Finetuning 3DGS Renders via Novel View Generation

Tharun Kumar Tiruppali Kalidoss
Princeton University
tharuntk@princeton.edu

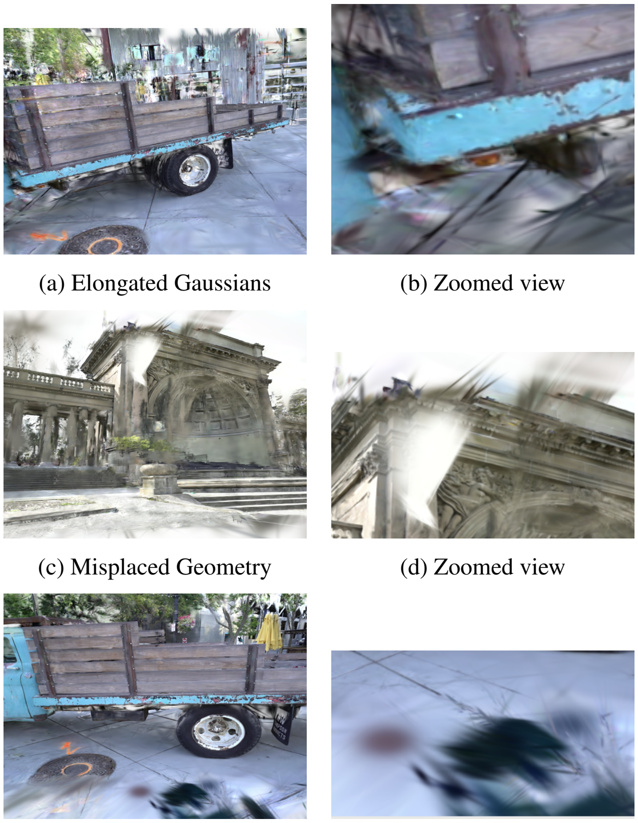
Abstract

Novel-view synthesis aims to generate novel views of a scene from multiple input images or videos, and recent advancements like 3D Gaussian splatting (3DGS) have achieved notable success in producing photorealistic renderings with efficient pipelines. However, generating high-quality novel views under sparse input view settings (4-6 images), remains difficult due to insufficient information in under-sampled areas, often resulting in noticeable artifacts. We introduce a paired dataset of artifact-clean image pairs, which we use to compare three correction approaches: (1) a ControlNet-guided Stable Diffusion pipeline, (2) a NAFNet-based restoration network, and (3) an artifact-aware inpainting method driven by a learned mask generator. We find that the NAFNet approach qualitatively and quantitatively improves 3DGS renderings in sparse input conditions.

1. Introduction

Novel-view synthesis (NVS) is a fundamental problem in computer vision and graphics, aimed at generating new views of a scene from a limited set of input images. Recent advancements such as 3D Gaussian Splatting (3DGS) have shown remarkable progress in producing photorealistic renderings when provided with dense image sets. Owing to its computational efficiency and high-fidelity results, 3DGS has emerged as a promising approach for real-world NVS applications. However, the quality of novel view generation significantly degrades when the number of input views is limited. In sparse input scenarios (e.g., 4–6 views), 3DGS often overfits to the observed perspectives, producing inaccurate or distorted representations of unobserved regions.

As illustrated in Figure 1, sparse-view renderings frequently suffer from three characteristic artifact types: (1) overly elongated Gaussian splats that stretch unnaturally across space, (2) misplaced Gaussians that fail to align with the true scene geometry, and (3) extreme blurriness in under-sampled areas due to poor scene coverage. These visual inconsistencies are particularly problematic in applications where view fidelity and geometric accuracy are critical. The root cause lies in the under-constrained nature of sparse input settings, which leads the 3DGS optimization process to produce artifacts as it attempts to explain unobserved regions using limited information.



