Cleaning Casually Captured Splatting Scenes With Diffusion Priors

Jeff Tan Joel Julin Bhuvan Jhamb Roshan Roy
Group 27, 16-825 Course Project, Spring 2024
{jefftan, jjulin, bjhamb, roshanr}@andrew.cmu.edu

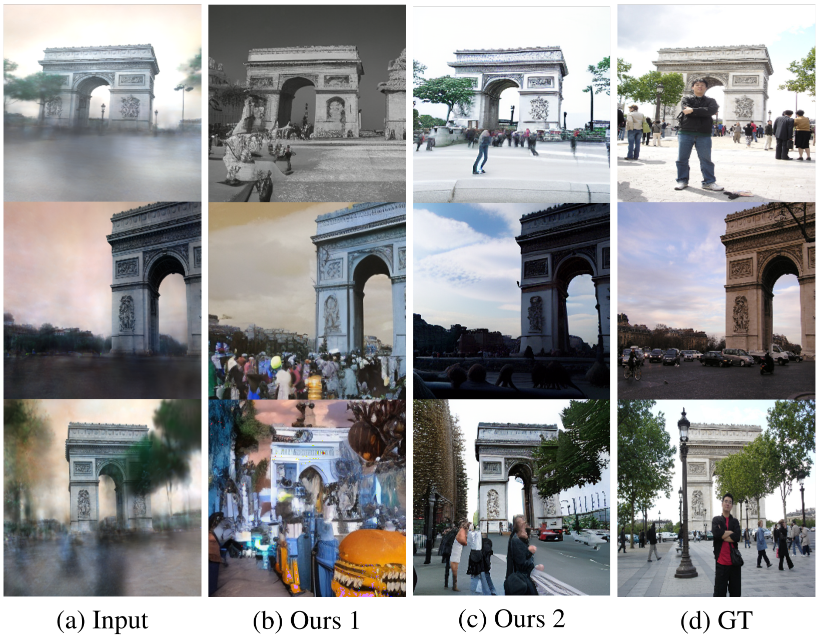

Figure 1. Gaussian splatting reconstructions of casually captured scenes often suffer from ghostly floater artifacts and incoherent geometry. Given a raw splatting render (a) that contains artifacts from an arbitrary novel viewpoint, our method fine-tunes an image-conditioned diffusion model to remove ghostly artifacts and infill plausible geometry. The outputted cleaned renders (b and c) have their artifacts removed, and better align with the ground-truth images (d). Column (b) shows our ControlNet+StableDiffusion architecture, which hallucinates aggressively, while column (c) shows our fine-tuned InstructPix2Pix architecture which yields the best quality.

Abstract

Gaussian splatting reconstructions of casually captured scenes often suffer from ghostly floater artifacts and incoherent geometry, especially when the splatting model is rendered at extreme views or when transient artifacts (e.g. cars) are present in the training images. Most prior splatting literature does not address these artifacts, as they evaluate on scenes with dense view coverage and no transients. To mitigate artifacts, our method fine-tunes image-conditioned diffusion models to remove ghostly artifacts and infill plausible geometry at arbitrary novel views. We use our method to clean up large-scale, real-world scenes, such that they appear plausible from extreme viewpoints and remain consistent with the observed views.

1. Introduction

Gaussian splatting has greatly improved the efficiency of novel-view synthesis from multiple posed RGB images. On established benchmarks (e.g. Tanks and Temples), common practice is to evaluate only on views close to the training trajectory. However, splatting reconstructions are most useful when they can be re-rendered from entirely new trajectories, generating imagery far beyond the training views. Such extreme-viewpoint re-rendering tends to produce pervasive floater artifacts and incoherent geometry (Fig. 2). These artifacts are especially prominent when the training images have only sparse view coverage, or are inconsistent due to the presence of frame-specific transient occluders (e.g. cars).

