with 3 views. To evaluate the proximity of the reconstructed geometry to the true scene, we applied a proximity classifier trained with 9 views, thus providing a broader reference set for assessing correctness.

#### A.1. Comparative Analysis of the Latest Methods

As shown in Fig. 7, the latest methods demonstrate different behaviors in sparse settings. Specifically:

- **FSGS [41]** exhibits a tendency to overfit to pseudo-ground truth during the training process. This leads to an uncontrolled proliferation of Gaussians in high-uncertainty regions, often resulting in geometrically incorrect structures. The method’s reliance on dense Gaussian generation without effective geometric alignment contributes to artifacts, especially in mono-view regions.
- **CoR-GS [38]** penalizes mono-view regions more strongly because the method is designed to retain Gaussians that are consistent across separate Gaussian models during training, ultimately prioritizing those that satisfy multiview geometry constraints. While this helps avoid overfitting in sparsely captured areas, it also leads to incomplete reconstructions or missing details in high-uncertainty regions. The lack of adaptive supervision across varying covisibility levels limits its ability to recover underrepresented structures.
- The proposed **CoMapGS** demonstrates significantly improved geometric alignment, as seen in the more coherent distribution of Gaussians across both high- and low-covisibility regions. By leveraging covisibility maps, CoMapGS applies adaptive supervision to ensure balanced updates for both multiview and mono-view regions. This is further supported by the proximity loss, which aligns Gaussians to the true scene structure without over-penalizing sparse areas.

#### A.2. Insights from Proximity Classification

In Fig. 7, the blue and red points denote the proximity classification results, with a threshold of 0.5. Blue points represent regions classified as closer to the true geometry, while red points indicate areas farther from the original structure. Key observations include:

- **Consistency in Multiview Regions:** In areas with high covisibility, CoMapGS effectively aligns Gaussians close to the ground truth, as evidenced by the predominance of blue points. This indicates that the adaptive weighting mechanism in multiview regions successfully guides reconstruction.
- **Improvement in Mono-View Regions:** Unlike FSGS [41] and CoR-GS [38], which either overfit or neglect mono-view regions, CoMapGS ensures that Gaussians in these high-uncertainty areas are plausibly positioned. The inclusion of proximity-based supervision mitigates structural divergence, which is critical for realistic novel view synthesis.
- **Balanced Distribution:** CoMapGS prevents the over-concentration of Gaussians in highly covisible regions, addressing the imbalance commonly observed in baseline methods. This balanced distribution directly translates to improved reconstruction quality, particularly in underrepresented areas.

#### A.3. Broader Implications

The ability of CoMapGS to produce geometrically correct Gaussian distributions highlights its robustness and adaptability across varying levels of sparsity. The integration of covisibility-aware supervision ensures that even regions with minimal training views are adequately represented, making the method suitable for complex real-world scenes as shown in Fig. 8.

#### A.4. Conclusion

The supplementary analysis presented here reinforces the findings in the main manuscript, demonstrating the effectiveness of CoMapGS in addressing the unique challenges of sparse view synthesis. By leveraging covisibility maps and proximity loss, the method achieves geometrically correct reconstructions, providing significant improvements over state-of-the-art 3DGS-based approaches.

### B. Supplementary Video Overview

In addition to the visualizations and analysis presented in this document, a supplementary demo video is provided.

