DC-Gaussian: Improving 3D Gaussian Splatting for Reflective Dash Cam Videos

Linhan Wang[1*] **Kai Cheng**[3*] **Shuo Lei**[1] **Shengkun Wang**[1] **Wei Yin**[5]

Chenyang Lei[4] **Xiaoxiao Long**[2†] **Chang-Tien Lu**[1]

[1]Virginia Tech [2]Hong Kong University [3]USTC [4]CAIR, HKISI-CAS [5]University of Adelaide

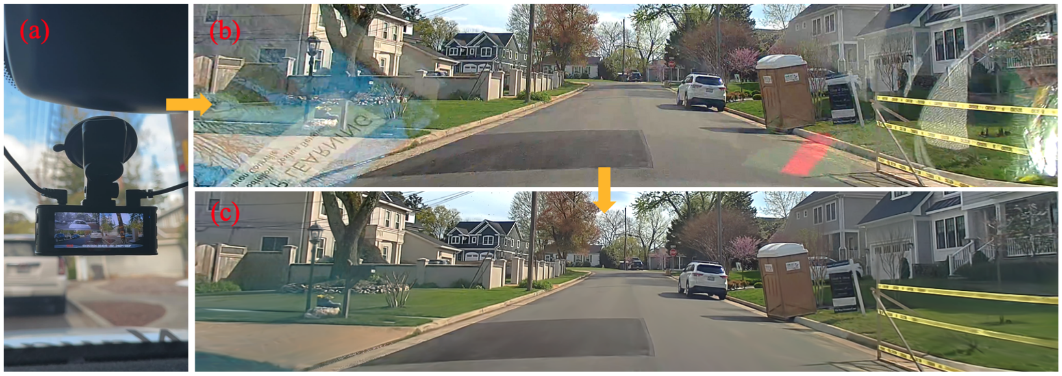

Figure 1: Given a sequence of video captured by a dash cam that may contain obstructions like reflections and occlusions, **DC-Gaussian** achieves high-fidelity novel view synthesis getting rid of the obstructions. (a) dash cam; (b) original video frame; (c) novel view rendering with obstruction removal.

Abstract

We present DC-Gaussian, a new method for generating novel views from in-vehicle dash cam videos. While neural rendering techniques have made significant strides in driving scenarios, existing methods are primarily designed for videos collected by autonomous vehicles. However, these videos are limited in both quantity and diversity compared to dash cam videos, which are more widely used across various types of vehicles and capture a broader range of scenarios. Dash cam videos often suffer from severe obstructions such as reflections and occlusions on the windshields, which significantly impede the application of neural rendering techniques. To address this challenge, we develop DC-Gaussian based on the recent real-time neural rendering technique 3D Gaussian Splatting (3DGS). Our approach includes an adaptive image decomposition module to model reflections and occlusions in a unified manner. Additionally, we introduce illumination-aware obstruction modeling to manage reflections and occlusions under varying lighting conditions. Lastly, we employ a geometry-guided Gaussian enhancement strategy to improve rendering details by incorporating additional geometry priors. Experiments on self-captured and public dash cam videos show that our method

[*] Equal contribution

[†] Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

not only achieves state-of-the-art performance in novel view synthesis, but also accurately reconstructing captured scenes getting rid of obstructions. See the project page for code, data: https://linhanwang.github.io/dcgaussian/.

1 Introduction

Neural Radiance Field (NeRF) [30] has revolutionized the image-based rendering area with its differentiable volume rendering technique. 3D Gaussian Splatting (3DGS) [19] pushes the frontier further with real-time rendering speed. These technologies have been applied to datasets captured by autonomous cars [42, 6, 25], opening up numerous new possibilities in autonomous driving, such as simulating driving scenarios [57, 51] for robust training of perception models and providing effective 3D scene representations to enhance comprehensive environmental understanding [14, 63, 66]. Although these datasets provide multi-modality sensor data, their diversity in real-world driving scenarios is still limited [7].

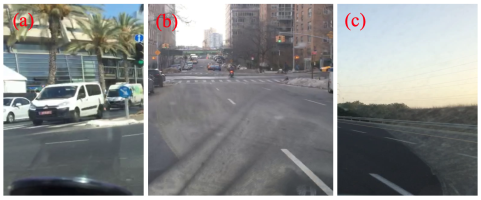

Figure 2: Common obstructions on windshields: (a) Mobile-phone holder; (b) Reflections; (c) Stains.

Fortunately, dash cam videos deeply reflect the diversity and complexity of real-world traffic scenarios [8]. They are used to provide large-scale, diverse driving video datasets in a crowd-sourced manner [59]. Dash cam videos also offer unique value by capturing multi-agent driving behaviors [7] and evaluating the robustness of algorithms under visual hazards [60]. Moreover, the global dash cam market is rapidly expanding, driven by increasing awareness of vehicular safety [34]. Therefore, the exploration of utilizing dash camera data in neural rendering shows great potential, offering enormous amounts of data for autonomous driving applications.

However, naively training 3DGS on dash cam videos often results in a significant deterioration of rendering quality and geometry. This degradation is primarily due to the common existence of obstructions (reflections and occlusions such as mobile phone holders and stains, as shown in Fig. 2) on windshields. In these scenarios, 3DGS models the obstructions as stationary geometries while they are dynamic in nature (moving with cars), thus unavoidably causing inaccurate geometry and blurry renderings in novel views.

