Cross-Resolution Gaussian Splatting (XR-GS) for Hybrid Resolution Photography

Shreyans Sethi Vaishnavi Khindkar

16-825: Learning for 3D Vision. Final Project Report

1 Abstract

Recent advances in 3D Gaussian Splatting (3D-GS) have demonstrated high-quality, real-time novel view synthesis under the assumption that all training images share a uniform resolution. However, this assumption rarely holds in practical photography pipelines, where mixed-resolution inputs naturally arise due to zoom, compression, sensor differences, and capture distance. Low-resolution images introduce noisy high-frequency gradients, leading to hallucinated geometry, over-sharpened artifacts, and excessive Gaussian growth, while high-resolution images contain semantically meaningful fine details that are often underutilized.

In this work, we introduce Cross-Resolution Gaussian Splatting (XR-GS), a resolution-aware training framework that enables 3D-GS to effectively exploit hybrid-resolution supervision. XR-GS integrates six modifications: resolution-aware loss scaling, gradient scaling, adaptive densification and pruning thresholds, low-resolution curriculum training, a cross-resolution consistency loss, and a resolution-aware rendering ablation. We evaluate XR-GS on a modified LLFF benchmark with synthetic mixed-resolution inputs and on two real-world captured scenes with true resolution diversity. Our results demonstrate that XR-GS significantly reduces Gaussian count, memory usage, and hallucinated artifacts while maintaining comparable or improved perceptual quality across cluttered, high-frequency scenes.

2 Introduction and Motivation

3D Gaussian Splatting (3D-GS) has emerged as a highly efficient alternative to Neural Radiance Fields (NeRFs), enabling real-time novel view synthesis with photorealistic quality. By representing a scene as a set of anisotropic 3D Gaussians optimized through differentiable rasterization, 3D-GS achieves an excellent balance between fidelity and computational cost. However, existing 3D-GS frameworks rely on an important implicit assumption: that all training views are captured at uniform spatial resolution.

This assumption is fundamentally violated in real-world image capture scenarios. In consumer photography, some views may be captured with optical zoom, while others rely on digital zoom and sensor upsampling. Mobile phone cameras routinely apply aggressive compression, while DSLR cameras provide high-frequency texture and edge information. Similarly, moving closer to a subject implicitly increases spatial resolution relative to distant views. As a result, real datasets inevitably contain a mixture of high-resolution (HR) and low-resolution (LR) supervision.

When forced to optimize jointly over mixed-resolution images without any resolution-awareness, 3D-GS exhibits several failure modes. First, LR images introduce spurious high-frequency gradients that are not grounded in true scene structure, leading to hallucinated geometry and visually unstable Gaussians. Second, shared-scale optimization causes over-sharpened edges and inconsistent surface thickness. Third, the densification mechanism aggressively increases Gaussian count to explain LR noise, resulting in unnecessary memory consumption and slower rendering. Finally, the fine-grained structural information present in HR images is not fully exploited due to conflicting supervision from LR views.

The goal of this project is to enable 3D-GS to explicitly utilize mixed-resolution supervision through resolution-aware training. We introduce Cross-Resolution Gaussian Splatting (XR-GS), a framework that treats HR and LR images differently across the optimization, densification, and supervision pipelines. By doing so, XR-GS preserves reliable geometry from LR views while allowing HR views to inject true fine-scale details. Our approach requires no architectural changes to the core 3D-GS representation and is compatible with existing Gaussian rasterization pipelines.

3 Background and Related Work

The original 3D Gaussian Splatting work [2] and later variants such as LightGaussian (which improved compression) implictly assume uniform, high-resolution input views and treat all images equally. As a result, when some views are low-resolution, the model overfits to LR noise and fails to recover fine structure. More recent systems like SuperGS [6] and S2Gaussian [4] address low-resolution inputs by employing two-stage pipelines that build a coarse Gaussian model and then tend to synthesize high-frequency detail through super-resolution priors or pseudo-views. While these produce visually sharp outputs, they do not incorporate true high-resolution evidence and do not explicitly handle heterogeneous-resolution inputs or reason about per-camera resolution which is the gap that XR-GS is designed to fill.