Given a casually captured splatting scene, it is challenging to detect and mitigate artifacts while remaining faith-

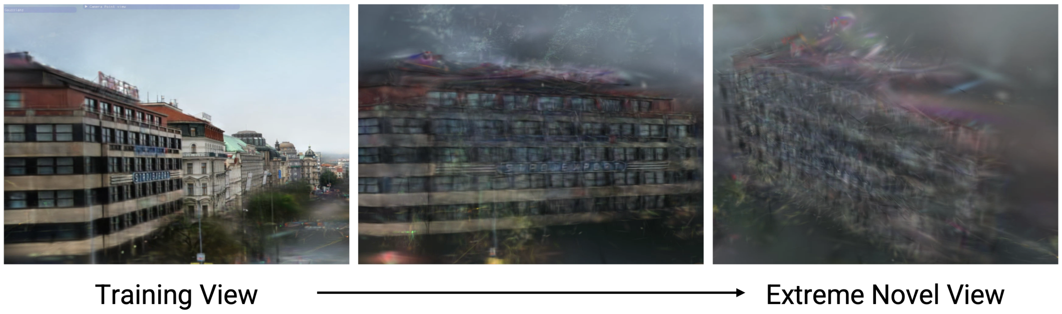

Figure 2. Example of Gaussian splatting artifacts at extreme novel views. The building's visual quality deteriorates significantly when the camera orbits from a training view (left) to an extreme novel view (middle and right).

ful to the observed views. Prior work such as Nerfbusters [12] uses a 3D diffusion prior to perform density control on volumetric NeRFs: Nerfbusters can remove implausible artifacts, but cannot generate new textures as it operates on density alone. Image-to-image latent diffusion models, such as Stable Diffusion Inpainting, do have impressive zero-shot capabilities to inpaint masked regions of existing scenes, however it require manual masks as input and is not designed to detect or mitigate the specific artifacts produced by gaussian splatting. Diffusion priors also frequently hallucinate new textures or objects which are inconsistent with the existing observations. Finally, several recent works leverage 2D diffusion models for generative editing of splatting reconstructions. These methods typically rely on text as input and are not capable of cleaning splatting artifacts out of the box. Beyond simply removing floater artifacts from novel-view renders, which can leave large gaps in the scene, our method aims to generate new geometry that plausibly completes the scene, while avoiding hallucination and remaining as faithful as possible to the observed views.

In this paper, we aim to detect and mitigate ghostly artifacts from raw novel-view renders of gaussian splatting scenes, by fine-tuning two foundational image-to-image diffusion models. We formulate this problem as a supervised learning task. First, we use a large library of casually captured outdoor scenes from the MegaDepth dataset [5], and generate a dataset that pairs raw novel-view splatting renders with the corresponding ground-truth images. Then, using this dataset, we fine-tune two image-to-image diffusion model architectures for the specific task of splatting artifact removal. We find that our ControlNet+StableDiffusion (CN+SD) architecture tends to heavily hallucinate scene contents, while fine-tuning InstructPix2Pix performs much better at removing ghostly artifacts from casually captured splatting scenes.

2. Related Works

2.1. Diffusion models for image editing

Latent diffusion models have been widely used for image generation and editing. Stable Diffusion [8] is a large-scale latent diffusion model that achieves state of the art results in text-to-image generation. Large-scale training over billions of text-image pairs allows Stable Diffusion to learn strong priors about what 3D scenes typically look like. Leveraging this foundational scene prior, InstructPix2Pix [1] use diffusion models to generate a large dataset of image editing examples, using these to train a conditional diffusion model to edit images according to human instructions. Although Stable Diffusion can only accept text as conditional input, ControlNet [15] is a framework for modifying StableDiffusion to accept image conditions such as depth maps and Canny edges. While the works above largely focus on image generation using text or image conditioning, our goal is to repurpose their foundational knowledge for mitigating gaussian splatting artifacts from novel-view renders.

2.2. Mitigating artifacts in Gaussian splatting

NeRF and Gaussian splatting reconstructions often suffer from ghostly artifacts when rendered at novel views. For example, incoherent geometry is often observed when a volumetric capture is re-rendered at an extreme novel view. Several prior works try to mitigate this. Nerfbusters [12] trains a local 3D diffusion model to perform density control on NeRFs. However, it leaves gaps in the scene and cannot infill new textures at missing scene regions. Concurrent work [2] directly repurposes the denoising process of text-to-image diffusion models for artifact removal, and introduce a bootstrapping technique to remove splatting artifacts from existing scenes by denoising sampled novel views. Our work attempts to solve the same problem via fine-tuning.