Although some single-image-based obstruction removal methods exist, directly applying them to this task is nontrivial. These obstructions arise from various sources, while existing removal methods impose strict assumptions on obstructions [12, 64, 56, 55, 38, 20]. For instance, assumptions like out-of-focus [64] and ghost cue [38] allow previous methods to perform well in specific cases, but these assumptions don't always hold in dash cam videos. Moreover, the performance of learning-based methods degrades for out-of-distribution images [47, 27, 21]. Several NeRF methods [16, 33, 68, 69] attempt to reconstruct scenes with reflections by decomposing transmission (background scene) and reflection with independent NeRFs. This approach can benefit from the strong multi-view aggregation power of NeRF. However, previous methods are insufficient for dash cam videos because the obstructions on windshields do not align well with NeRF's design. The vanilla NeRF is intended for static scenes, whereas windshields and their reflected objects move with the cars.

In this paper, we introduce DC-Gaussian, a method for modeling high-fidelity obstruction-free 3D Gaussian Splatting from dash cam videos. We introduce three key innovations: 1) **Adaptive Image Decomposition**. To clearly decompose images with complex reflections and occlusions, we propose an adaptive image decomposition approach. We use an opacity map to learn the transmittance of the windshield, which adaptively estimates the contribution of the background scene to the image. 2) **Illumination-aware Obstruction Modeling(IOM)**. We observe that the obstructions are mainly caused by the objects being relatively static with the dash camera, but the obstructions present varying effects due to changing illumination. We therefore propose modeling global obstructions that are shared for all views. Moreover, a novel **L**atent **I**ntensity **M**odulation (**LIM**) module is introduced to learn the illumination changes from the scene and enable synthesis of reflection with varying intensity. 3) **Geometry-Guided Gaussian Enhancement(G3E)**. We further leverage multi-view stereo to introduce geometry prior into 3DGS training, which enhances the details and sharpness of 3DGS rendering.

To evaluate the efficacy of our method, we conduct extensive experiments on public datasets [59] and a self-captured dataset. The experiments show that our method not only achieves state-of-the-art novel view synthesis, but also clearly removes obstructions from neural rendering.

2 Related Work

2.1 Novel View Synthesis for Driving Scenes

Novel view synthesis aims to render novel views given posed images of the same scene. NeRF [30], which combines multiple layer perceptions and differentiable volume rendering, initiated a revolution in this area by rendering photorealistic images. Subsequent works [3] [61] extended NeRF to unbounded large-scale scenes by warping the space into a bounded cube. BlockNeRF [44] first introduced NeRF to driving scenes by dividing the scenes into blocks and training separate models on them. The method of scene division was further improved in later works [45, 65]. To apply NeRF to multi-camera systems, UC-NeRF [10] addresses the under-calibration problem by refining the poses with spatio-temporal constraints. S-NeRF [52] uses sparse LiDAR points to enhance the training of NeRF and learn robust geometry. Decomposing dynamic objects and static backgrounds in driving scenes presents another challenge. Some works [46, 54] tackle this challenge with the help of LiDAR and 2D optical flows.

Recently 3DGS [19] has attracted great attention in the research community. It achieves optimal results in novel view synthesis and rendering speed by explicitly modeling a 3D scene with 3D gaussians. Some researchers have extended it to dynamic objects and scenes. Given a set of dynamic monocular images, a deformation network is introduced to model the motion of Gaussians [58]. DrivingGaussian [67] models and decomposes dynamic driving scenes based on composite gaussian splatting. GaussianPro [9] improves the geometry of 3DGS by controlling the densification process of 3DGS with classic PatchMatching algorithm [2]. Zhou et al. [66] propose to utilize 3D Gaussian Splatting for holistic urban scene understanding. However, dash cam videos, an important data source for understanding driving scenes, remain unexplored due to obstructions on the windshield. Despite previous works [67, 54, 58] making progress in separating dynamic objects from background scenes, the image decomposition problem we are addressing presents unique challenges because of the obstructions' transparent or semi-transparent nature.

2.2 Obstruction Removal and Layer Separation

Single-image reflection removal. To address the highly ill-posed problem of single-image reflection removal, various methods leverage different cues. Polarization cues are particularly valuable as they are inherently present in all natural light sources [36, 13, 21, 28]. Gradient priors [23, 24, 1] are utilized based on the observation that reflection and background layers often exhibit different levels of sharpness. Additionally, ghosting cues [38, 64, 18] and flash/non-flash pairs [20, 22] can be effective in certain scenarios. However, these assumptions do not always hold in real-world situations. With the advancement of deep learning technology, learning-based methods [17, 12, 50, 17] have been developed to model reflection properties more comprehensively. Despite their success, reflection removal from a single image remains challenging due to the inherently ill-posed nature of the problem, the absence of motion cues [27], and the difficulties in out-of-domain generalization [47].

Multi-image layer separation. Existing methods often exploit differences in motion patterns between transmission and obstruction layers and use learned image priors [15] to decompose images into multiple components. These layer separation methods estimate dense motion fields for each layer using optical flow [43], SIFT flow [40], and deep learning-based flow estimation methods [41, 27]. Recently, Nam et al. [32] propose a multi-frame fusion framework based on neural image representation, achieving strong performance on various layer-separation tasks. Similarly, NSF [11] fuses RAW burst image sequences by modeling optical flow with neural spline fields. However, methods designed for burst images struggle with large pixel motions in driving scenes.