NeRF-based super-resolution methods (e.g., RefSR-NeRF [1], NeRF-SR [5]) demonstrate that combining a few HR "anchor" images with many LR support views can successfully inject real high-frequency detail. Yet these methods inherit the computational burden of NeRF: long training times, expensive rendering, and implicit representations that limit scalability. XR-GS aims to bring this high-resolution reference principle into the efficient Gaussian splatting domain and create resolution-aware optimization without hallucinating detail. By weighting losses and gradients based on view resolution, modifying densify/prune rules, and enforcing cross-resolution consistency, XR-GS leverages real HR evidence while safely incorporating LR images, thus addressing a gap not covered by prior Gaussian or NeRF-based approaches.

4 Dataset

To evaluate XR-GS under realistic mixed-resolution settings, we construct both a synthetic hybrid-resolution benchmark using LLFF and real-world datasets captured under controlled resolution diversity.

4.1 Synthetic Mixed-Resolution LLFF Dataset

We adopt seven scenes from the LLFF dataset [3] as our primary benchmark. Each scene is preprocessed using COLMAP to obtain calibrated camera poses. To simulate hybrid resolution, we designate 30% of the images as high-resolution (HR) while the remaining 70% are downsampled by a factor of 5 to create low-resolution (LR) views. This process ensures controlled resolution diversity while maintaining consistent camera geometry. The HR/LR flags are propagated into the dataset loader to enable resolution-aware supervision during optimization. One additional LLFF scene (Leaves) was excluded because downsampling caused COLMAP pose recovery to fail.

4.2 Real-World Mixed-Resolution Captures

We capture two real-world scenes exhibiting true resolution diversity. The first scene, Dining Room, simulates resolution variation through changes in capture distance: close-up views act as HR supervision while distant views function as LR supervision. The second scene, Camera Keychain, is captured using two physically distinct cameras with different megapixel counts, yielding authentic sensor-level resolution differences. These datasets validate that XR-GS extends beyond synthetic downsampling and generalizes to real hybrid-resolution imaging pipelines.

5 Methodology

We build XR-GS as a set of training-time modifications on top of the open-source 3D Gaussian Splatting (Gsplat repository [7]) without altering the underlying scene representation or renderer architecture.

Our goal in XR-GS is to make 3D Gaussian Splatting (Gsplat repository) explicitly aware of mixed-resolution training supervision. Standard 3D-GS treats all input views identically, which leads to conflicting gradient signals when high-resolution (HR) and low-resolution (LR) images have fundamentally different frequency content. XR-GS modifies the optimization pipeline in six complementary ways: (1) loss weighting, (2) gradient scaling, (3) resolution-aware densification and pruning, (4) an LR-only curriculum warm-up, (5) cross-resolution consistency loss, and (6) resolution-aware rendering (ablation).

Figure 1 shows the overall training pipeline.

5.1 Overview of XR-GS Pipeline

Given a set of multi-view calibrated images with heterogeneous spatial resolutions, we preprocess them to assign each image a binary HR/LR tag. During each training step, XR-GS performs the following resolution-aware sequence:

- identify whether the current view is HR or LR,
- adjust its loss contribution accordingly,
- scale gradients flowing into Gaussian scale parameters,
- apply HR-specific or LR-specific densification/pruning thresholds, and

