Diffusion-Guided Gaussian Splatting for Autonomous Driving

Patrick Wang
Stanford University
patwang@stanford.edu

Marcelo García
Stanford University
mrgaresc@stanford.edu

Maxton Huff
Stanford University
maxton@stanford.edu

Abstract

3D Gaussian Splatting (3DGS) has emerged as a powerful technique for real-time novel-view synthesis, offering high visual fidelity and speed. However, in autonomous driving scenarios, where camera viewpoints are limited and scenes are often noisy or partially observed, 3DGS suffers from artifacts such as floating Gaussians, color bleeding, and geometric distortions. We present Diffusion-Guided Gaussian Splatting (Di3DGS), a hybrid framework that integrates a diffusion-based denoising model into the 3DGS training loop. By generating synthetic pseudo-views from novel angles, refining them through a diffusion network, and reprojecting the cleaned images back into the optimization pipeline, Di3DGS proactively reduces artifacts and improves reconstruction quality in underconstrained regions. Using the nuScenes dataset, we evaluated our approach to see if it achieves better geometric consistency and visual realism in sparse-view settings.

1. Introduction

Real-time novel-view synthesis is a crucial component in perception systems for autonomous vehicles, enabling downstream tasks such as object tracking, occlusion reasoning, and scene understanding. Among recent approaches, 3D Gaussian Splatting (3DGS) has emerged as a compelling solution, offering photorealistic rendering with low latency and competitive fidelity. Unlike volumetric methods or mesh-based rendering, 3DGS directly fits a sparse set of anisotropic Gaussians to match reprojected views, making it especially attractive for time-sensitive applications.

However, this approach breaks down in practical autonomous driving scenarios, where camera viewpoints are limited by physical constraints and the environment often contains motion blur, dynamic objects, or sensor noise. These conditions lead to severe artifacts during 3DGS optimization, including floating or ”fuzzy” Gaussians, color bleeding, and geometric distortions, especially in regions not well-constrained by input views. Such errors not only degrade visual quality but also jeopardize the reliability of downstream tasks such as tracking and localization.

To address this, we propose Diffusion-Guided Gaussian Splatting (Di3DGS), a hybrid pipeline that augments 3DGS with a learned diffusion-based denoiser. Our method inserts a diffusion cleanup module into the training loop by first rendering synthetic pseudo-views from novel, underconstrained viewpoints. These views are then passed through a diffusion network trained to suppress artifacts and enhance structural detail. The cleaned views are reprojected into 3DGS as corrected supervision signals, guiding the optimizer to converge on more accurate and photorealistic scene representations. Crucially, our closed-loop formulation retains differentiability through the renderer while keeping the diffusion model frozen, preserving training stability.

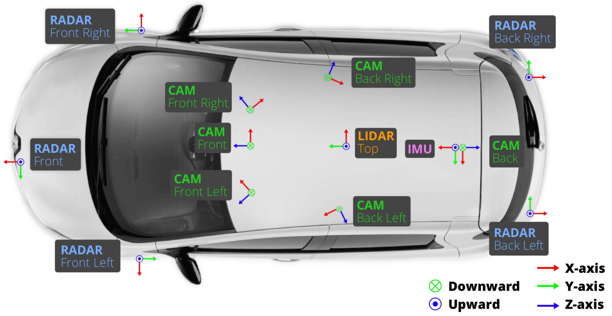

Figure 1. Camera Setup on Data Collection Cars

We evaluate Di3DGS on a subset of the nuScenes dataset, using five of the six available cameras for training and reserving the sixth (front-facing) view for validation as demonstrated in Figures 1. Initial results using the baseline 3DGS model show over-smoothed textures, geometry warping, and low contrast in reconstructed scenes. These artifacts are notably reduced when integrating our diffusion-guided refinement, suggesting improved generalization and geometric consistency. This work provides a potential approach toward robust real-time 3D perception under real-world sensor limitations.

2. Related Work

Real-time 3D scene reconstruction has been a central challenge in autonomous driving, where fast and accurate environment modeling is crucial for safe navigation. Various techniques have emerged to address this problem, with an increasing focus on novel-view synthesis, artifact correction, and leveraging limited sensor inputs.