NeRF with reflections. NeRFRen [16] is the pioneering work that adapts NeRF to model scenes with reflections by proposing to model transmitted and reflected components with separate NeRFs. NeuS-HSR [33] achieves high-fidelity 3D surface reconstruction by explicitly modeling the glasses

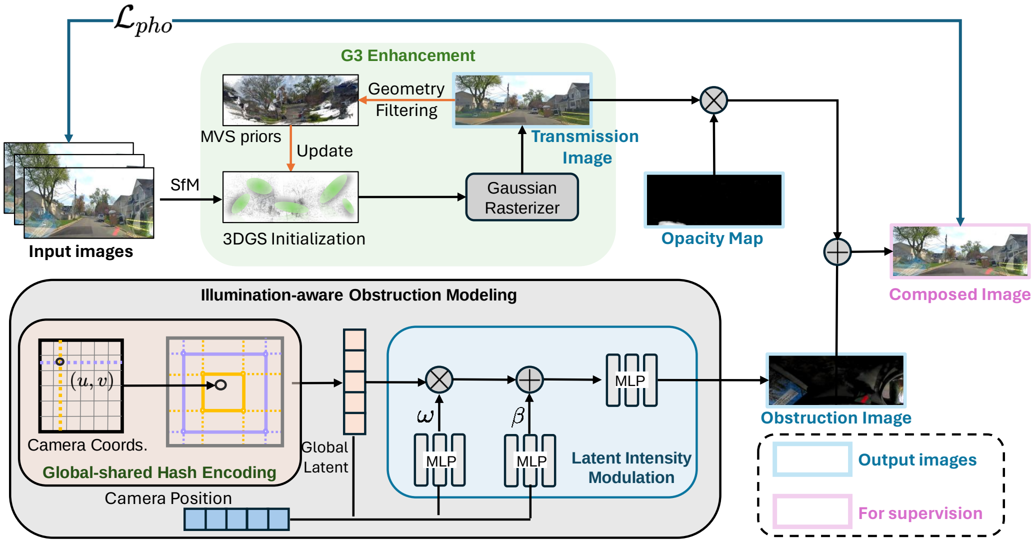

Figure 3: Overview of DC-Gaussian framework. To model obstructions with different opacities in a unified manner, we use an learnable opacity map to adaptively reweight the contribution of transmission. The global-shared multiresolution hash encoding is introduced to fully utilize the static motion prior of obstructions. We propose a Latent Intensity Modulation module to grasp the intensity changes of reflections conditioned on camera positions. Finally, in the G3 Enhancement module, we run geometry filtering on obstruction-suppressed images to enhance the geometry of 3D Gaussians.

with an auxiliary plane module. Zhu et al. [68] introduce recurring edge cues to achieve robust results under sparse views. However, previous methods are insufficient for dash cam videos because the obstructions on windshields do not align well with NeRF's design. The vanilla NeRF is intended for static scenes, whereas windshields and their reflected objects move with the cars. The varying illumination in the wild makes reflections modeling even more challenging. In contrast to existing methods, our proposed obstruction modeling approach leverages the static nature of obstructions in camera coordinates and captures the varying intensity of reflections.

3 Method

Our DC-Gaussian extends the standard 3DGS to corrupted dash cam videos. We begin by reviewing the standard 3DGS pipline 3.1. Then we introduce the Adaptive Image Decomposition 3.3 to decompose the reflections and occlusions from the corrupted dash cam images. In 3.3, we propose the Illuminate-aware Obstruction Modeling module. A novel Latent Intensity Modulation is introduced to enable high quality modeling of reflection even under vary illumination. Finally, the Geometry Guided Gaussian Enhancement strategy is explained in 3.4.

3.1 Preliminary of 3D Gaussian Splatting

3DGS [19] models the 3D scene as a set of 3D Gaussians. Each 3D Gaussian G is defined as:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent its mean vector and covariance matrix, respectively. For optimization purpose, the covariance matrix is further expressed as $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{T}\boldsymbol{S}^T\boldsymbol{R}^T$, where $\boldsymbol{S}$ and $\boldsymbol{R}$ are the scaling matrix and rotation matrix, respectively. To render an image, the splatting technique [70] is applied. Specifically, the color of each pixel $\boldsymbol{p}$ is calculated by blending N ordered Gaussians $\{G_i | i = 1, ..., N\}$ overlapping $\boldsymbol{p}$ as:

$$\boldsymbol{c}(\boldsymbol{p}) = \sum_{i=1}^{N} \boldsymbol{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{2}$$

where α_i is obtained by evaluating a projected 2D Gaussian [70] from G_i at $\boldsymbol{p}$ multiplied with a learned opacity of G_i, and $\boldsymbol{c}_i$ is the learnable color of G_i.

3.2 Adaptive Image Decomposition

To synthesize images with obstructions, previous methods [16, 33] render **transmission image** I_t and **obstruction image** I_o separately and utilize a naive linear combination of I_t and I_o to render the final output I, as illustrated in Eq. 3:

$$I = \phi_1 * I_t + \phi_2 * I_o, \tag{3}$$

where ϕ_1 and ϕ_2 are manually chosen. While this approach achieves descent performance on the pure reflection corrupted images, it cannot achieve good performance on dash cam videos with complex obstructions. Among the common obstructions, mobile-phone holders are opaque, stains is semi-transparent and reflections are transparent. Faced with this complex situation, inspired by [32], we propose to learn the opacity map $\boldsymbol{\phi}$ from the input images. As a result, we reformulate the rendering process:

$$I(u, v, j) = (1 - \boldsymbol{\phi}) I_t(u, v, j) + \boldsymbol{\phi} * I_o(u, v, j) \tag{4}$$

where $u, v \in [0, 1]$ are the continuous image coordinates and $j \in [0, 1, ..., N-1]$ is the frame index. Instead of defining the opacity field in 3D world space, we define the opacity relative to the 2D image space of each view, resulting in a learnable 2D tensor in practice. This approach is more convenient for modeling obstructions because view-dependent effects are only related to the training images. In this image decomposition method, the transmission images represent the driving scenes viewed through the windshield. We can use standard 3DGS to model these scenes due to its multi-view consistent property. Thus, $I_t(u, v, j)$ can be easily calculated by Eq. 2, given the camera pose of frame j. $I_o(u, v, j)$ represents the appearance of the complex obstructions on the windshield. In the implementation, we incorporate ϕ into $I_o(u, v, j)$ in the second term of Eq. 4 for robust training. We explain the modeling of $I_o(u, v, j)$ in the next section.

3.3 Illumination-aware Obstruction Modeling

Decomposing the images into transmission and obstruction is challenging due to the strong ambiguity between the two components. It is an ill-posed problem without prior information. We have two observations about the obstructions on the windshield that can strongly mitigate this ambiguity.

Observation 1 As shown in Fig. 4 (a), the reflections on the windshields are from objects (air conditioner vent) inside the car, which means they are relatively stationary with the car [39]. Additionally, occlusions are attached to the windshields, which are also relatively stationary with the car. Consequently, **we can assume that the appearance of the obstructions is like an image shared globally by all the frames in a dash cam video**.

Observation 2 As cars move along the road, the trees and buildings on the sides occasionally block the incident light, affecting the intensity of the reflections. For example, under strong light, reflections are also strong, as shown in Fig. 4 (a), while the reflections "disappear" under weak light, as seen in Fig. 4 (d). Thus, **the strength of the reflections is conditioned on the car's position**.

These two observations require us to design a model for obstructions that takes advantage of the global-sharing property of obstructions and also grasps varying intensity reflections conditioned on the car's position.

Global-shared Multi-resolution Hash Encoding. To align our design with Observation 1, we use a global-shared latent representation for obstructions' appearance. Specifically, we use continuous image coordinates $u, v \in [0, 1]$ as the input. Then we use a multi-resolution hash encoder γ [31] to map these coordinates into high-dimensional learnable latent features. For example, we use an L

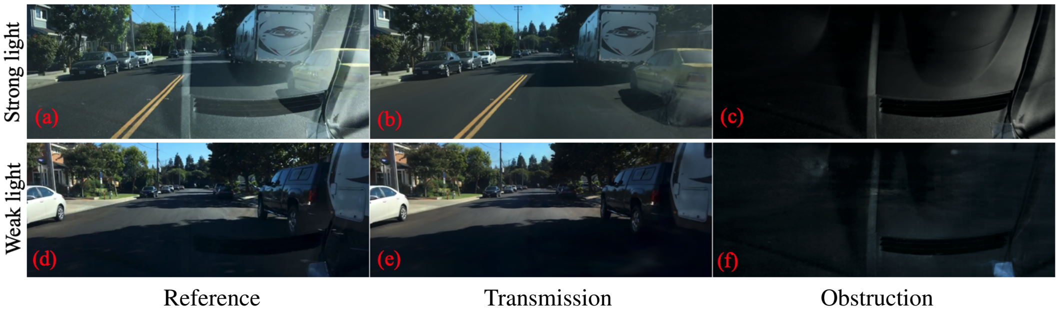

Figure 4: When the intensity of incident light changes, the strength of reflections also changes accordingly (a, d). Our method achieves high-fidelity reflections synthesis (c, f) and reasonable decomposition results (b, e) under varying light. The reflections in (f) are too weak to be seen by the eye, so we brighten it to reveal the details.

level hash encoder where each level stores F dimensional features. Thus, each u, v pair maps to an $L * F$ dimensional latent features $\gamma(u, v)$. While our method is not restricted to this specific type of spatial encoding, we choose the multi-resolution hash encoder for two reasons. First, its hierarchical multi-resolution representation can adaptively learn the obstruction appearance in a coarse-to-fine fashion. Second, its efficient implementation matches the speed of 3DGS, without slowing down the training significantly.

Latent Intensity Modulation. In order to accurately capture the intensity variations in environmental lighting conditioned on car positions, we propose a novel Latent Intensity Modulation (LIM) module. Specifically, to enable selective activation of reflections conditioned on camera positions, we design a *Scaling Gate* ω and an *Offset Gate* β, which are generated by two MLPs with the concatenation of camera positions $\boldsymbol{\pi}_j$ and $\gamma(u, v)$ as input.

$$\omega(u, v, \boldsymbol{\pi}_j) = \mathrm{MLP}([\boldsymbol{\pi}_j, \gamma(u, v)]), \beta(u, v, \boldsymbol{\pi}_j) = \mathrm{MLP}([\boldsymbol{\pi}_j, \gamma(u, v)]), \tag{5}$$

Then, the latent features are modulated element-wise through these two gates. Finally, we use a MLP to decode the modulated latent features into RGB information of obstructions.

$$I_o(u, v, \boldsymbol{\pi}_j) = \mathrm{MLP}(\omega(u, v, \boldsymbol{\pi}_j) \odot \gamma(u, v) + \beta(u, v, \boldsymbol{\pi}_j)) \tag{6}$$

With the LIM module, our method achieves effective image decomposition and synthesizes high-fidelity reflections under varying light conditions, as shown in Fig. 4.