Another source of artifacts arises when the lighting, camera settings, or transient occluders in the scene vary across different training views, which is often the case in crowdsourced Internet-scale data [5]. In Gaussian splatting, such frame-specific artifacts are typically explained by floater artifacts very close to the impacted cameras. NeRF in the Wild [7] handles this by using a view-dependent appearance code to modulate the rendering at each view, and Gaussians in the Wild [14] extends this approach to Gaussian splatting. VastGaussian [6] explicitly models view-specific details in the training images, such as different lighting patterns, by passing both images and appearance embeddings into a post-processing convolutional network. Our method aims to clean up splatting reconstructions after the fact, and is orthogonal to this category of work which mitigates artifacts during training.

2.3. Editing radiance fields with diffusion priors

Several works leverage 2D diffusion models for generating or editing 3D radiance fields. DreamFusion [9] introduces score distillation sampling for generating 3D objects by supervising with pretrained 2D text-to-image diffusion models. DreamGaussian [10] improves the efficiency of 3D generation by using 3D gaussians instead of NeRFs as a scene representation. InstructNerf2Nerf [3] performs instruction-based editing of NeRFs by using InstructPix2Pix to iteratively update the dataset images [1]. InstructGS2GS [11] extends this approach to perform instruction-based editing of 3D gaussians. GaussCtrl [13] introduces depth conditioned editing and attention based latent code alignment to improve multi view consistency while editing.

3. Background

Our approach utilizes the differentiable Gaussian rasterization pipeline as demonstrated by [4].

Each 3D Gaussian is parameterized by a full 3D covariance matrix Σ, opacity α, mean position μ, and color represented by spherical harmonics (SH). Given the viewing transformation W and approximation of the affine transformation J, we follow [16] to obtain the 2D view-space covariance matrix Σ' to render the gaussians:

$$\Sigma' = JW\Sigma W^T J^T \tag{1}$$

A covariance matrix assumes physical meaning only if it is positive semi-definite, so it is challenging to directly optimize it with constraints on gradient descent to represent a scene's radiance field. [4] obscures this complexity by utilizing an alternate paramaterization which implictly maintains the positive semi-definiteness of the matrix. Intuitively, it views the covariance matrix Σ as a decomposition into an ellipsoid's scaling and orientation with rotation matrix R and scaling matrix S.

$$\Sigma = RSS^T R^T \tag{2}$$

We optimize the axes of the ellipsoid, R and S, which must be positive—instead of an unconstrained Σ. The color C is computed from reverse depth-sorted 2D gaussians via the standard volumetric rendering equation along a ray:

$$C = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))c_i, \tag{3}$$

with:

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j). \tag{4}$$

In the optimization process, we employ adaptive density control per [4] to control the density of the 3D Gaussians that best represent the scene. Consequently, the total number of Gaussians will change over the iterations.

4. Method

We treat cleaning artifacts from casually captured splatting scenes as a supervised learning problem. (1) In Sec. 4.1, we generate a dataset of raw splatting renders paired with ground-truth images, used to train generative models tailored to our task. (2) Then, we fine-tune two pretrained image editing diffusion backbones on this generated dataset: ControlNet+StableDiffusion (Sec. 4.2) and fine-tuning InstructPix2Pix (Sec. 4.3). Our model is able to generalize to cleaning splatting artifacts from unseen scenes. We describe both steps of our approach below.

4.1. Dataset Generation

In order to generate a dataset of raw splatting renders paired with ground-truth images, we leverage the MegaDepth dataset [5]. Megadepth contains 300K images from 196 casually captured large scale outdoor scenes depicting famous landmarks, with COLMAP SfM outputs for each scene. As this data is crowdsourced from large-scale Internet image collections, the images encompass a wide range of global lighting conditions, as well as transient occluders (e.g. people and cars). These variations across training views introduce artifacts in the Gaussian splatting reconstructions, particularly at ground level where the transient occluders are most numerous. We reconstruct MegaDepth scenes with 3D Gaussian Splatting and sample training and novel views for re-rendering.