3D Gaussian Splatting (3DGS), introduced by Kerbl et al. [1], is a seminal method that enables high-quality novel-view synthesis by fitting anisotropic Gaussians to a sparse set of 3D observations. The primary advantage of 3DGS is its ability to model a scene with high speed and low computational overhead, making it suitable for real-time applications. However, as with many real-time methods, the accuracy of 3DGS diminishes in sparse-view scenarios, leading to artifacts such as floating Gaussians, color bleeding, and geometric distortions, particularly in regions with insufficient observation.

Building on this foundation, DrivingForward [2] proposed a feed-forward pipeline that leverages only RGB input for real-time scene reconstruction. This approach removes the dependency on LiDAR, which is typically expensive and resource-intensive, while still delivering comparable reconstruction quality. However, it faces similar challenges to the original 3DGS method in terms of handling sparse viewpoints and sensor noise, particularly in dynamic urban environments.

To address dynamic objects, DrivingGaussian [4] extends 3DGS by incorporating LiDAR priors and dynamic Gaussian graphs. This approach allows the model to better handle moving objects and occlusions in driving scenarios. While this improves the model's robustness in dynamic settings, the approach still relies heavily on accurate LiDAR data, which can be noisy and sparse in real-world conditions.

In the domain of artifact correction, DIFIX3D+ [3] introduces a single-step diffusion model that applies denoising to 3D reconstructions. While this method successfully reduces rendering artifacts, it is limited by its one-step denoising process, which may not effectively address the complex, multi-step optimizations required for real-time autonomous driving scenarios.

In summary, while previous works have made significant strides in improving speed, fidelity, and artifact correction for 3D scene reconstruction, our Di3DGS framework introduces a novel approach by incorporating diffusion-based artifact removal directly into the optimization loop, offering a solution for real-time novel-view synthesis in autonomous driving environments.

3. Data

Our project uses a curated subset of the nuScenes dataset, a large-scale autonomous driving dataset collected in urban environments. nuScenes includes multi-modal sensor data such as RGB images, LiDAR, radar, and GPS/IMU. For this project, we focus solely on the RGB camera data captured from six fixed surround-view cameras mounted on the vehicle (front, rear, left, right, front-left, and front-right), which together provide a full 360-degree view of the scene.

To ensure stable 3D reconstruction and minimize the impact of dynamic objects, we select scenes based on the density of moving entities (e.g., vehicles, pedestrians). Using the detection annotations provided with the dataset, we rank scenes by dynamic object count and select those with the least activity. This filtering step allows us to target scenarios with minimal motion, reducing reconstruction artifacts caused by object movement between frames. By doing this, we can choose the least dynamic scenes for our training and evaluation without other noises.

4. Methods

Our approach, Diffusion-Guided Gaussian Splatting (Di3DGS), builds upon the strengths of 3DGS by introducing a closed-loop, diffusion-guided refinement step. We integrate a trained diffusion model, stable-diffusion-v1-5, directly fit it into the 3DGS optimization pipeline to correct artifacts proactively during training. By generating and refining pseudo-views from intermediate angles, Di3DGS tackles the challenge of sparse-view reconstruction in driving environments, reducing artifacts such as color bleeding and geometry warping. The model architecture is illustrated in Figures 2.

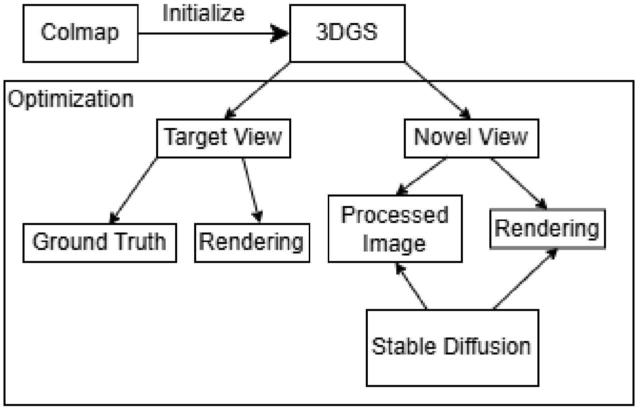

Figure 2. Di3DGS Architecture

4.1. Novel View Generation

Novel View Generation. Let $\mathcal{C}ref$ denote the set of reference cameras used to train the 3D Gaussian Splatting (3DGS) model, and let $\mathcal{C}\mathrm{tgt}$ denote the held-out target cameras reserved for evaluation. For each target camera $c_{\mathrm{tgt}} \in \mathcal{C}\mathrm{tgt}$, we identify its nearest neighbor $c_{\mathrm{ref}} \in \mathcal{C}\mathrm{ref}$ according to pose-space distance. During training, a series of novel views is synthesized by interpolating the camera pose from c_{ref} toward c_{tgt} over multiple optimization iterations. As the 3DGS model parameters are updated, these interpolated views gradually converge to the held-out target poses, enabling quantitative evaluation on $\mathcal{C}_{\mathrm{tgt}}$.