3.4 Geometry-Guided Gaussian Enhancement

In IOM, we utilize the motion pattern prior of obstructions to reduce ambiguity in decomposing images. Despite this, some ambiguity still remains. To further enhance performance, we incorporate geometry priors. Typically, strong obstructions affect only portions of the images. As cars move, the same objects in the 3D world can appear in several image fragments which are not or less interfered with by obstructions. In these image fragments, the texture in the transmission is less blurred by obstructions, and multi-view consistency is maintained. Based on this intuition, we leverage a multi-view stereo (MVS) algorithm to identify these "image fragments" and generate geometry priors for 3D Gaussians.

Specifically, we employ a deep MVS method [48] to generate depth maps for all views. A geometric consistency filtering process [37] is leveraged to generate masks M_j masking the multi-view consistent areas, which are essentially the "image fragments" we want to identify. The masked depth maps are then mapped to 3D space as dense point clouds, which are used to initialize 3D Gaussians at physically accurate positions. To find more multi-view consistent "image fragments," we propose suppressing the obstructions in training images by employing our proposed method. Specifically, we remove obstructions from input images to obtain transmission $\hat{I}_t^j$. For areas blocked by occlusions(where ϕ has large value), we inpaint the content with I_t^j.

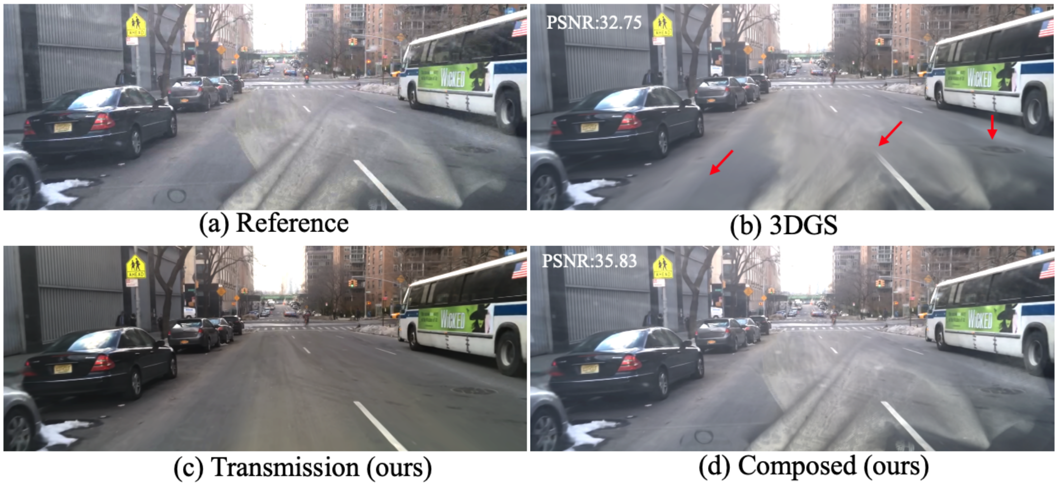

Figure 5: Comparisons with 3DGS on novel view synthesis. Because the obstructions violate multi-view consistency, the performance of 3DGS degrades significantly, resulting in artifacts and blurry renderings (highlighted by red arrows). In contrast, our method not only faithfully synthesizes novel view renderings but also renders transmission with fine details, exhibiting an improvement of 3.05dB in terms of PSNR.

$$\hat{I}_t^j = \begin{cases} (I^j - I_o^j)/(1-\phi), & \text{if } \phi < \tau \\ I_t^j & \text{otherwise} \end{cases} \tag{7}$$

Here I^j is the j_{th} input image. I_o^j and I_t^j are synthesized by trained IOM and 3DGS, respectively. Eq. 7 is derived from Eq. 4. We use 0.5 for the threshold τ in all the experiments.

3.5 Implementation Details

We develop our method based on 3DGS. We borrow multi-resolution hash encoding and fast MLP implementation from tiny-cuda-nn [31] to build IOM. We choose PatchMatchNet [48] as the MVS method in G3E. We follow previous driving scenes reconstruction works [46, 10] to separately model the sky areas. The final loss function is:

$$\mathcal{L} = \mathcal{L}_{pho} + \lambda_1 \mathcal{L}_{sky} + \lambda_2 \mathcal{L}_{opacity} \tag{8}$$

$\mathcal{L}_{pho}$ is the same as in standard 3DGS [19]. $\mathcal{L}_{sky}$ is borrowed from UCNeRF [10]. We use a L_1 loss $\mathcal{L}_{opacity} = \sum_{(u,v)} \|\phi(u,v)\|_1$ to regularize the opacity field, where (u,v) are the image coordinates. This opacity loss encourages the opacity map to have the minimum areas that could satisfy the optimization. This design is based on the prior knowledge that opaque objects typically occupy only small portions of the windshield. We use 0.001 for both λ_1 and λ_2. We run all the experiments with an A100 GPU. Each scene contains approximately 300 images in our evaluation datasets. Combined training of the 3DGS and IOM for 30k iterations with Adam optimizer takes about 30 minutes. It takes 40 minutes in total to evaluate MVS and run geometry filtering.

4 Experiments

4.1 Datasets

BDD100K. To evaluate the performance of our method and baselines, we adopt BDD100K [59] for evaluation. This dataset contains 8 scenes that are from dash cam videos captured in daily life. They contain common obstructions, such as reflections, mobile phone holders, stickers and stains. Evaluation on this dataset reflects performance on real life dash cam videos.