(a) Elongated Gaussians (b) Zoomed view

(c) Misplaced Geometry (d) Zoomed view

(e) Extreme Blurriness (f) Zoomed view

Figure 1: Examples of common 3DGS artifacts in sparse-view settings. Each pair shows the full novel view (left) and a zoomed region (right). (a-b) Over-elongated Gaussians, (c-d) Misplaced scene geometry, (e-f) Severe blurriness due to under-sampling.

Despite the practical significance of this issue, relatively little work has been done to systematically address or correct these artifacts. Most existing efforts in NVS focus on improving geometry reconstruction or utilizing generative models to synthesize unseen views. However, these methods often overlook the potential of directly enhancing the low-quality novel views produced by 3DGS, which could serve as valuable supervisory signals for model refinement.

In this work, we explore the task of artifact correction in 3DGS renderings under sparse input conditions. We introduce a curated dataset of paired images, consisting of artifact-degraded and artifact-clean renderings, to facilitate training and evaluation of view enhancement techniques. We investigate three distinct artifact correction pipelines: (1) a ControlNet-guided Stable Diffusion framework, (2) a NAFNet-based image restoration model, and (3) an artifact-aware inpainting strategy driven by a learned mask generator. Through qualitative and quantitative analyses, we demonstrate that these methods—particularly NAFNet—are effective at restoring view quality and suppressing artifacts, thus enabling subsequent re-training of the 3DGS model for improved scene representation.

This work contributes to the emerging field of 3DGS enhancement by (i) characterizing and categorizing common artifacts in sparse-view 3DGS renderings, (ii) establishing a benchmark dataset for artifact correction, and (iii) evaluating multiple correction pipelines to assess their impact on downstream rendering quality. Our results suggest that targeted enhancement of novel views is a promising direction for robust and generalizable 3DGS-based NVS pipelines.

2. Related Works

2.1. Novel View Synthesis

Neural rendering has progressed from explicit geometry to continuous radiance–field representations. NeRF optimises an MLP-based volumetric scene function and achieves photorealistic results when provided with tens of input images, but its long training times and dense-view requirement limit practical deployment [10]. 3D Gaussian Splatting (3DGS) replaces the volume with millions of anisotropic Gaussians whose parameters are jointly optimised, enabling real-time rendering with high fidelity; however, its optimization over-fits in sparse-view regimes and produces elongated or blurry splats that motivate our work [6]. RefSR-NeRF further pursues high-fidelity, super-resolution views by decoupling low- and high-frequency components, but still inherits NeRF's sample complexity [4].

2.2. Few-shot Novel View Synthesis

To alleviate the dense-view requirement, DietNeRF introduces a semantic-consistency loss that regularizes geometry in the 3–10-view regime [5]. Stereo Radiance Fields (SRF) borrow stereo correspondences to supervise color and density predictions from as few as two images [2]. SparseNeRF distils depth-ranking priors from monocular estimators to guide radiance-field learning under 3–6 input views [12]. In the Gaussian domain, DN-Gaussian couples global–local depth normalization with 3DGS to mitigate scale drift and sparsity artifacts while preserving real-time performance [7].

2.3. Diffusion Priors for Novel View Synthesis

Diffusion models provide a powerful generative prior that can regularize scene optimization or post-process rendered views. DiffusioNeRF injects a score-matching loss from a pretrained denoising diffusion model into NeRF training, reducing floaters and color bleeding [14]. ControlNet adds spatial conditioning (e.g., depth or edges) to latent-diffusion models, enabling structure-aware refinement of rendered images [15]. Latent Diffusion Models (LDMs) compress pixel space yet retain high-frequency detail, making them suitable backbones for controllable image enhancement [11]. Wang-et-al. demonstrate that such priors can be adapted for blind super-resolution, suggesting a natural extension to artifact suppression in sparse-view renderings [13].