To simulate the category of artifacts caused by rendered views being distant from training views, we only use 25% of the available images per scene during training. The remaining 75% of available images contain artifacts only visible from novel views, such as floater Gaussians or incoherent geometry. Presently, the held-out views in each scene are randomly chosen. However, we also tried performing k-means clustering of camera centers and randomly holding out clusters of rendered views.

For the artifact removal task, we treat ground-truth images as training targets and rendered views as input image

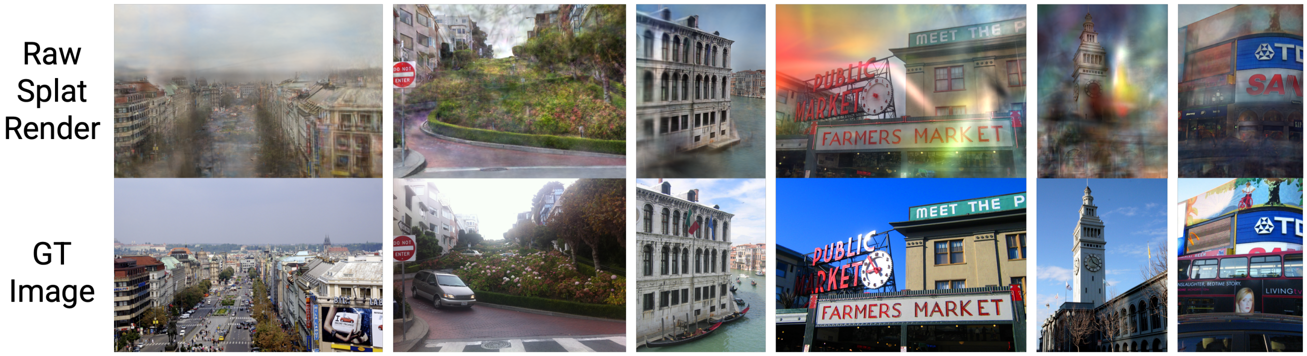

Figure 3. Examples from our generated dataset (Sec. 4.1). We show the raw splatting render on the top row, and the ground-truth dataset image in the bottom row. Note the variety of Gaussian splatting artifacts, such as blurry and incoherent geometry, as well as floater gaussians with different colors.

conditions to the diffusion model. For our experiments, we generate 30k (rendered image, ground-truth image) pairs from six MegaDepth sequences for training, and use a seventh held-out MegaDepth sequence for evaluation. Fig. 3 shows some representative examples from the dataset.

4.2. ControlNet + Stable Diffusion

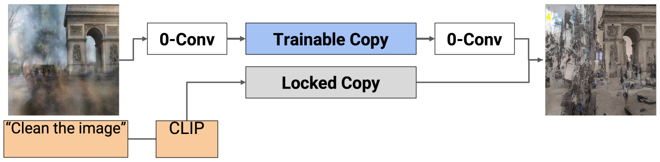

Figure 4. Our ControlNet+StableDiffusion (CN+SD) architecture. Given a raw splatting render as the input image, and a generic text prompt, our method produces a cleaned image with artifacts removed, using the ground-truth image as the training objective for Stable Diffusion with ControlNet. At each layer, trainable copy of the Stable Diffusion weights is connected to the locked copy by zero convolutions.

Our initial architecture aims to fine-tune Stable Diffusion for the specific task of artifact removal from Gaussian splatting renderings. To accomplish this, we employ the ControlNet framework [15]. ControlNet is an end-to-end framework for learning conditional controls for large pretrained text-to-image diffusion models. ControlNet preserves the quality and capabilities of the large model by locking its parameters, and also makes a trainable copy of its encoding layers. The trainable copy and locked copy are connected with zero convolution layers, with weights initialized to zeros such that they progressively grow during the training. The locked copy preserves the capabilities of the pretrained diffusion model, while the trainable copy reuses the pretrained model to learn a backbone capable of handling diverse input conditions. ControlNet shows the capability to control Stable Diffusion with various conditioning inputs, such as depth maps and Canny edges.