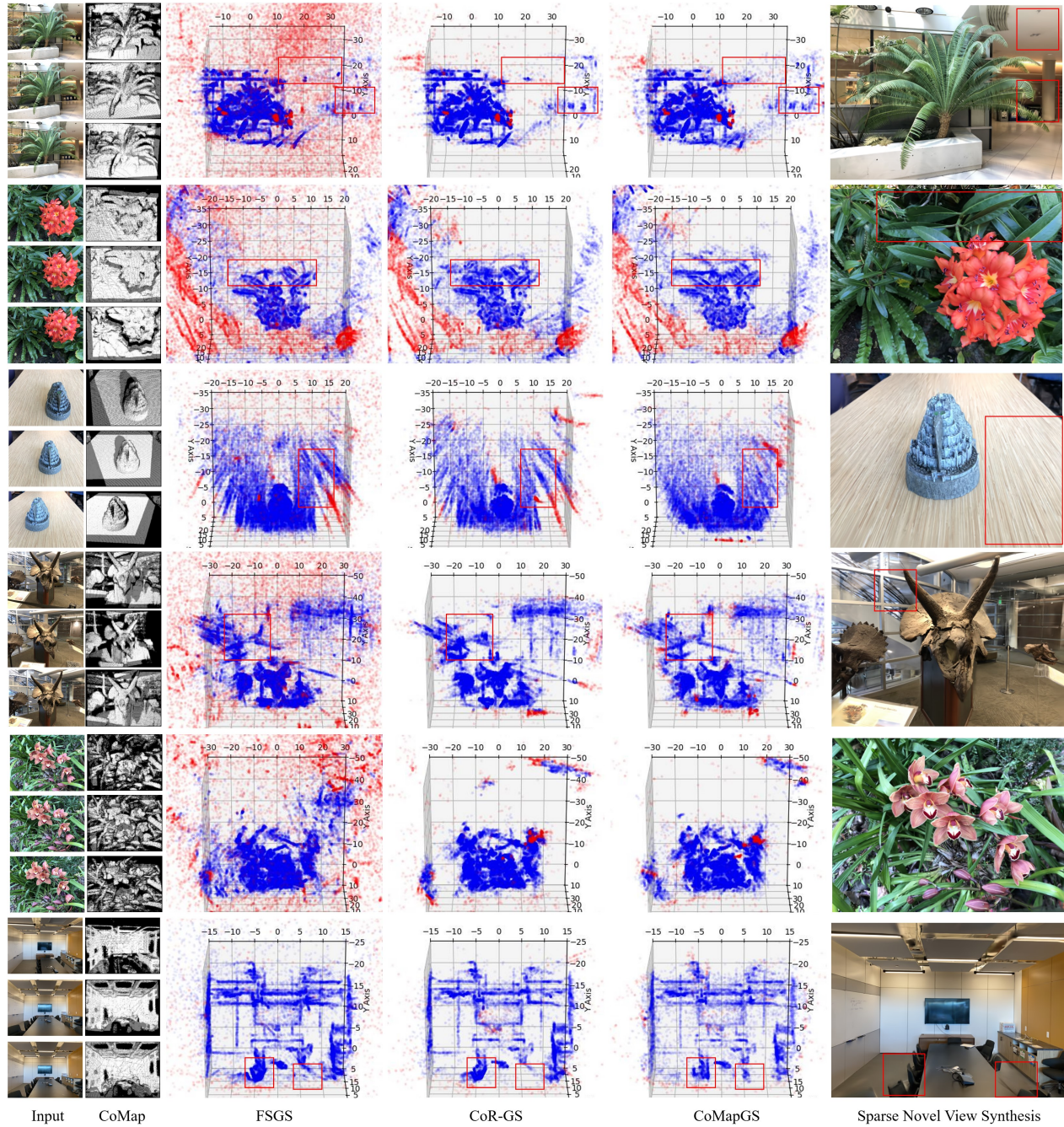


Figure 7. Visualization of Gaussian positions after training on the LLFF dataset. FSGS shows overfitting with uncontrolled Gaussian proliferation, while CoR-GS strongly penalizes low-covisibility regions. CoMapGS achieves geometrically aligned Gaussians across both high- and low-covisibility regions. Blue and red points indicate proximity classification results (threshold 0.5), with proximity classifiers trained on 9 views for evaluation.

The video includes:

1. Visualizations of covisibility maps generated for both LLFF and Mip-NeRF 360 datasets.
2. Enhanced initial point cloud updates guided by covisibility maps.
3. Explanation of the covisibility map-based supervision approach.
4. Qualitative comparisons with baseline methods, demonstrating the effectiveness of CoMapGS in recovering underrepresented regions.