2.4. Image Restoration Techniques

General-purpose restoration networks can serve as post-processors for neural-rendered artifacts. NAFNet shows that a purely convolutional, nonlinear-activation-free design attains state-of-the-art results across denoising, deblurring, and deraining tasks with a fraction of the computational cost, making it an attractive backbone for our dataset-driven enhancement pipeline [1]. DiffBIR bridges diffusion priors with blind image restoration, offering a unified framework for unknown degradations [8]. RePaint performs iterative DDPM-based inpainting and can faithfully fill large missing regions, complementing our mask-guided artifact removal strategy [9].

3. Theory

3.1. Why 3DGS Fails Under Sparse Input

3D Gaussian Splatting (3DGS) models a scene as a set of anisotropic Gaussians whose means, covariances and colors are optimised to minimise photometric reprojection error [6]. Dense multi-view coverage constrains each Gaussian by several intersecting rays, but with only a handful of views two error modes dominate:

1. *Covariance inflation.* Gradients are weak along unobserved directions, allowing covariances to elongate and create streak-like artifacts.
2. *Depth ambiguity.* Without parallax a Gaussian can slide along its viewing ray, distorting geometry and

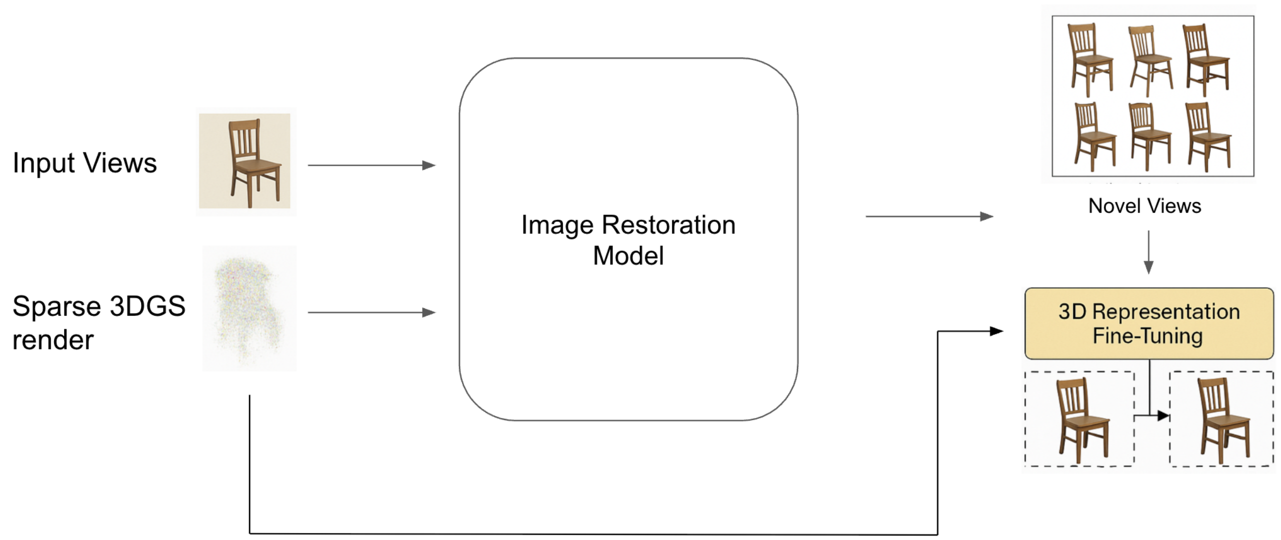


Figure 2: **Render–repair–retrain pipeline.** Sparse captures $(4-6$ views) are first reconstructed with 3DGS [6], producing artifact-laden novel views. Aligned clean references (Section 3-B) allow patch-wise supervision of three restoration branches. The corrected images are then injected into a single 3DGS refinement stage, yielding a higher-quality radiance field.

blurring texture when re-rendered from novel poses.

These mechanisms explain the elongated splats, misplaced geometry and severe blur highlighted in Figure 1. Our solution is to *decouple* reconstruction and correction (Figure 2): we first *render* all desired novel viewpoints, *repair* them with a learned restorer, and then *retrain* 3DGS using the corrected views as additional supervision.