Specifically, suppose that $\mathcal{F}$ is a trained neural block with parameters Θ, that transforms an input feature map x into another feature map y as $y = \mathcal{F}(x; \Theta)$. To add ControlNet to this pretrained neural block, the parameters Θ of the original block are frozen, and simultaneously cloned to a trainable copy with parameters Θ_c. The trainable copy is connected to the locked model with zero convolution layers $\mathcal{Z}$ where both weight and bias are initialized to zeros. The complete ControlNet then computes the output y_c of the ControlNet block as follows:

$$y_c = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + \mathcal{Z}(c; \Theta_{z1}); \Theta_c); \Theta_{z2}) \qquad (5)$$

In our method, we provide Stable Diffusion with a noisy splatting render as the image condition, a standard prompt "clean the image" as the text condition, and the ground-truth image as the training target. After fine-tuning, the model is capable of identifying artifacts and infilling new geometry in their place. However, we find that this approach tends to hallucinate new spurious details everywhere in the scene, as Stable Diffusion is primarily an image generation model.

4.3. Fine-tuning InstructPix2Pix

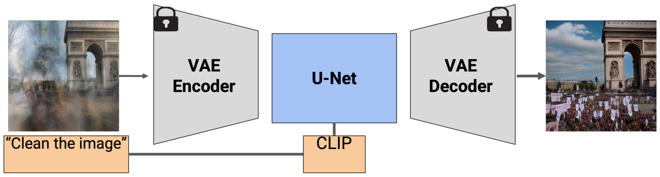

Figure 5. Our architecture for fine-tuning InstructPix2Pix (IP2P). Given a raw splatting render as the input image, and a generic text prompt, our method produces a cleaned image with artifacts removed, using the ground-truth image as the training objective.

ControlNet+StableDiffusion is carefully designed with zero convolutions, such that the trainable copy weights progressively grow during the training. This is necessary to ensure that the large-scale pretrained backbone is perfectly preserved at the beginning of training. Perfectly preserving the capabilities of Stable Diffusion is extremely helpful in the generative setting, but is not actually desirable in our case, where we want the model to carefully preserve the features of the conditioning image. Given this, the next architecture we tried was to adapt the InstructPix2Pix backbone for the task of cleaning artifacts from Gaussian splatting.

Diffusion models learn to generate data samples by passing raw Gaussian noise through a series of denoising autoencoders, which estimate the score of a data distribution. Latent diffusion models improve on the efficiency and quality of diffusion models by operating in the latent space of a pretrained variational autoencoder with encoder $\mathcal{E}$ and decoder $\mathcal{D}$. Given an an image x, the diffusion process adds noise to the encoded latent $z = \mathcal{E}(x)$, producing a noisy latent z_t where the noise level increases over timesteps $t \in T$. A network ϵ_θ is learned to predict the noise added to the noisy latent z_t, given the conditioning image c_I and the text instruction c_T. In our case, the conditioning image is the raw splatting render and the text instruction is a generic prompt such as ”clean this image”. We minimize the following latent diffusion objective:

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{E}(c_I), c_T))\|_2^2\right] \tag{6}$$

Similar to InstructPix2Pix, we initialize the weights of our model with a pretrained InstructPix2Pix checkpoint, leveraging its vast capabilities to edit images according to human instructions. To support image conditioning, z_t and $\mathcal{E}(c_I)$ are concatenated and passed through additional input channels to the first convolutional layer. We use the same text conditioning mechanism as Stable Diffusion and InstructPix2Pix.

5. Experiments

In this section, we evaluate each of our proposed methods on a held-out scene and present quantitative and qualitative results. Tab. 1 compares PSNR and LPIPS metrics across the proposed methods. In Sec. 5.1, we compare to an out-of-the-box InstructPix2Pix, used without fine-tuning. Sec. 5.2 shows the result of ControlNet + Stable Diffusion. Sec. 5.3 shows the result of fine-tuning InstructPix2Pix. Overall, we find the ControlNet + Stable Diffusion model heavily hallucinates spurious details in the image, while the fine-tuned InstructPix2Pix model does a better job at cleaning the splatting artifacts and adding missing details.

5.1. Baseline: InstructPix2Pix

As shown in Fig. 6, InstructPix2Pix is not designed to clean splatting artifacts from images, and therefore heavily hallucinates when used in our setup out-of-the-box. In all three examples, the context of the scene is completely different after processing by InstructPix2Pix, and the modified contents look extremely unnatural.