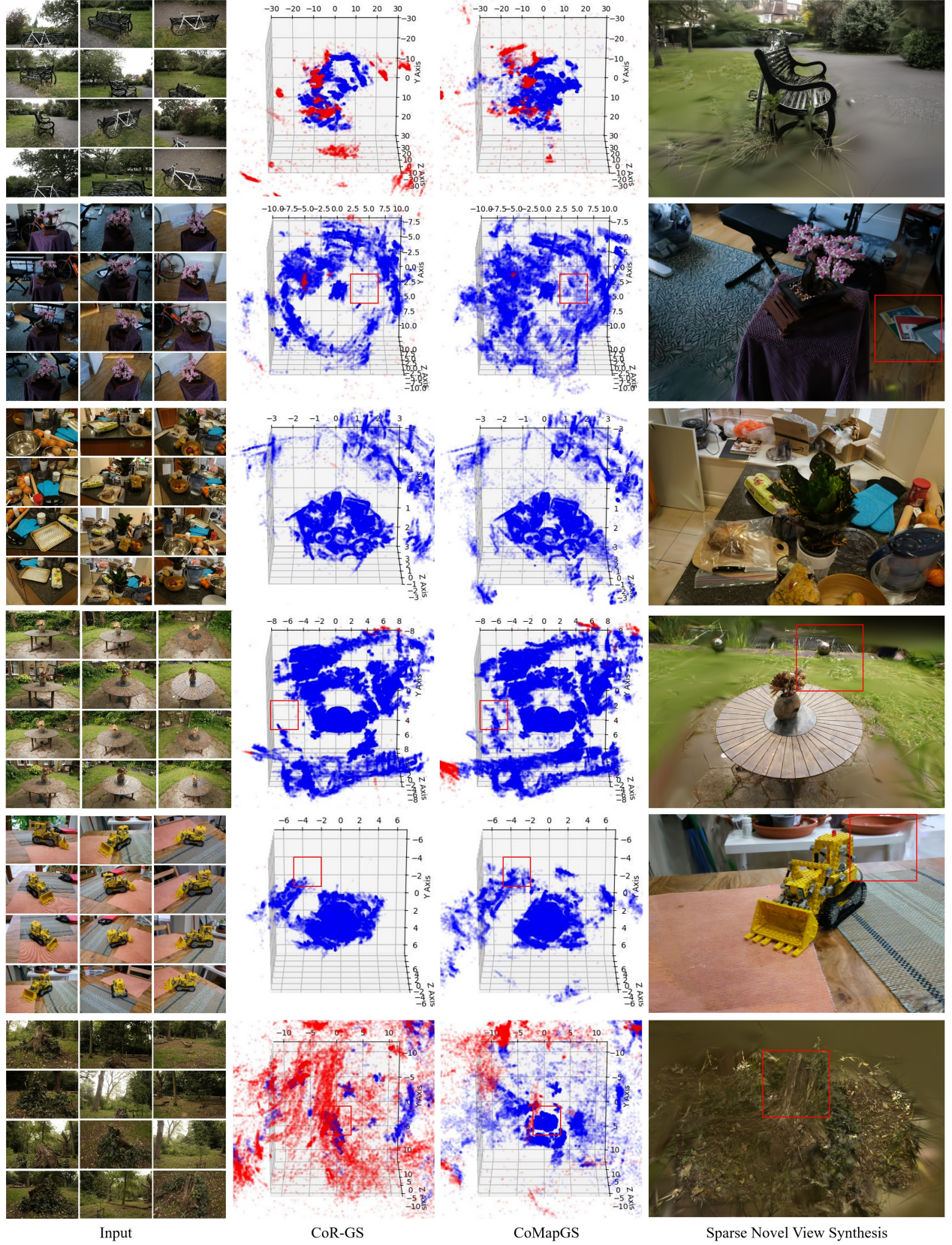


Figure 8. Visualization of Gaussian positions after training on the Mip-NeRF 360 dataset. CoR-GS [38] penalizes low-covisibility regions strongly, leading to incomplete reconstructions. CoMapGS achieves balanced Gaussian distributions with improved geometric alignment across both high- and low-covisibility regions. Blue and red points indicate proximity classification results (threshold 0.5), with classifiers trained on 24 views for evaluation.



## C. Addressing Concerns of the Proximity Loss

### C.1. Visual Results from Ablation Studies

Compared to Fig. 9(a), which uses sparse point clouds (PCLs), Fig. 9(b) shows that our enhanced PCLs correct the distorted geometry in the left box. Furthermore, Fig. 9(c), which incorporates proximity loss, extends supervision beyond the frustum—preserving the overall 3D structure and preventing errors caused by incomplete supervision, as illustrated in the right dotted box (missing chair). As discussed in Sec. 4.3, this highlights the complementary nature of proximity and photometric losses, leading to robust optimization across the entire scene.

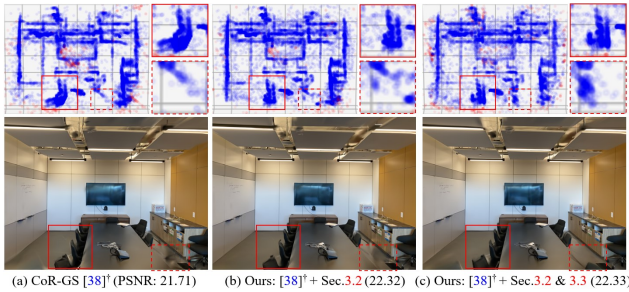


Figure 9. Ablation results with 3 training views, highlighting the effects of enhanced initial PCLs and proximity loss on novel view synthesis.

### C.2. Expected Behavior Outside the Frustum

The above ablation results also address concerns regarding the behavior of gaussians outside the frustum during training. Conventional 3DGS-based methods relying on photometric or depth losses adjust Gaussian parameters (3D coordinates, color, opacity, rotation, and scale) to improve rasterized color or depth images but do not explicitly guide 3D positions to converge. Proximity loss, computed independently of visibility via an MLP, penalizes wrongly located Gaussians while preserving geometrically correct points generated during initialization and densification, resulting in significantly improved geometric accuracy in the final reconstruction.