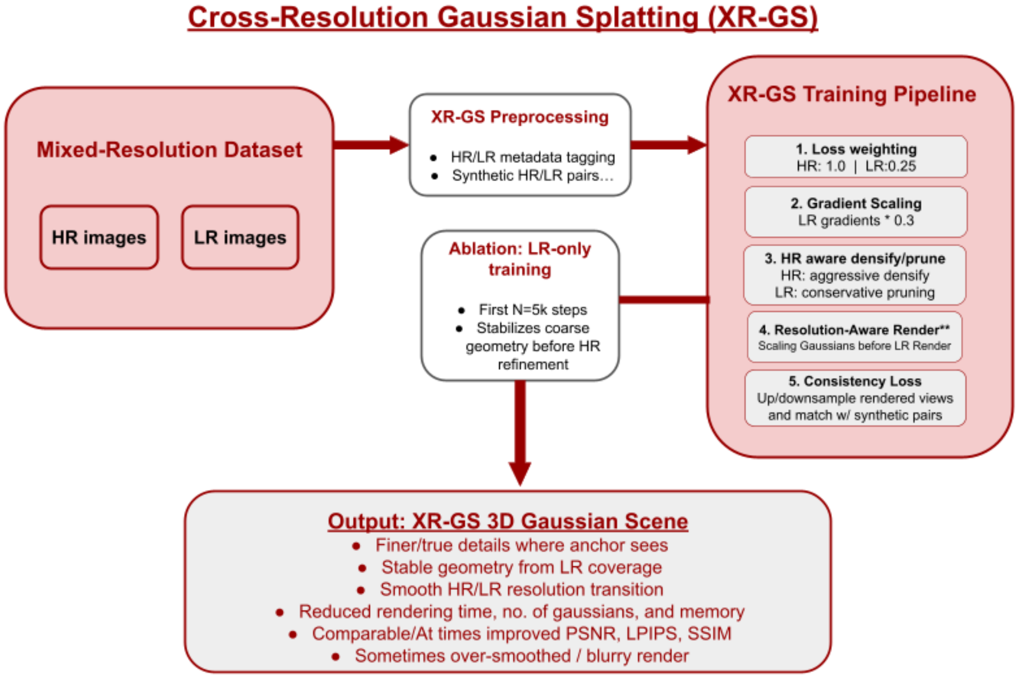

Figure 1: Overview of the XR-GS training pipeline. Mixed-resolution images enter the pipeline with HR/LR flags. XR-GS applies resolution-aware loss weighting, gradient scaling, densification and pruning, curriculum LR-only training, and cross-resolution consistency loss. The final 3D Gaussian scene incorporates both coarse structure from LR views and fine details from HR views.

- apply consistency constraints using synthetic resolution pairs.

These components are complementary: loss and gradient scaling reduce conflicting supervision from LR views, densify/prune adjusts the structural evolution of the Gaussian field, and consistency loss enforces multi-scale agreement. Overall, the aim is to ensure LR images provide large-scale geometric structure while HR images encode surface detail. Note that while the Resolution-Aware Render and LR-only training were added to the XR-GS system, ablations showed their limitations and failure cases which is why they are disabled by default in the XR-GS system.

5.2 Resolution-Aware Loss Weighting

Standard 3D-GS optimizes a photometric loss:

$$\mathcal{L}_{\text{base}} = (1-\lambda)\|\mathbf{C}-\mathbf{C}^*\|_1 \,+\, \lambda \cdot (1-\text{SSIM}(\mathbf{C},\mathbf{C}^*))\,,$$

where $\mathbf{C}$ and $\mathbf{C}^*$ denote the rendered and ground-truth pixel colors.

But this assumes all images carry equally reliable supervision.

Low-resolution images often contain:

- aliasing artifacts,
- blurred edges,

- upsampling-induced sharpening, and
- quantization noise.

Allowing such images to strongly influence optimization harms geometry and texture. Therefore, XR-GS attenuates their loss contributions. We found empirically that weighting LR losses by 0.25 provides stable training, but any reduction factor that significantly reduces LR influence yields similar trends.

$$\mathcal{L} = \begin{cases} 1.0 \cdot \mathcal{L}_{\text{base}}, & \text{if HR view} \\ 0.25 \cdot \mathcal{L}_{\text{base}}, & \text{if LR view} \end{cases}$$

This simple modification dramatically reduces hallucinated high-frequency structure.

5.3 Gradient Scaling for Gaussian Radii

Even with loss weighting, LR images can induce large gradients in Gaussian *scale* parameters. These gradients act on the anisotropic covariance matrices controlling splat size. Because LR images cannot reliably constrain fine structure, their gradients tend to incorrectly sharpen or expand splats.

XR-GS suppresses this by multiplying LR gradients by a factor $\alpha = 0.3$ (Chosen through grid search):

$$\nabla_{\text{LR}}\sigma \leftarrow 0.3 \cdot \nabla_{\text{LR}}\sigma$$

This targets only the geometry-sensitive parameters (scale components), keeping color and opacity gradients unchanged. The effect is smoother surfaces, fewer spurious sharp edges, and reduced instability across LR views.

5.4 Resolution-Aware Densification and Pruning

In standard 3D-GS, Gaussians are:

- densified (split) when they receive large gradients,
- pruned when they contribute little to rendered images.