4.2. Two-Step Diffusion Enhancer

To mitigate residual artifacts in rendered views, we employ the Stable Diffusion v1.5 model in a two-step denoising pipeline. Specifically, each input image is first encoded into the latent space and then iteratively refined over two diffusion steps. The model is conditioned on both a positive prompt—"A sharp, high-resolution photo of a road scene, no blur, no motion artifacts"—and a negative prompt—"blurry, warped, noisy, distorted, painterly." We configure the denoising strength parameter to 0.3 and the classifier-free guidance scale to 0.2. Under these settings, the diffusion process effectively suppresses noise and geometric distortions while preserving high-frequency detail, as demonstrated in Figures 3.

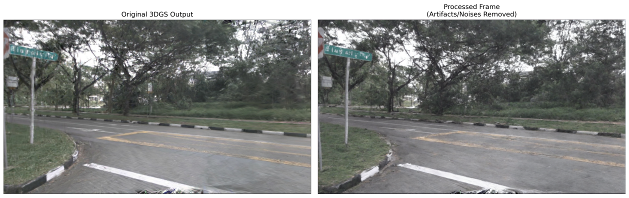

Figure 3. Result of applying the diffusion model to the 3DGS-generated output

4.3. 3DGS Optimization

We introduce a hyperparameter $T_{\mathrm{diff}} = 50$, meaning that every T_{diff} iterations we invoke the diffusion enhancer on all novel-view renders. Concretely, let $\mathcal{C}_{\mathrm{target}}$ and $\mathcal{C}_{\mathrm{novel}}$ denote the sets of held-out reference cameras (with ground-truth images I^c_{gt}) and interpolated novel camera poses, respectively. At iteration t, if $t \bmod T_{\mathrm{diff}} = 0$, then for each $c \in \mathcal{C}_{\mathrm{novel}}$ we perform

$$I^c_{\mathrm{rend}} = \mathrm{Render}(c), \quad I^c_{\mathrm{diff}} = \mathrm{DiffusionEnhancer}\big(I^c_{\mathrm{rend}}\big).$$

The overall loss used to optimize the 3DGS parameters is

$$\mathcal{L}\infty_{\mathrm{total}} = \sum_{c \in \mathcal{C}_{\mathrm{target}}} \mathcal{L}\big(I^c_{\mathrm{gt}}, I^c_{\mathrm{rend}}\big) + \lambda \sum_{c \in \mathcal{C}_{\mathrm{novel}}} \mathcal{L}\big(I^c_{\mathrm{diff}}, I^c_{\mathrm{rend}}\big),$$

where $\mathcal{L}(\cdot, \cdot)$ denotes a L_1 pixel-wise reconstruction loss. λ is a hyperparameter for regularization. Because the diffusion network is kept fixed (frozen), no gradients are back-propagated through it.

5. Experiments

5.1. Environment Setup

Configuring the full Di3DGS pipeline required a non-trivial orchestration of heterogeneous software stacks and GPU-accelerated containers. Initially, we built three separate Docker images—one for COLMAP, one for 3D Gaussian Splatting (3DGS), and one for Stable Diffusion v1.5—each leveraging NVIDIA's Container Toolkit to expose CUDA capabilities. During local development, all containers ran on an NVIDIA RTX A5500 Laptop GPU (Driver 570.133.20, CUDA 12.8, 16 GB VRAM). The Stable Diffusion container remained a severe bottleneck, anchored by its hardware requirement. On the A5500, invoking a two-step denoising pass took approximately six minutes per image. To overcome these constraints, we migrated to an NVIDIA GeForce RTX 4090 (Driver 575.51.03, CUDA 12.9, 24 GB VRAM). Once deployed on the RTX 4090 instance, the 3DGS + diffusion model training loop. It resulted in an end-to-end runtime of 1 hour 32 minutes 14 seconds.