Table 1. Comparison of PSNR and LPIPS metrics on a held-out validation scene from MegaDepth. From top to bottom, we show metrics for the raw splatting render with artifacts, InstructPix2Pix baseline, and outputs from our CN+SD and fine-tuned IP2P models. Fine-tuned IP2P performs the best of all methods we tried, and removes artifacts while infilling blurry regions of the image. Although the PSNR metric favors the blurry regions of the raw splatting renders, our fine-tuned IP2P model achieves the best perceptual LPIPS scores.

Method	PSNR ↑	LPIPS ↓
Input Render	**15.35**	0.53
Baseline IP2P	5.78	0.73
CN+SD	9.58	0.66
Fine-tuned IP2P	13.02	**0.51**

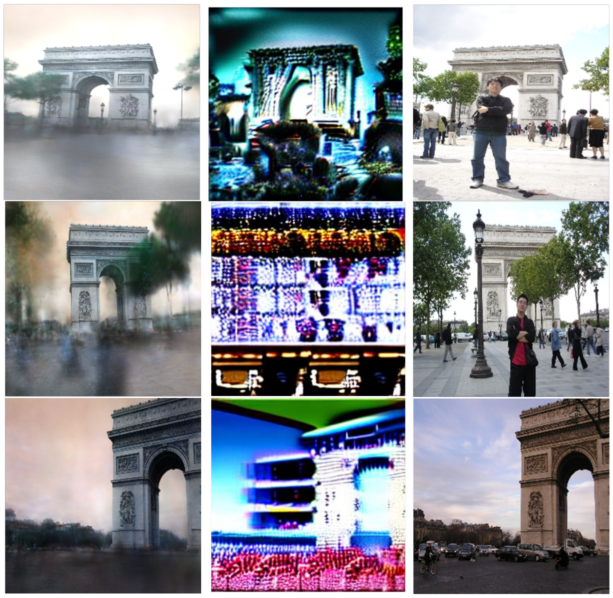
(a) Input (b) Baseline IP2P (c) GT

Figure 6. Qualitative results from the baseline InstructPix2Pix model. From left to right, we show (a) the input image, (b) cleaned splatting outputs, and (c) the ground-truth image. InstructPix2Pix is not designed to clean artifacts from input images, and in all three examples, we find that the raw IP2P model deviates significantly from the input image.

5.2. ControlNet + Stable Diffusion Results

As shown in Fig. 7, our ControlNet + Stable Diffusion model frequently hallucinates spurious details. We attribute this to the design of ControlNet, which perfectly preserves the capabilities of Stable Diffusion at initialization and is therefore best suited for generative tasks. At initialization, ControlNet’s zero convolutions are designed to zero out any influence from the conditioning image, meaning that in our setup the raw splatting render does not influence the result at first.

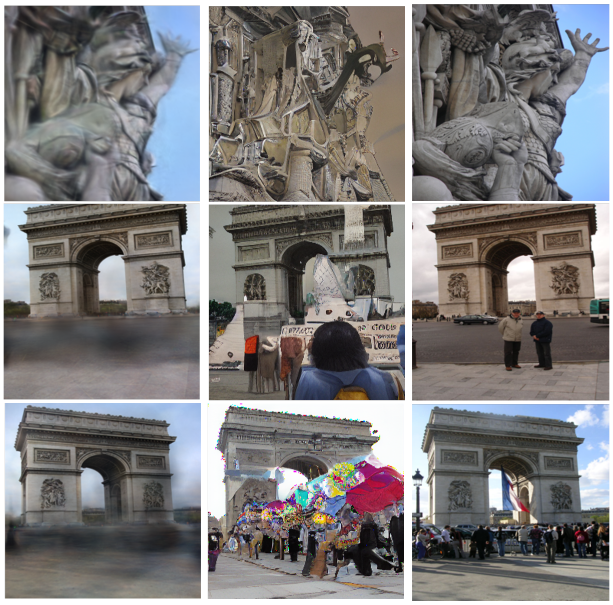

Figure 7. Qualitative results from the ControlNet + Stable Diffusion model. From left to right, we show (a) the input image, (b) cleaned splatting outputs, and (c) the ground-truth image. In all three examples, the CN+SD model deviates significantly from the input image and hallucinates spurious details, such as the elephant and newspaper text in the middle row and the colorful geometry on the bottom row.