Table 1: Evaluation of novel view synthesis on BDD100K and DCVR. We indicate the best and second best with bold and underlined respectively. Our method consistently outperforms state-of-the-art methods in both datasets and all the evaluation metrics.

	BDD100K			DCVR			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑
NeRF-W [29]	22.58	0.708	0.395	-	-	-	0.18
ZipNeRF [4]	27.89	0.875	0.176	24.41	0.786	0.228	0.27
GaussianPro [9]	27.75	0.894	0.192	23.71	0.770	0.270	**210**
3DGS [19]	28.02	0.897	0.188	23.73	0.783	0.248	155
DCGaussian (Ours)	**29.44**	**0.914**	**0.143**	**24.74**	**0.822**	**0.202**	120

DCVR. To further evaluate the performance of our method on strong reflection conditions, we established **DCVR** (Dash Cam Videos with Reflection) dataset. This dataset contains 10 dash cam videos we collect. The original videos are undistorted [5] to ease the structure-from-motion algorithm. We utilize the popular tools COLMAP [37] and HLoc [35, 26] to estimate the camera parameters.

In both datasets, each sequence consists of approximately 300 frames, extracted from 10-second videos at a frame rate of 30 Hz. For scenes containing car hoods, the lower parts of the images are removed during preprocessing. Seven out of eighteen scenes in two datasets involve car turns, introducing diverse illumination changes. Additional details and visual results are provided in the appendix.

4.2 Baselines

We choose 3DGS as our baseline [19]. We also compare our method with other state-of-the-art methods Zip-NeRF [4], NeRF-W [29] and GaussianPro [9]. We use the unofficial implementations of Zip-NeRF [3] and NeRF-W [4]. To evaluate the performance of novel view synthesis, following common settings [3], we select one of every eight images as testing images and the remaining ones for training. Since our method is also designed for image decomposition, we also compare our method with state-of-the-art obstruction removal methods, including DSRNet [17], NIR [32] and Liu et al. [27].

4.3 Quantitative Results

For the quantitative evaluation, we conduct comparison with baselines on both BDD100K and DCVR. We apply the three widely-used metrics for evaluation, i.e., PSNR, SSIM [49], and LPIPS [62]. The results are shown in Tab. 1. The consistent superior performance of our method shows the efficacy of the proposed modules. Though GaussianPro and Zip-NeRF achieve great performance in obstruction-free scenes with their progressive propagation strategy and anti-aliasing mechanism, without separate obstruction modeling, they cannot handle the obstructions corrupted dash cam videos. NeRF-W is designed to handle illumination and content difference between images taken at different times but still cannot handle obstructions on the windshield. We show more visual results in the appendix. DC-Gaussian achieves 120 fps at a resolution of 1920x1080 on an RTX 3090 GPU. Although ours is slightly slower than 3DGS, the speed still enables real-time rendering, which is crucial for applications such as autonomous driving simulators.

4.4 Qualitative Results

Novel view synthesis. We show the novel view synthesis results in Fig. 5. Without a proper representation for obstructions, 3DGS can hardly synthesize high-quality obstructions. Moreover, its wrong geometry also results in blurry renderings and artifacts on the road surface. In contrast, our method effectively tackles the ambiguity between obstructions and transmissions and synthesizes both components with high-fidelity. We provide depth map in the appendix.

Obstruction removal. We show the obstruction removal results in Fig. 6. The single image reflection removal method (e) only marginally suppresses reflections. Multi-image layer separation methods