5.2. Reconstruction Analysis

Shown below are several reconstructions showing the ground truth as well as before and after applying our diffusion model to the reconstruction. Out of the shown examples, we can qualitatively analyze the results. In the left target view of Figure 5.2, we can see that our method applied to the target view produced a much clearer image than without our diffusion model. The reconstruction is of fairly high quality, showcasing the capabilities of our method.

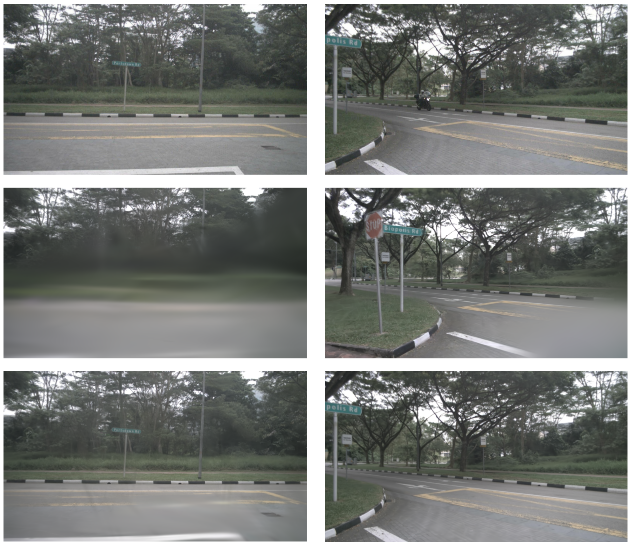

Figure 4. This figure shows two individual image reconstructions with the target view, one on the left column and one on the right column. Top: ground truth, Middle: without diffusion, Bottom: with diffusion.

In the right target view in Figure 5.2, however, while our method with the diffusion model does produce a better and nearly identical result to the ground truth, it is distinctively missing the motorcyclist who is present in the ground truth image. This omission may be attributed to limitations in the input conditioning data provided to the diffusion model. Since our approach relies on multi-view consistency and prior 3D reconstructions, transient or fast-moving objects like the motorcyclist may not be consistently captured across all input views or might be poorly represented in the Gaussian splatting stage. As a result, the diffusion model may interpret such inconsistent elements as noise and suppress them during the reconstruction process.

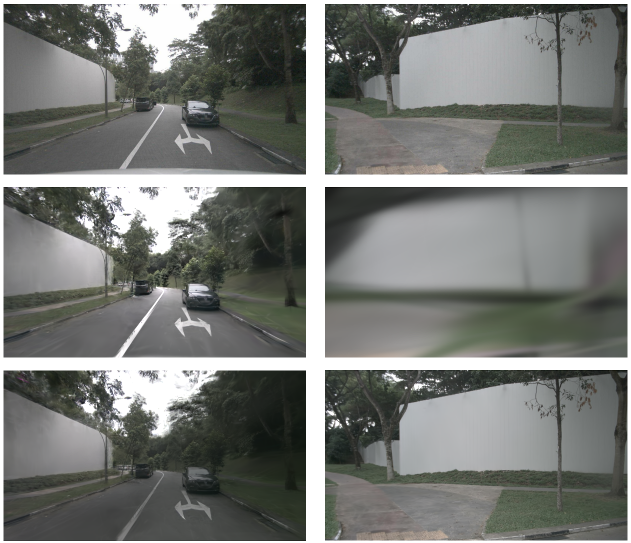

Figure 5. This figure shows two individual image reconstructions with the novel view, one on the left column and one on the right column. Top: ground truth, Middle: without diffusion, Bottom: with diffusion.

This highlights a potential failure mode of our method in handling dynamic objects, especially when they are not well-represented in the underlying geometric prior. In the left novel view shown in Figure 5.2, we can see a case where the diffusion output is noticeably darker than the non-diffusion and the ground truth images. This may be attributed to the diffusion model's tendency to favor smoother, globally consistent shading, which can sometimes lead to underexposed outputs in regions with complex lighting. Additionally, the darker lighting might indicate that the model inferred shadows or ambient occlusion effects that were not fully captured in the original 3DGS reconstruction. While this results in a more visually coherent and less artifact-prone image as the image without diffusion appears over-exposed to light which loses detail in some areas, the darker diffusion image also reduces contrast and obscures finer details when compared to the ground truth. Additionally, the diffusion image doesn't represent the actual brightness of the ground truth which the non-diffusion image does capture, even though the overall details and quality of the diffusion output are more accurate. This highlights a trade-off between perceptual realism and photometric accuracy in the novel view synthesis pipeline.