However, LR views naturally produce noisy gradients, causing unwanted densification and large Gaussian counts.

XR-GS modifies this by maintaining **two separate sets of thresholds**:

$$\text{threshold}_{\text{HR}} < \text{threshold}_{\text{LR}}$$

This means:

- **HR views** $\rightarrow$ easier to trigger densification (trusted detail)
- **LR views** $\rightarrow$ harder to densify, easier to prune (avoid noise)

This significantly reduces:

- Gaussian count (by 25–70%),

- memory footprint,
- runtime per frame.

This reduction is particularly important for real-time rendering budgets and mobile deployment scenarios.

It also enforces that fine structure comes from HR views, not noisy LR inputs.

5.5 LR-Only Warm-Up (Curriculum Stage)

We introduce a stage where only LR views supervise the optimization for the first N iterations (typically 5000):

$$\mathcal{D}_{\text{train}}(t < N) = \{\text{LR views only}\}$$

The goal is to allow the model to establish coarse global geometry using many LR views before HR supervision introduces fine-detail refinement. This should theoretically make the optimization more stable when HR images are introduced. However, we include the curriculum stage for completeness as its impact was negligible compared to other XR-GS components, and it is disabled by default in our final system.

5.6 Cross-Resolution Consistency Loss

To explicitly enforce geometric consistency across resolution scales, we introduce a synthetic cross-resolution supervision mechanism. This forms the **core conceptual contribution** of XR-GS.

For each training image I, we generate a paired synthetic image $\bar{I}$ at the opposite resolution:

- High-resolution (HR) images are downsampled to create a low-resolution (LR) counterpart.
- Low-resolution (LR) images are upsampled to create a pseudo high-resolution (HR) counterpart.

These paired synthetic images are precomputed once per training image and reused throughout optimization. During training, the model renders an image I_{rend} from the Gaussian scene, and we project it to the opposite-resolution domain of its input view. For a high-resolution (HR) input, I_{rend} is downsampled; for a low-resolution (LR) input, it is upsampled. We denote this resolution-projected render as

$$I_{\text{proj}} = \phi(I_{\text{rend}}),$$

where $\phi(\cdot)$ is bicubic upsampling or downsampling to match the synthetic paired target $\bar{I}$.

The cross-resolution consistency loss is defined as:

$$\mathcal{L}_{\text{cons}} = w\left(\lambda_1 \,\|I_{\text{proj}} - \bar{I}\|_1 \;+\; \lambda_2 \left(1 - \text{SSIM}(I_{\text{proj}}, \bar{I})\right)\right),$$

where w controls the overall strength of the consistency term, and λ_1, λ_2 balance the L1 and SSIM components.

Intuition. High-resolution views introduce reliable fine details, while low-resolution views contain blur and aliasing that should *not* be interpreted as genuine structure. The consistency loss enforces that any detail the model learns must remain coherent when viewed at different resolutions, while spurious LR-specific noise is discouraged from turning into sharp geometry when supervised by HR images. This stabilizes geometry across scales, reduces resolution-dependent hallucinations, and prevents Gaussian fragmentation in novel views.

5.7 Resolution-Aware Rendering (Ablation Only)

We experimented with scaling Gaussian radii by the camera resolution:

$$\sigma' = k(R) \cdot \sigma$$

In this idea, the Gaussian radii are explicitly scaled at render time depending on the input view resolution. The intuition is to physically match Gaussian footprint to the spatial resolution of each view. However, this approach introduces excessive blur and global gradient smoothing, leading to degraded PSNR and unstable geometry. As a result, this component is not included in the final XR-GS configuration but is reported for completeness. Failure cases of this change are explained in the ablation studies below.

6 Ablation Studies / Analysis of XR-GS Changes

In this section, we present ablation studies where we isolate each XR-GS modification and compare its scene rendering to the baseline. This highlights the visual impact of each change, along with its advantages and failure modes. Note, the ablation for the LR-Only Warm-Up is not shown here since the results were not significantly different from the baseline.