## D. Implementation and Runtime Efficiency

### D.1. Details of the Proximity Classifier

The proximity classifier  $f_p$  is a three-layer MLP represented as  $(\text{Input: } \mathbb{R}^3) \xrightarrow{\text{FC+ReLU}} \mathbb{R}^{128} \xrightarrow{\text{FC+ReLU}} \mathbb{R}^{128} \xrightarrow{\text{FC+Sigmoid}} \mathbb{R}^1$ , where each activation function follows its corresponding fully connected (FC) layer. The classifier is trained by minimizing binary cross-entropy loss using the Adam optimizer with a learning rate of 0.001 for 1,000 iterations per scene. Details on positive and negative data preparation are provided in Sec. 3.3.

## D.2. Time Complexity

Table 4 shows that the proposed CoMapGS introduces only a marginal overhead compared to CoR-GS [38], primarily due to the preprocessing step  $\alpha$  (+1.12 m) and the additional cost of running an MLP during training (+5.36 m).  $\alpha$  includes CoMap generation (11.51 s), initial PCL updates (multiview: 23.20 s, mono-view: 0.24 s), and proximity classifier training (data generation: 8.41 s, training: 28.06 s).

Compared to RegNeRF [21], a representative NeRF-based method which requires 69,768 training iterations, both CoR-GS [38] and our method require 10K iterations. While [38]+Full runs  $f_p$  at every iteration, the additional training cost remains minimal (+5.36 m), keeping it computationally efficient. Inference runs in real-time. Overall, CoMapGS achieves higher efficiency than RegNeRF while staying comparable to [38].

Table 4. Time complexities for the Fern scene (3-view) in the LLFF.

Process	RegNeRF [21]	[38]	[38]+PCLs	[38]+Full
Prep. (C)+ $\alpha$	-	COLMAP	C+35.35 s	C+72.22 s
Train	307.12 m	8.16 m	7.51 m	13.27 m
Render/Img	9,226.33 ms	3.16 ms	2.56 ms	3.30 ms

\* m = minutes, s = seconds, ms = milliseconds

## E. Limitations

### E.1. MAST3R and Depth Estimation Dependence

Our method relies on MAST3R and depth estimation, which can potentially propagate errors. To mitigate this, we register triangulated 3D points (using MAST3R’s dense correspondences) into COLMAP’s optimized camera poses with reprojection error validation ( $\leq 2$  pixels), as shown in Fig. 10. This prevents misaligned geometry across views and eliminates the need for manual confidence threshold tuning when using MAST3R’s predicted point clouds directly. While depth estimation errors may affect mono-view regions; however, their limited presence in the training set

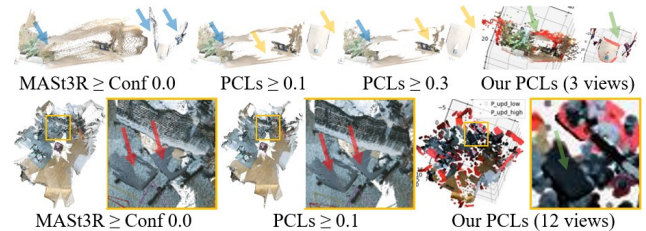


Figure 10. Illustration of how reprojection error validation robustly prevents  $\downarrow$  MAST3R errors (e.g., noisy PCLs  $\downarrow$ , missing details  $\downarrow$ , and misalignment  $\downarrow$ ) when registering 3D points into COLMAP camera poses.

minimizes the impact. Additionally, we adjust the proximity loss weight (Eq. 11) based on the scene-level covisibility score  $S$  to further reduce error propagation from depth predictions.

## E.2. Limitations on DTU Dataset

The DTU dataset focuses on masked regions captured in an artificial, black, and structureless environment, where both depth prediction and COLMAP camera pose estimation often fail due to a lack of contextual information. While CoMapGS performs well in generic, real-world scenarios, it is less effective in controlled settings—or in scenes where the background is predominantly sky and thus geometrically projected to infinity. In such cases, objects are centrally positioned under controlled lighting, and the scene geometry is highly artificial, posing challenges for CoMapGS. Consequently, high-certainty mono-view regions may contain invalid geometry, increasing the likelihood of depth estimation errors. This, in turn, hinders the enhancement of initial point clouds and reduces the effectiveness of the proximity loss.

Nevertheless, our tests on a subset of scenes from the DTU dataset indicate that performance does not degrade and remains comparable to the baseline CoR-GS. However, we were unable to run COLMAP successfully on the full DTU dataset and thus could not reproduce the results reported by recent methods. Preliminary experiments further suggest that synthetic or highly constrained environments may present similar limitations.

## E.3. Comparisons with NeRF-based Methods

While our paper does not include an extensive set of qualitative comparisons with the latest NeRF-based methods, Fig. 11 presents a representative example. It highlights qualitative differences between our method and recent NeRF-based approaches, illustrating contrasts in rendering quality and geometric fidelity under sparse-view conditions.



Figure 11. Visual comparisons with the latest NeRF-based methods.