In the right novel view shown in Figure 5.2, however, we can see drastic improvements over the non-diffusion output. The diffusion image is very similar to the ground truth while the non-diffusion image is very blurry and loses most of the image's detail. This suggests that the diffusion model effectively refines the geometry and texture reconstruction in challenging regions where the base 3DGS rendering struggles. The structures in the image are effectively preserved with much higher fidelity in the diffusion output. The sharper appearance and improved semantic alignment with the ground truth indicate that the model is not only denoising the input but also enhancing structural consistency across views. This reinforces the effectiveness of integrating learned priors from diffusion models into the reconstruction pipeline, particularly for synthesizing novel views where the original data may be sparse or noisy.

5.3. Quantitave Analysis

Table 1 summarizes the reconstruction quality of a standard 3D Gaussian Splatting (3DGS) pipeline against our enhanced model across three evaluation metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

Model	PSNR	SSIM	LPIPS
3D gaussian splatting	21.9095363	0.79961448	0.3499600
Our Model	25.6269416	0.84147250	0.286909192

Table 1. Quantitative evaluation between 3DGS and our model.

1. PSNR (Peak Signal-to-Noise Ratio) The PSNR metric is defined as

$$\mathrm{PSNR} = 10\log\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right),$$

where MAX is the maximum possible pixel value (e.g., 1.0 for normalized images) and MSE is the mean-squared error between the reconstructed image and the ground-truth. A higher PSNR indicates lower average pixel-wise error. From Table 1, 3DGS attains $\mathrm{PSNR} \approx 21.91\,\mathrm{dB}$, whereas our model achieves $\mathrm{PSNR} \approx 25.63\,\mathrm{dB}$, representing an improvement of approximately $3.72\,\mathrm{dB}$. This gain implies that our model's pixel-level fidelity is substantially better: subtle color shifts and noise present in 3DGS renders are significantly reduced, bringing reconstructed intensities much closer to the reference photographs.

2. SSIM (Structural Similarity Index) SSIM evaluates perceived image quality by comparing luminance, contrast, and structural correlation over local patches. Its value ranges from 0 to 1, where 1 indicates perfect structural agreement. Table 1 shows:

$$\mathrm{SSIM}_{\mathrm{3DGS}} \approx 0.7996, \quad \mathrm{SSIM}_{\mathrm{Ours}} \approx 0.8415.$$

An increase of approximately 0.0419 in SSIM means our reconstructions preserve edges, textures, and geometric details more faithfully, especially for thin structures (e.g., foliage, railings) and fine surface patterns (e.g., brickwork). Whereas 3DGS tends to blur or wash out such details under sparse-view conditions, our model maintains higher local contrast and sharper structural fidelity.

3. LPIPS (Learned Perceptual Image Patch Similarity) LPIPS measures perceptual similarity by comparing feature activations extracted from a pre-trained deep network (e.g., VGG or AlexNet). Lower LPIPS values correspond to reconstructions that are more perceptually similar to the ground-truth. The values in Table 1 are:

$$\mathrm{LPIPS}_{\mathrm{3DGS}} \approx 0.34996, \quad \mathrm{LPIPS}_{\mathrm{Ours}} \approx 0.28691.$$

A reduction of about 0.0630 in LPIPS indicates that, in deep feature space, our rendered images appear significantly more realistic. This improvement translates to crisper lighting gradients, more accurate shading, and fewer “floaters” or distorted patches in areas where 3DGS struggles to constrain geometry.

4. Summary

- *Photometric Accuracy (PSNR):* Our model reduces pixel-wise MSE by over half (as evidenced by a $\sim$4 dB PSNR gain), meaning reconstructed intensities align much more closely with real photographs.

- *Structural Fidelity (SSIM):* Boosting SSIM from $\sim$ 0.80 to $\sim$0.84 reflects a clear enhancement in preserving geometric edges and local contrast which is critical for accurately rendering thin or high-frequency structures.

- *Perceptual Realism (LPIPS):* A $\sim$ 0.06 decrease in LPIPS shows that the perceptual “look and feel” of the images is significantly more convincing to human observers, with fewer unnatural textures or ghosting artifacts.