5.3. Fine-tuned InstructPix2Pix Results

As shown in Fig. 8, the fine-tuned InstructPix2Pix model has the ability to plausibly remove blurry artifacts and replace them with the missing geometry. Most of the remaining mistakes are from global lighting variations or transient occluders, however it is unrealistic to expect the IP2P model to infill these exactly. Even though the global lighting and hallucinated transients do not exactly match the training images, we find that the predicted global lighting is reasonable, and the transients are indeed inserted into reasonable parts of the scene such as the people added to the courtyard.

6. Conclusion

Our project aims to automatically detect and clean artifacts from Gaussian splatting renders using generative diffusion priors. We found that a generative design, such as ControlNet + Stable Diffusion, is prone to aggressive hallucination that ignores the context of the scene, while fine-tuning the InstructPix2Pix architecture yields more faithful behavior that plausibly removes the artifacts.

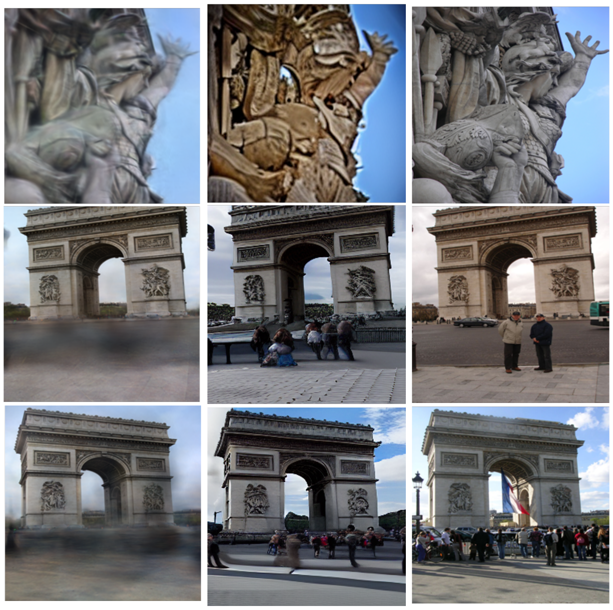

Figure 8. Qualitative results from the fine-tuned InstructPix2Pix model. From left to right, we show (a) the input image, (b) cleaned splatting outputs, and (c) the ground-truth image. In all three examples, the fine-tuned IP2P model enhances the blurry regions of the input images, predicting for example a tiled floor pattern in the middle row and a smooth courtyard with pedestrians in the bottom row.

Limitations. (1) Our proposed method only post-processes the Gaussian splatting renderings. Artifacts are still present in the original splatting scene, and might be resolved differently at different views. Improving the multi-view consistency of our method is an important, and could perhaps be addressed by incorporating multi-view conditioning into diffusion models, or by using our fine-tuned diffusion models to update the original splatting scene with SDS loss.

(2) Even though fine-tuned IP2P models produce results faithful to the input image, they are not faithful to the scene as a whole, often hallucinating transient occluders such as people and cars. We also noticed drastic lighting inconsistencies between the input and cleaned images. These inconsistencies can likely be mitigated by training on higher-quality data where global lighting variations and transient occluders are less prominent.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 2, 3

[2] Yifei Gao, Jie Ou, Lei Wang, and Jun Cheng. Bootstrap 3d reconstructed scenes from 3d gaussian splatting, 2024. 2

[3] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, 2023. 3

[4] 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):1–14, 2023. 3

[5] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[6] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *CVPR*, 2024. 3

[7] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections, 2021. 3

[8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 2

[9] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

[10] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 3

[11] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. 3

[12] Frederik Warburg*, Ethan Weber*, Matthew Tancik, Aleksander Hołyński, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[13] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Prisacariu. GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. In *ArXiv*, 2024. 3

[14] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections, 2024. 3

[15] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 4

[16] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS '01.*, pages 29–538, 2001. 3