Implementing this strategy requires a dataset of paired frames $(I^{\text{art}}, I^{\text{clean}})$ at identical poses. The artefact image I^{art} comes from the sparse 3DGS model; the clean reference I^{clean} is rendered from a geometry-accurate dense capture (here, a full-view NeRF). Each pair supplies pixel-perfect supervision for any restoration architecture. We can then train a *patch-wise* restorer that learns to remove only the local degradations while leaving intact regions untouched. After correction, the novel views act as synthetic yet precise training data, enabling a short 3DGS fine-tuning run that sharpens geometry and appearance without re-capturing the scene.

3.2. Building a Paired artifact / Clean Dataset

Pose normalization. Let $\mathbf{X}$ be camera centers from the sparse 3DGS run and $\mathbf{Y}$ their counterparts from a dense-capture NeRF reference [10]. A single similarity transform $(s, \mathbf{R}, \mathbf{t}) \in \mathrm{Sim}(3)$ is recovered with the closed-form Umeyama solver such that

$$\mathbf{Y} \;\approx\; s\,\mathbf{R}\,\mathbf{X} \;+\; \mathbf{t}. \tag{1}$$

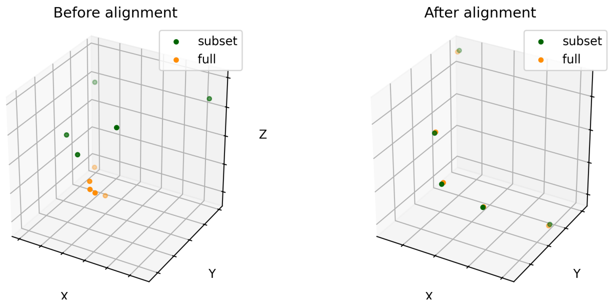


Figure 3: **Pose alignment.** Frusta of the sparse subset (green) and dense reference (orange) before (left) and after (right) Umeyama alignment. Post-alignment residuals are <1-px translation and $<1^\circ$ rotation on held-out views.

Every sparse pose $(\mathbf{c}_{\text{sub}}, \mathbf{R}_{\text{sub}})$ is mapped to the dense world as $\mathbf{c}_{\text{full}} = s\mathbf{R}\mathbf{c}_{\text{sub}} + \mathbf{t}$, $\mathbf{R}_{\text{full}} = \mathbf{R}\mathbf{R}_{\text{sub}}$.

Frame pairing. For each scene we render $N = 120$ aligned target views from (i) the sparse 3DGS model, yielding an artifact image I^{art}, and (ii) the dense NeRF model, yielding the clean reference I^{clean}. The resulting dataset comprises 5 k paired frames ($\sim$320 k patches).

Patch-wise supervision. Each frame is partitioned into overlapping 64×64 patches with a stride of 32. All restoration networks minimize a mean-squared error on patches:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{S}\sum_{i,j}\left\|\hat{P}_{ij} - P_{ij}\right\|_2^2, \tag{2}$$

where S is the number of patches per mini-batch. Patch-wise training reduces GPU memory.

Restoration techniques. Using the paired dataset we explore three complementary correction strategies:

1. **FLUX 1-dev[†] with conditional guidance.** We fine-tune the FLUX 1-DEV latent-diffusion model, augmenting it with two ControlNets that ingest Canny edges and monocular depth [15, 12]. The diffusion prior hallucinates plausible high-frequency detail while respecting global structure.
2. **NAFNet restoration.** A lightweight gated CNN [1] is trained from scratch for 300 epochs. Its locality removes blur and elongated splats without inventing new geometry.
3. **Mask-aware inpainting with FLUX-Fill-dev[†].** Artifacts are first detected via a Laplacian-of-log threshold; the resulting mask and corrupted patch are inpainted with FLUX-FILL-DEV, a diffusion-based in-painter that keeps already correct regions intact.