[3] https://github.com/SuLvXiangXin/zipnerf-pytorch

[4] https://github.com/kwea123/nerf_pl

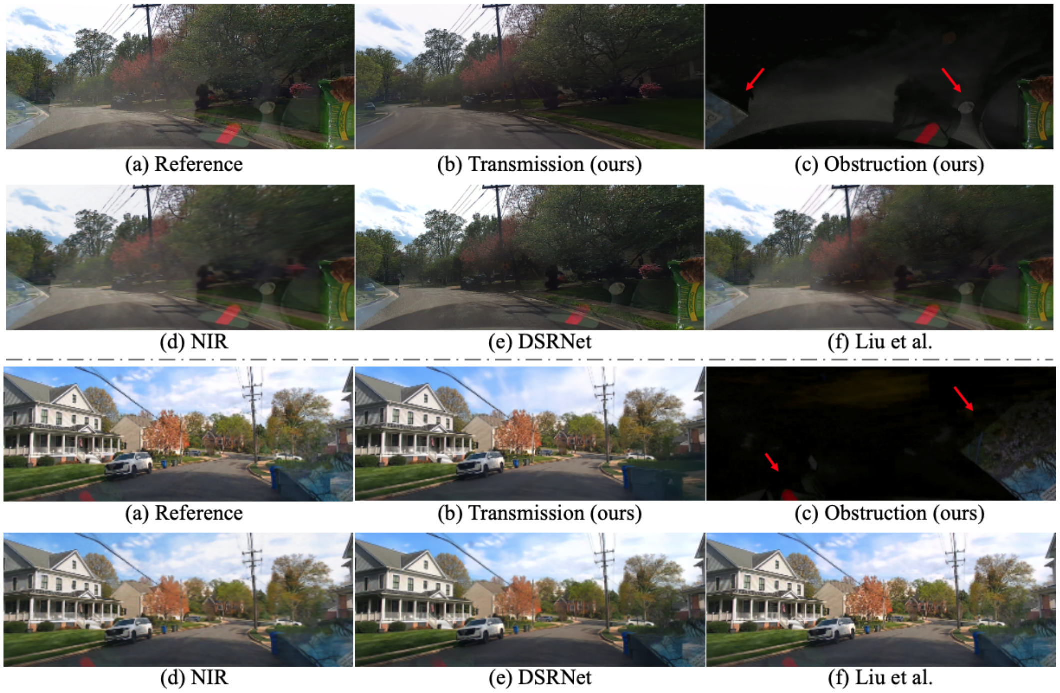

Figure 6: Comparisons with the image-based reflection removal methods on removing reflections. (a) Reference is a frame in a dash cam video. (b) and (c) are transmission and obstruction decomposed by our method. (d), (e), and (f) are results from previous obstruction removal methods, NIR [32], DSRNet [17] and Liu et al. [27], which are not effective in this scenario. In comparison, our method decomposes the image and synthesizes (b) transmission and (c) obstruction with high fidelity.

Table 2: Ablations studies on DCVR. Metrics are calculated on obstruction influenced areas.

NOM	AD	LIM	G3E	PSNR ↑	SSIM ↑	LPIPS ↓
✗	✗	✗	✗	23.99	0.738	0.287
✓	✗	✗	✗	25.21	0.776	0.252
✓	✓	✗	✗	25.65	0.791	0.236
✓	✓	✓	✗	25.90	0.798	0.229
✓	✓	✓	✓	**26.30**	**0.814**	**0.210**

(d)(f) struggle with accurate optical flow estimation, resulting in blurry outputs. In comparison, our method models reflections (c) with high quality and retains fine details in transmission (b). These visual results demonstrate our method's potential in reconstructing obstruction-free driving scenes from dash cam videos.

4.5 Ablation Study

We conduct extensive ablation studies on DCVR to explore the impact of each module in DC-Gaussian. Quantitative results are shown in Table 2. To assess the efficacy of global-shared hash encoding, we evaluate 3DGS with a Naive Obstruction Module (NOM), where latent features are directly decoded by an MLP to generate RGB. NOM shows significant improvement over the baseline, demonstrating that global-shared hash encoding effectively leverages the static prior of obstructions in dash cam videos. The adaptive image decomposition (AD) strategy further enhances results by effectively modeling occlusions, as shown in Fig. 7. Additionally, LIM improves performance by capturing intensity changes in reflections. Finally, incorporating geometry priors into 3D Gaussians with G3E effectively suppresses artifacts on the road and reveals sharper details, as illustrated in Fig. 8.

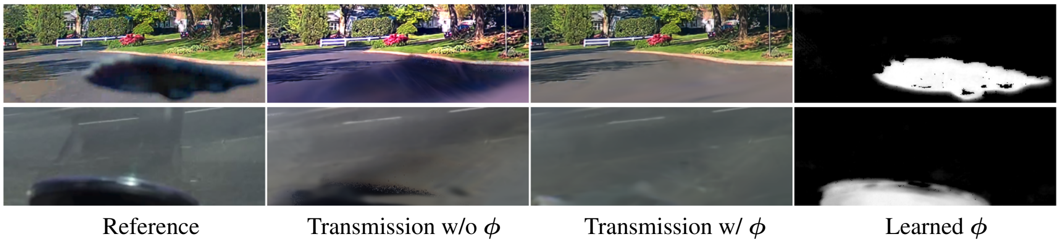

Figure 7: Ablation study about the Learnable opacity map ϕ. Incorporating the opacity map allows our method to accurately identify the positions of opaque objects, enhancing physical simulation and improving view synthesis and obstruction removal. Without the opacity map, severe artifacts appear.

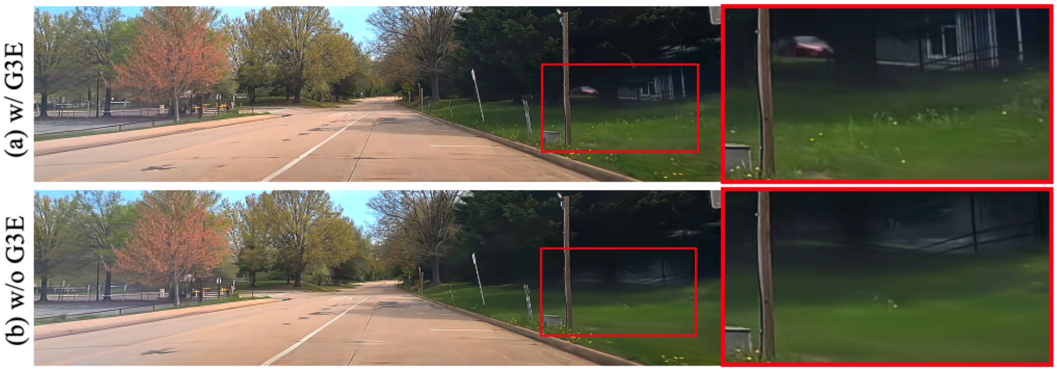

Figure 8: Ablation study on G3E module. G3E helps suppress artifacts and reveal sharper details.