6.1 Gradient Scaling and Resolution-Aware Loss Weighting

In this ablation study, we evaluate XR-GS using only its loss-weighting and gradient-scaling components, with Figure 2 showing a view that was provided as a low-resolution input. The baseline Gsplat reconstruction of the fern tree appears oversharpened and noticeably noisy (particularly along the fine leaves of the plant) suggesting that the model is overfitting to pixel-level noise specific to the LR view. This happens because the baseline allocates additional Gaussians to explain artifacts that are not true scene structure, causing the resulting render to look less realistic. In contrast, XR-GS downweights both the photometric loss and associated gradients coming from LR views, preventing these noisy signals from dominating the optimization. As a result, the model relies more heavily on HR images to determine where genuine detail should be represented, leading to a reconstruction that is smoother, more consistent with the true geometry, and visually more believable. Although some level of artificial noise is still visible in the XRGS render (See the right side of the fern, especially with light coming through the leaves), it is still able to preserve true fine structure. This indicates that even this minimal change, without densify/prune modifications or consistency loss, already provides some robustness to LR noise.

6.2 Resolution-Aware Densification and Pruning

Figure 3 compares the LLFF Orchids scene between baseline Gsplat (middle) and an XR-GS ablation using only resolution-aware densify/prune rules. In this high-resolution view, the baseline

Figure 2: Comparison of Ground Truth, Baseline Render, and XR-GS Render for Loss Weighting + Gradient Scaling Ablation.

reconstruction shows unstable geometry with jittery artifacts, especially on the top-right petal and lower leaves, because the default densification strategy creates many small Gaussians in response to high-frequency signals, which can fail to stabilize. XR-GS applies easier densify / prune thresholds for HR views (and harder for LR), reducing unnecessary Gaussian growth and eliminating weak, unstable primitives. This produces a cleaner, more stable reconstruction, though this can come at the cost of mild over-smoothness, resulting in the rendering having a "painted" appearance. Unlike loss or gradient scaling, which only adjust supervision strength, the densify/prune modification changes the structure of the Gaussian field itself, preventing long-term accumulation of unstable Gaussians that degrade geometry.

Figure 3: Comparison of Ground Truth, Baseline Render, and XR-GS Render for Densification/Pruning Ablation.

6.3 Resolution-Aware Rendering

Here, we conduct an ablation of the resolution-aware rendering idea where, for low-resolution views, we temporarily inflated the Gaussian scales before rendering to mimic optical blur, computed the loss against the LR image, and then restored the original scales. The intuition was that LR images should supervise a blurred version of the shared Gaussian field rather than forcing splats to match pixel-level noise. However, even after grid-searching suitable scale factors, this approach proved ineffective: as shown in Figure 4, increasing splat sizes caused gradients to diffuse broadly, erasing structure in regions like the floor and planter while producing overly smooth, smeared geometry. Because this method perturbs the Gaussian field in an unnatural, inconsistent way during training, the model becomes unstable. As a result, we discarded this approach, and it is not part of the final XR-GS system.

Figure 4: Comparison of Ground Truth, Baseline Render, and XR-GS Render for Resolution Aware Render Ablation.

6.4 Cross-Resolution Consistency Loss

Figure 5 shows the effect of adding only the Consistency Loss to the XR-GS pipeline. In the dining-room example, the visualized input image is low-resolution; thus, any noise or aliasing present in the LR input becomes amplified when upsampled for comparison with HR predictions in the consistency loss. This means spurious high-frequency patterns that appear in the baseline reconstruction are penalized more strongly, leading XR-GS to suppress unsupported artifacts and produce a cleaner, more coherent reconstruction. The key principle is that true geometric structure should remain stable across resolutions, but noise should not. However, the novel-view flower render (Figure 6) illustrates a limitation of this approach: because consistency loss rewards agreement across scales, the model sometimes gravitates toward larger, smoother Gaussians that remain consistent under both upsampling and downsampling. This reduces high-frequency noise but can also wash out fine details, yielding a more "blended" appearance that may be overly smooth compared to the baseline. Overall, consistency loss improves stability and reduces hallucinations, but at the risk of oversmoothing when the HR signal is complex or highly detailed.

7 Results and Discussion

7.1 Quantitative Results

To evaluate the effectiveness of the XR-GS modifications, we ran experiments on the modified LLFF dataset where, for each of the seven scenes, we trained both the baseline Gsplat implementation and our XR-GS variant for 30,000 iterations, with the baseline using default hyperparameters.