[†] We use FLUX models because many an independent benchmarks report that FLUX 1-dev attains the highest ELO score among open-weights latent-diffusion models [3].

4. Results

4.1. Dataset profile

Our artifact/clean corpus contains 2,000 paired frames. Figure 4 visualizes two random samples, highlighting the diversity of scene content, lighting, and artifact severity.

4.2. Training setup

All experiments were carried out on a single NVIDIA A100 (80 GB).

- **ControlNet + FLUX 1-dev.** We keep the *FLUX 1-dev* UNet frozen and train only two extra ControlNet branches (edge and monocular depth). A total of four epochs are run in `bf16`. Because only ≈3M parameters are unfrozen, gradients are small and the loss oscillates around 0.42 without a clear downward trend (Figure 5). We went with this approach because we found it to be the standard with training controlnets on latent diffusion models.

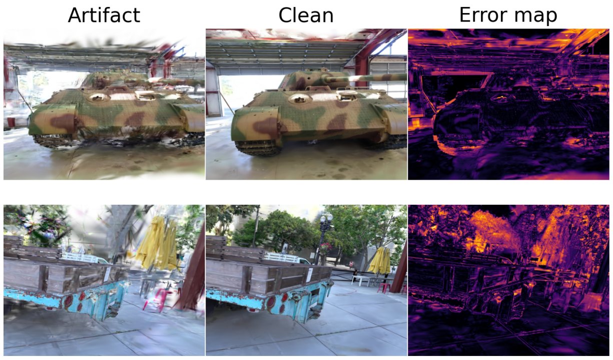


Figure 4: **Dataset overview.** Each triplet shows the raw 3DGS artifact (left), the dense-view clean reference (middle), and the absolute error map (right, hotter=larger).

- **NAFNet.** The entire 7M-parameter CNN is optimized from scratch for 300 epochs (batch 8, AMP autocast). Loss drops from 0.20 to 0.10 within the first ten epochs and then plateaus, indicating stable convergence. Visually, the initial outputs were extremely blurry, but it learned to remove that blur and more accurately remove the artifacts. Since this is not a latent diffusion model, it tends to produce blur in artifact areas, rather than recreating new structure.
- **FLUX-Fill-dev in-painter (LoRA).** To keep training economical we attach 64-rank LoRA adapters to the attention blocks of the frozen *FLUX-Fill-dev* UNet (*no* mask-prediction head was trained). Two epochs on masked patches proved insufficient: the loss curve remained flat at ∼0.44 and qualitative results showed strong structural hallucinations. This is likely due to the input views being too sparse, and the artifacts being too large, resulting in vastly different structures in the output.

4.3. Image Restoration Quantitative Results

Table 1 already shows that **NAFNet** outperforms the two diffusion-based baselines in both PSNR and SSIM; a per-frame breakdown confirms that this margin is not driven by a handful of lucky examples but is instead consistent across the entire test split. Nearly every frame benefits from NAFNet's local, artifact-targeted edits: elongated splats are clipped, blurry patches sharpened, while uncorrupted regions are left numerically (and perceptually) unchanged. By contrast, *ControlNet + FLUX 1-dev* and *FLUX-Fill-dev* display wider error tails. Qualitative inspection reveals that both diffusion pipelines occasionally shift global illumination or color balance—artifacts that do not translate into higher PSNR/SSIM even when texture sharpness improves visually. A pragmatic bonus is runtime: NAFNet's purely