Taken together, these quantitative gains confirm that our diffusion-based artifact removal and distillation strategy, when integrated into the 3DGS pipeline, yields reconstructions that are both quantitatively closer (in PSNR/SSIM sense) and perceptually more faithful (per LPIPS) to the ground-truth compared to vanilla 3DGS.

6. Conclusion

In this work, we have introduced **Diffusion-Guided Gaussian Splatting (Di3DGS)**, a hybrid reconstruction framework that leverages a pretrained diffusion denoiser to correct artifacts in 3D Gaussian Splatting (3DGS) under sparse-view, autonomous-driving scenarios. Our main findings and contributions can be summarized as follows:

1. **Significant Improvement with Fewer Views.** By integrating a two-step diffusion enhancer (Stable Diffusion v1.5) into the 3DGS training loop, Di3DGS dramatically reduces common artifacts, such as floating or “fuzzy” Gaussians, color bleeding, and geometric warping, even when only six camera views are available. Quantitatively, compared to vanilla 3DGS, our method raises PSNR from $\sim$ 21.91 dB to $\sim$ 25.63 dB (+3.72 dB), increases SSIM from $\sim$ 0.80 to $\sim$ 0.84 (+0.04), and lowers LPIPS from $\sim$ 0.35 to $\sim$ 0.29 (–0.06). These gains are remarkable given that many competing techniques employ 10+ cameras or LiDAR priors. In short, **with just six RGB images**, Di3DGS achieves reconstruction fidelity on par with, or better than, methods that rely on far denser sensing. This suggests that diffusion-guided supervision effectively “hallucinates” plausible detail in underconstrained regions, allowing the 3DGS model to converge on high-quality geometry and texture with minimal input.

2. **Qualitative Benefits and Failure Modes.** Beyond the numerical metrics, our qualitative analysis shows that Di3DGS yields noticeably sharper edges and more coherent textures, especially around thin structures (e.g., roadside signs, poles) and fine-grained surfaces (e.g., painted road stripes). In scenes where vanilla 3DGS produces washed-out or “smeared” colors, Di3DGS outputs appear more photorealistic and closer to ground truth. However, we also observe that transient or fast-moving objects (e.g., a motorcyclist in one example) can be suppressed by the diffusion enhancer if they are not consistently visible across the six input views. This highlights an important failure mode: *dynamic elements* may be interpreted as “noise” by the denoiser and hence dropped from the reconstruction. Similarly, in regions with complex lighting, the diffusion model sometimes underexposes certain patches (favoring smoother shading) and may not faithfully reproduce absolute brightness. These trade-offs between perceptual realism and photometric accuracy point to areas for future refinement.

3. **Trade-off Between Speed and Quality.** One of the original appeals of 3D Gaussian Splatting is its *real-time performance*, with latencies measured in tens of milliseconds on commodity GPUs. Unfortunately,

injecting a diffusion denoiser into the training loop necessarily slows down the pipeline. On our local RTX A5500 (16GB VRAM), each two-step diffusion enhancement required $\approx$ 6 minutes per image; on an RTX 4090 (24GB VRAM), the overhead was reduced but remained substantial, resulting in an overall 1 h 32min 14s run time for 30,000 optimization iterations (with diffusion applied every 50 iterations). In other words, while Di3DGS significantly improves reconstruction quality, it cannot yet meet strict real-time demands in its current form. **Real-time applications, such as live-view rendering for intelligent vehicles, would require further acceleration or approximation strategies.**

4. **Key Lessons Learned.**

   - *Sparse-View Viability.* Even with only six cameras, a diffusion-augmented supervision signal can push a 3DGS model to reconstruct high-fidelity scenes. This suggests that **learned 2D priors** (e.g., single-step diffusion networks) are extremely effective at filling in missing geometry and texture when posed with underconstrained inputs.
   - *Diffusion as a Strong Prior, but at a Cost.* Integrating a frozen diffusion network into a differentiable rendering loop can dramatically reduce artifacts, but the computational cost is non-negligible. Building a stable and efficient interface between a PyTorch-based denoiser and a CUDA-accelerated 3DGS backend required careful orchestration (multi-stage Docker builds, memory-efficient data pipelines) and continues to pose optimization challenges.
   - *Dynamic Content Is Hard.* Because our diffusion model was trained on static images, fast-moving objects (vehicles, pedestrians) often violated the multi-view consistency assumed by Di3DGS. Consequently, these dynamic elements were sometimes “infilled” as if they were artifacts, rather than preserved in the final reconstruction.