Table 1 reports the percentage change from baseline to XR-GS across key metrics. XR-GS substantially reduces resource usage, with an average memory reduction of 46%, training time reduction of 44%, and a 70% decrease in the number of Gaussians. These improvements suggest that

Figure 5: Comparison of Ground Truth, Baseline Render, and XR-GS Render for Consistency Loss Ablation.

Figure 6: Comparison of Baseline and XR-GS for Novel View Rendering in Consistency Loss Ablation.

Metric	Avg. % Change from Base Gsplat to XR-GS
Training Time	−44.57 ↑
Num Gaussians	−69.68 ↑
Memory Used	−46.50 ↑
LPIPS	+23.15 ↓
SSIM	+0.45 ↑
PSNR	−0.80 ↓

Table 1: Average percentage change across 7 modified LLFF scenes. Arrows indicate whether the direction of change represents an improvement (↑) or degradation (↓).

XR-GS produces a more compact scene representation. The resolution-aware loss weighting, gradient scaling, and adjusted densify/prune thresholds reduce the influence of low-resolution images and thus, prevent unnecessary Gaussian growth which results in this faster and lighter optimization.

In terms of reconstruction quality, we see a mixed pattern. SSIM improves slightly (+0.45%) and PSNR decreases marginally (–0.8%), indicating that XR-GS maintains roughly comparable reconstruction fidelity which is encouraging considering the large efficiency gains. However, LPIPS increases (+23%), suggesting reduced perceptual similarity. This may stem from (1) XR-GS intentionally down-weighting LR details, which can also create smoother, less textured reconstructions, and (2) LPIPS's sensitivity to certain smoothing artifacts, as it relies on deep network features that do not always correlate perfectly with human perception.

7.2 Strengths and Weaknesses

Metric	Dining Room (% Change)	Keychain (% Change)
Training Time	−24.87 ↑	−16.32 ↑
Num Gaussians	−70.99 ↑	−66.05 ↑
Memory Used	−33.08 ↑	−2.59 ↑
LPIPS	−8.83 ↑	+33.23 ↓
SSIM	+3.60 ↑	−1.56 ↓
PSNR	+3.92 ↑	−15.81 ↓

Table 2: Percentage change from baseline Gsplat to XR-GS for two collected scenes. Arrows indicate whether the direction of change represents an improvement (↑) or degradation (↓).

During these experiments, a consistent trend was also observed: XR-GS outperformed on cluttered, highly textured LLFF scenes (e.g., Fern, Flower, Orchids), while baseline Gsplat tends to perform better on simple, low-texture scenes or ones with a singular object (e.g., Fortress, Room). We hypothesize that cluttered scenes provide stronger structural cues in HR images, allowing XR-GS's resolution-aware mechanisms to suppress LR noise without harming detail. On the other hand, in scenes with minimal texture, XR-GS's smoothing tendencies can blur the limited high-frequency details available and the consistency loss does not punish visual differences as strongly at different levels since the lack of texture allows for differences to "get away with it".

To validate this idea, we compared baseline and XR-GS on two self-captured scenes with intentionally contrasting characteristics: a dining-room scene rich in clutter and fine texture, and a keychain scene with a uniformly colored background and a single object. These two setups allowed us to directly test how scene complexity influences the relative performance of XR-GS versus the baseline. As can be seen in Table 2, XR-GS outperforms the baseline GSplat on all the metrics for the Dining Room, whereas it only shows efficiency improvements on the Keychain scene, underperforming in all the visual reconstruction metrics. We visualize side-by-side comparisons of the ground truth image (high-res input), the Gsplat reconstruction and the XR-GS reconstruction in Figure 7 and Figure 8.

The results are consistent with our hypothesis: in the Dining Room scene, the baseline Gsplat hallucinates noise and forms unstable blobs because LR views introduce artificial high-frequency artifacts that pollute the Gaussian field. XR-GS mitigates this by allowing HR views to dominate optimization and reinforcing true structure, while suppressing LR artifacts through its loss/gradient scaling, HR-biased densify/prune rules, and consistency loss.