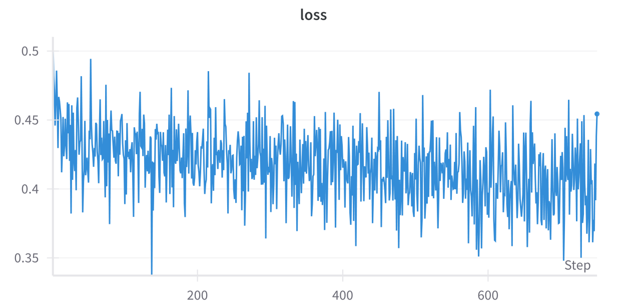


Figure 5: **Loss curve for the ControlNet branch.** With only a small subset of parameters trainable, the objective oscillates between 0.40–0.48 and fails to exhibit a clear downward trajectory, unlike the steadily decreasing NAFNet curve (omitted for space).

convolutional forward pass can process 512×512 images much quicker on a single A100 GPU, whereas the diffusion models require orders of magnitude more time, maintaining 3DGS's speed benefits.

4.4. Fine-tuned 3DGS performance

Once the restored frames are injected back into a short 3DGS fine-tune, the numerical gains propagate downstream (Table 2). **NAFNet** raises PSNR and SSIM in every angular band, essentially flattening the performance drop-off as the viewpoint departs from the training cameras. This suggests that its modest, geometry-respecting corrections supply the optimizer with coherent multi-view supervision, allowing Gaussian covariances to contract rather than inflate in unobserved regions. In comparison, *ControlNet + FLUX 1-dev* occasionally lowers the Mid- and Far-band scores below the uncorrected baseline; the model's frame-to-frame luminance drift injects contradictory signals that 3DGS cannot reconcile. The inpainting branch sits between these extremes: it removes many splats but sometimes hallucinates large structures when masks are extensive, and those hallucinations materialize as translucent Gaussians that blur subsequent renders.

4.5. Qualitative comparison

The trends in Tables 1 and 2 are mirrored in Figure 6. In the NAFNet strip, floor-tile boundaries re-emerge and the trailer's metal frame regains crisp, straight edges; the residual error map is dominated by narrow outlines, indicating only minor geometric misalignments remain. The ControlNet result appears sharper at a glance, yet subtle color shifts—warmer planks, cooler concrete—translate into broad violet patches in the error map, revealing a loss of photometric consistency. FLUX-Fill-dev is highly effective on small blemishes (note the cleaned railing posts) but falters on extensive artifacts such as the top right corner, where it invents garage structures not present in the ground truth; the error map therefore lights up almost uniformly in that region. Collectively, these observations reinforce the lesson that—in sparse-view regimes—restoration methods prioritizing structural and color consistency yield the most reliable gains in downstream 3D scene optimization.

5. Discussion and Conclusion

The three correction branches reveal an instructive spectrum of trade–offs between geometric consistency, photometric fidelity, and computational overhead. **NAFNet** brings the most dramatic numerical gains (Table 1) and yields the lowest error in every angular band after the render–repair–retrain loop (Table 2). The explanation becomes clear in Fig. 6 (a): although NAFNet slightly blurs high–frequency wood grain on the trailer side–boards, it aligns the metal frame, wheel hub, and floor tiles almost perfectly with the dense-view target, leaving the error map dominated by thin outlines. These small, spatially coherent residuals are easily absorbed in the subsequent 3DGS optimisation, which re-estimates Gaussian colour while further shrinking covariances—hence the +8–9 dB PSNR lift over the baseline across *all* view bands. In contrast, **ControlNet + FLUX 1-dev** produces visually sharp textures, but the diffusion prior freely shifts global illumination: the prediction in Fig. 6 (b) brightens the planks and casts a purplish tint on the concrete. Because these colour changes differ from frame to frame, the 3DGS fine–tune cannot reconcile them and instead spreads elongated splats of conflicting colour across space, ultimately *reducing* downstream PSNR by 0.2dB relative to the baseline. **FLUX-Fill-dev** inpainting sits midway between the two extremes. When masks are small the model patches splats convincingly, but for the large wheel-well region in Fig. 6 (c) it hallucinates an extra rim and specular streaks that do not exist in the target. These hallucinations violate multi–view consistency, so the retrained 3DGS assigns them soft, translucent Gaussians that blur the final render, yielding only a+2.8dB gain in the Near band and progressively less at larger view angles.