6.1. Future Directions

While Di3DGS represents a promising step toward artifact-free, sparse-view novel-view synthesis, several avenues remain open for improvement and new applications:

1. **Speed Optimization & Real-Time Feasibility.**

   - *Distilled or Quantized Denoiser.* Replace the current two-step Stable Diffusion model with a distilled, low-latency U-Net (e.g., a lightweight single-step diffusion network or a quantized version). This could reduce per-image inference time from minutes to seconds or sub-seconds.
   - *Asynchronous / Patch-Based Denoising.* Instead of running diffusion on entire full-resolution images, one could denoise only “problematic” patches or use a coarse-to-fine strategy, focusing computational budget where the 3DGS model is most uncertain.

2. **Handling Dynamic and Transient Scene Elements.**

   - *Dynamic Masking or Reprojection Consistency.* Introduce a module that identifies dynamic pixels (e.g., using optical flow or per-view consistency checks) and prevents the diffusion enhancer from “removing” them.
   - *Joint 2D-3D Motion Models.* Incorporate a motion-aware diffusion prior that conditions on both spatial and temporal context (e.g., feeding multiple time-adjacent frames into the denoiser) to better preserve moving objects.

3. **Generalization to Other Domains.**
   While we have focused on autonomous-driving scenes, Di3DGS could be extended to *other fields* where real-time or near-real-time 3D reconstruction is desirable but data are sparse or noisy:

   - *Robotics & SLAM.* For indoor or warehouse robotics, where a limited number of cameras (or a single RGB-D sensor) capture novel environments, a diffusion-guided pipeline could improve semantic mapping and obstacle avoidance under challenging lighting or clutter.
   - *Augmented/Virtual Reality (AR/VR).* Mobile devices and headsets typically rely on few cameras (or low-resolution depth sensors). Di3DGS-style denoising could allow for faster scene capture and more photorealistic virtual object insertion, even in handheld or wearable form factors.

4. **Extending to Multimodal Priors.**

   - *LiDAR + Diffusion.* Incorporate LiDAR-derived point clouds as a complementary prior to the diffusion network, so that the denoiser no longer has to “hallucinate” geometry from six camera views alone. By combining sparse yet accurate LiDAR depth with learned image priors, the system could more reliably reconstruct both static and dynamic elements.

- *Semantic Conditioning.* Train a diffusion model that takes semantic segmentation or object-detection heatmaps as additional conditioning inputs. This could allow the denoiser to explicitly respect object boundaries (e.g., cars, pedestrians) and avoid merging them into the background.

5. **Adaptive View Scheduling.**

    - *Confidence-Based Diffusion Triggers.* Instead of applying diffusion every 50 iterations, monitor a per-view uncertainty metric (e.g., 3D variance in Gaussian weights) and invoke diffusion only when the model is "unsure." This would save compute by avoiding unnecessary denoising when reconstructions are already confident.

In summary, Di3DGS demonstrates that **diffusion-based priors** can empower sparse-view 3D reconstruction to achieve surprising fidelity, far exceeding what vanilla 3D Gaussian Splatting produces when only six cameras are available. The quantitative boost in PSNR/SSIM/LPIPS and qualitative improvements in structure and texture confirm the efficacy of combining pretrained 2D denoisers with differentiable 3D renderers. At the same time, our study underscores the **trade-off between quality and speed**: integrating diffusion currently precludes strict real-time performance. For application domains where offline or near-real-time processing is acceptable such as large-scale mapping, urban-drive dataset curation, or AR/VR content generation, Di3DGS offers a powerful tool to reduce artifacts without requiring dense multi-camera rigs or expensive LiDAR arrays. Looking forward, we anticipate that further algorithmic optimizations (distilled denoisers, dynamic scheduling) and tighter GPU integration will enable the **next generation of real-time, diffusion-guided 3D reconstruction**, expanding the reach of 3DGS-style rendering into ever more resource-constrained and dynamic environments.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 42, 2023. 2

[2] Q. Tian, X. Tan, Y. Xie, and L. Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 2

[3] J. Z. Wu, Y. Zhang, H. Turki, X. Ren, J. Gao, M. Z. Shou, S. Fidler, Z. Gojcic, and H. Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv preprint arXiv:2503.01774*, 2025. 2

[4] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*, 2024. 2