In contrast, for the Keychain scene, XR-GS tends to oversmooth, which is obvious in its blurry looking output. Both LR and HR input views will look nearly identical and contain minimal fine structure. Because XR-GS suppresses LR gradients and consistency loss provides only weak guidance in such scenes, the model receives little incentive to preserve sharp edges. As a result, XR-GS converges toward smooth, blurry Gaussians that satisfy the loss terms but lose the sharper boundaries that baseline Gsplat maintains.

Overall, XR-GS is designed to trust HR supervision over LR signals, which is beneficial in textured scenes but can be harmful when the scene lacks distinctive high-frequency detail. This explains the degradation in LPIPS, SSIM, and PSNR observed on simple scenes. How XR-GS could be improved for simple scenes:

1. **Adaptive Weighting:** Estimate image texture (e.g., via Fourier or gradient magnitude) and increase LR weighting when the scene is globally low-frequency, preventing oversmoothing.

2. **Region-Aware Scaling:** Apply spatially varying loss/gradient scaling so smooth regions retain LR supervision while textured regions benefit from XR-GS attenuation.

3. **Hybrid Training Schedule:** Instead of only downweighting LR early on, reintroduce LR influence in late training stages to recover subtle structure that HR views alone cannot reinforce.

8 Conclusion

We introduced XR-GS, a resolution-aware extension of 3D Gaussian Splatting designed to handle the mixed-resolution setting common in real image capture. By incorporating loss and gradient scaling, resolution-dependent densify/prune rules, and a cross-resolution consistency loss, XR-GS reduces Gaussian count, memory usage, and hallucinated artifacts while preserving strong visual

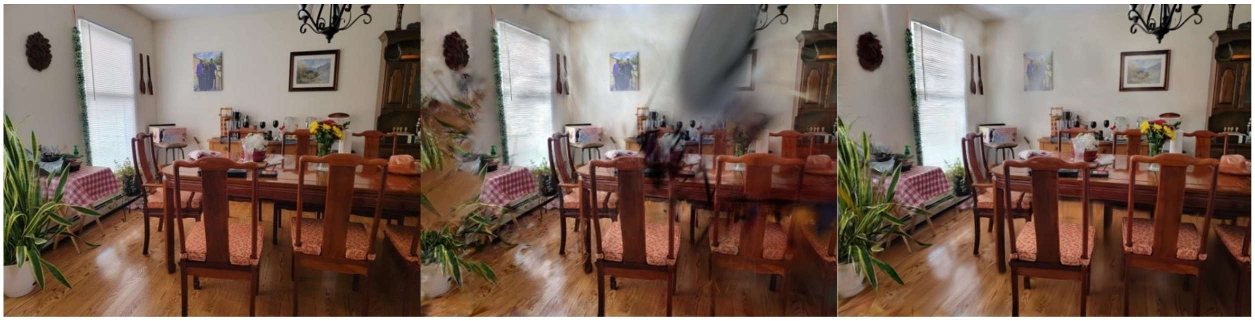

Figure 7: Ground Truth, Baseline, and XR-GS for Cluttered Dining Scene.

Figure 8: Ground Truth, Baseline, and XR-GS for Single Object Keychain Scene.

quality on cluttered, high-frequency scenes. However, XR-GS can oversmooth in low-texture settings where LR and HR images provide similarly weak structural cues. Future work could be focused on addressing this limitation and replacing the binary HR/LR separation with a continuous, texture-aware weighting scheme to make the system more adaptive and robust across a wider range of scene characteristics.

References

[1] Xudong Huang, Wei Li, Jie Hu, Hanting Chen, and Yunhe Wang. Refsr-nerf: Towards high-fidelity and super-resolution view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8244–8253, 2023.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139:1–139:14, 2023.

[3] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):29:1–29:14, 2019.

[4] Yecong Wan, Mingwen Shao, Yuanshuo Cheng, and Wangmeng Zuo. S2gaussian: Sparse-view super-resolution 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 711–721, 2025.

[5] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High-quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, pages 6445–6454, 2022.

[6] Shiyun Xie, Zhiru Wang, Yinghao Zhu, and Chengwei Pan. Supergs: Super-resolution 3d gaussian splatting via latent feature field and gradient-guided splitting. *arXiv preprint arXiv:2410.02571*, 2024.

[7] 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.