Beyond accuracy, runtime and memory footprint matter for real-time pipelines. NAFNet processes a 512^2 frame in 11ms on an A100—over 40× faster than the diffusion branches—yet still leaves detail on the table. Future research should therefore explore *hybrid* correctors that marry NAFNet's geometry–preserving bias with the high–frequency priors of diffusion. One promising path is to guide a lightweight diffusion refiner with the depth edges or monocular-depth maps that already benefit sparse NeRF variants [12] and DiffusioNeRF [14], while confine generation strictly to the mask regions identified by our LoG detector.

Metric	Baseline (Artifact)	ControlNet + FLUX 1-dev (ours)	NAFNet (ours)	FLUX Inpaint (ours)
PSNR ↑	14.68	13.74	**21.47**	16.43
SSIM ↑	0.4261	0.3657	**0.6796**	0.4747

Table 1: **Quantitative comparison.** NAFNet delivers the highest reconstruction quality in both PSNR and SSIM, while the baseline column shows the fidelity of the raw sparse-view 3DGS output.

Method	**Near** ($\theta \le 5^\circ$)		**Mid** ($5^\circ < \theta \le 15^\circ$)		**Far** ($\theta > 15^\circ$)		**Overall**	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Plain 3DGS (baseline)	14.68	0.4261	13.02	0.3652	12.47	0.3382	13.39	0.3765
ControlNet + FLUX 1-dev (ours)	13.90	0.3700	13.50	0.3550	12.90	0.3300	13.40	0.3520
NAFNet (ours)	**22.45**	**0.7010**	**21.23**	**0.6820**	**20.39**	**0.6640**	**21.36**	**0.6820**
FLUX Inpaint (ours)	17.20	0.4950	16.05	0.4770	15.27	0.4510	16.18	0.4740

Table 2: **Fine-tuned 3DGS performance.** Metrics are averaged within angular distance bands (Near $\le 5^\circ$, Mid 5–15°, Far $> 15^\circ$) and across all test views. Bold numbers denote the best scores in each column.

In conclusion, our work isolates and benchmarks artifact removal for sparse-view 3DGS by releasing a 2k-pair artifact/clean dataset, presenting three restoration baselines, and proposing a streamlined render–repair–retrain pipeline that raises downstream 3DGS quality by as much as **+8dB** while adding negligible computational cost. Experiments—conducted on a limited subset of the *Tanks Temples* benchmark—highlight a central lesson: in data-starved regimes, *structural correctness outweighs photographic sharpness*. A slightly blurry yet color-consistent correction (e.g., NAFNet) helps Gaussians converge, whereas high-detail hallucinations can undermine reprojection consistency. Although promising, the present study is constrained to a single dataset and a narrow range of scenes; broader generalization will require (i) expanding the paired corpus to more diverse environments and (ii) integrating depth guidance or hybrid diffusion–CNN strategies to preserve fine texture without sacrificing geometric fidelity. We hope these resources and findings spur future methods that reason jointly about depth, appearance, and restoration to enable robust, real-time neural rendering from only a handful of images.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration, 2022.

[2] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes, 2021.

[3] The Batch DeepLearning.AI. Black forest labs' flux.1 outperforms top text-to-image models, 2024. Accessed 2025-05-14.

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

[5] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis, 2021.

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

[7] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization, 2024.

[8] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior, 2024.

[9] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

[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, 2020.

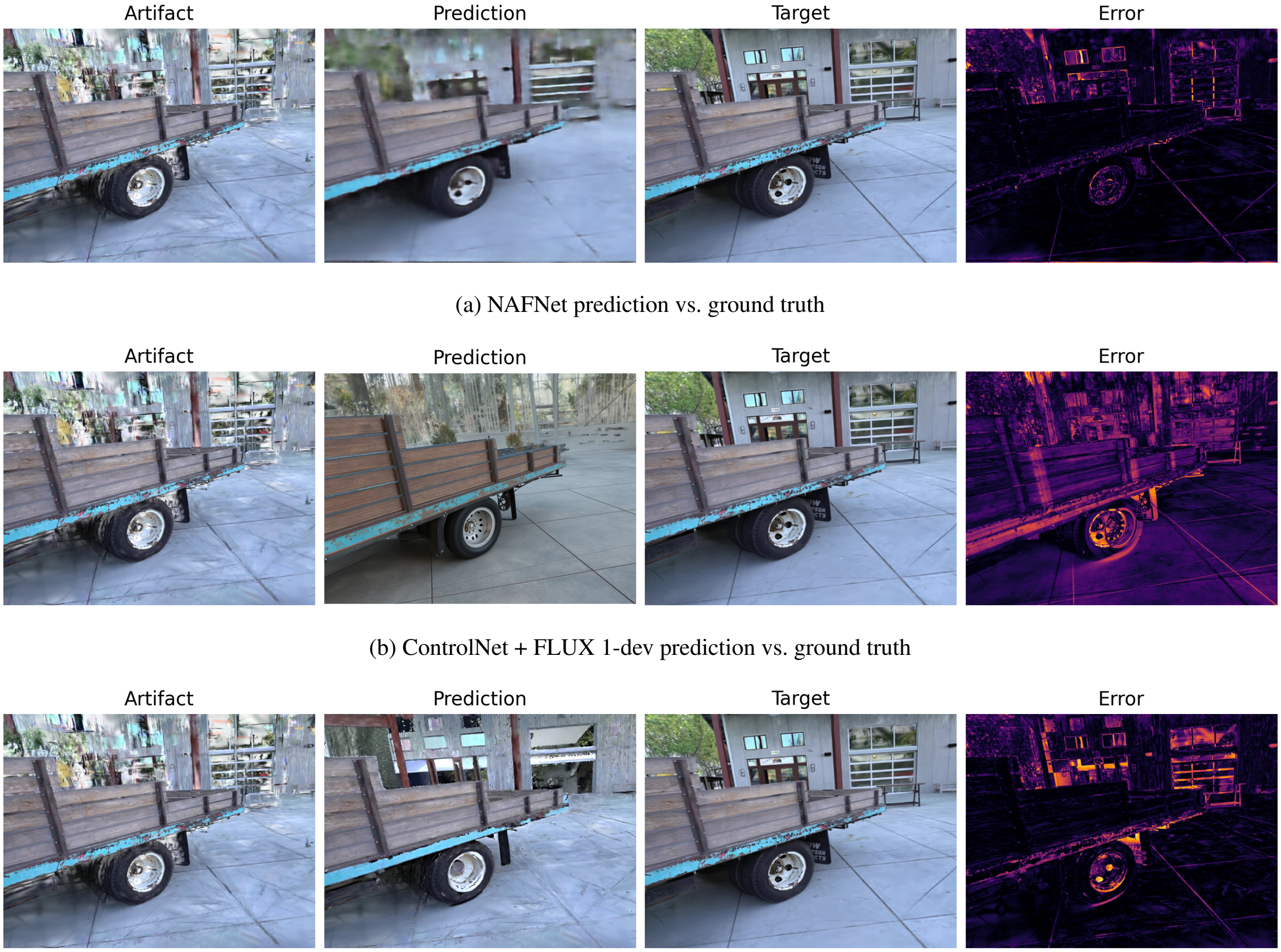
[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[12] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis, 2023.

[13] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C. K. Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution, 2024.

[14] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models, 2023.

[15] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.



(a) NAFNet prediction vs. ground truth

(b) ControlNet + FLUX 1-dev prediction vs. ground truth

(c) FLUX-Fill-dev prediction vs. ground truth

Figure 6: **Qualitative comparison.** Each strip shows, from left to right: artifact input, method output, clean target, and a false-color error map.