5 Conclusions

In conclusion, we propose DC-Gaussian, which effectively addresses the challenges of extending 3D Gaussian Splatting to dash cam videos for the first time. The proposed Adaptive Image Decomposition module enables unified modeling of reflections and occlusions. To handle the reflections and occlusions under challenging lighting conditions, we introduce Illumination-aware Obstruction modeling. Additionally, we employ a Geometry-Guided Gaussian Enhancement strategy to further improve rendering quality. Experiments on BDD100K and DCVR demonstrate significant improvements in rendering quality and image decomposition, setting a new benchmark for neural rendering with dash cam videos.

Limitations and future work Currently, DC-Gaussian has only been evaluated on single-sequence videos. However, considering the vast amount of dash cam footage available, extending DC-Gaussian to a multi-sequence video setting and leveraging dense view images to achieve more pleasing results would be a promising direction for future research. In addition, Our method is not specifically designed to improve performance on dynamic scenes. We provide additional experimental results in the appendix. The results demonstrate that dynamic objects do not significantly impact the performance of obstruction removal. When dynamic objects move at a slow speed, our method also presents reasonable results. We plan to incorporate techniques [53] for dynamic objects modeling into our method in future research to enable robust dynamic modeling.

Acknowledgments and Disclosure of Funding

The authors acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: https://arc.vt.edu/

References

[1] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4498–4506, 2017.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[7] Rohan Chandra, Xijun Wang, Mridul Mahajan, Rahul Kala, Rishitha Palugulla, Chandrababu Naidu, Alok Jain, and Dinesh Manocha. Meteor: A dense, heterogeneous, and unstructured traffic dataset with rare behaviors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9169–9175. IEEE, 2023.

[8] Zhengping Che, Guangyu Li, Tracy Li, Bo Jiang, Xuefeng Shi, Xinsheng Zhang, Ying Lu, Guobin Wu, Yan Liu, and Jieping Ye. D^2-city: a large-scale dashcam video dataset of diverse traffic scenarios. *arXiv preprint arXiv:1904.01975*, 2019.

[9] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. *arXiv preprint arXiv:2402.14650*, 2024.

[10] Kai Cheng, Xiaoxiao Long, Wei Yin, Jin Wang, Zhiqiang Wu, Yuexin Ma, Kaixuan Wang, Xiaozhi Chen, and Xuejin Chen. Uc-nerf: Neural radiance field for under-calibrated multi-view cameras. In *The Twelfth International Conference on Learning Representations*, 2023.

[11] Ilya Chugunov, David Shustin, Ruyu Yan, Chenyang Lei, and Felix Heide. Neural spline fields for burst image fusion and layer separation. *arXiv preprint arXiv:2312.14235*, 2023.

[12] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3238–3247, 2017.

[13] Hany Farid and Edward H Adelson. Separating reflections and lighting using independent components analysis. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 262–267. IEEE, 1999.

[14] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022.

[15] Yosef Gandelsman, Assaf Shocher, and Michal Irani. " double-dip": unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11026–11035, 2019.

[16] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

[17] Qiming Hu and Xiaojie Guo. Single image reflection separation via component synergy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13138–13147, 2023.

[18] Yan Huang, Yuhui Quan, Yong Xu, Ruotao Xu, and Hui Ji. Removing reflection from a single image with ghosting effect. *IEEE Transactions on Computational Imaging*, 6:34–45, 2019.

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

[20] Chenyang Lei and Qifeng Chen. Robust reflection removal with reflection-free flash-only cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14811–14820, 2021.

[21] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1750–1758, 2020.

[22] Chenyang Lei, Xudong Jiang, and Qifeng Chen. Robust reflection removal with flash-only cues in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[23] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647–1654, 2007.

[24] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2752–2759, 2014.

[25] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

[26] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021.

[27] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14215–14224, 2020.

[28] Youwei Lyu, Zhaopeng Cui, Si Li, Marc Pollefeys, and Boxin Shi. Reflection separation using a pair of unpolarized and polarized images. *Advances in neural information processing systems*, 32, 2019.

[29] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021.

[30] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[32] Seonghyeon Nam, Marcus A Brubaker, and Michael S Brown. Neural image representations for multi-image fusion and layer separation. In *European conference on computer vision*, pages 216–232. Springer, 2022.

[33] Jiaxiong Qiu, Peng-Tao Jiang, Yifan Zhu, Ze-Xin Yin, Ming-Ming Cheng, and Bo Ren. Looking through the glass: Neural surface reconstruction against high specular reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20823–20833, 2023.

[34] Grand View Research. Dashboard camera market size, share & trends analysis report by technology (basic, advanced, smart), by product, by video quality, by application, by distribution channel, by region, and segment forecasts, 2024 - 2030, 2023. Accessed: 2024-05-16.

[35] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[36] Yoav Y Schechner, Joseph Shamir, and Nahum Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 814–819. IEEE, 1999.

[37] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.

[38] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3193–3201, 2015.

[39] Christian Simon and In Kyu Park. Reflection removal for in-vehicle black box videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4231–4239, 2015.

[40] Chao Sun, Shuaicheng Liu, Taotao Yang, Bing Zeng, Zhengning Wang, and Guanghui Liu. Automatic reflection removal using gradient intensity and motion cues. *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2017.

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[43] Richard Szeliski, Shai Avidan, and Padmanabhan Anandan. Layer extraction from multiple images containing reflections and transparency. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 246–253. IEEE, 2000.

[44] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[45] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022.

[46] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023.

[47] Renjie Wan, Boxin Shi, Haoliang Li, Yuchen Hong, Ling-Yu Duan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1424–1441, 2022.

[48] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14203, 2021.

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[50] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019.

[51] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023.

[52] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023.

[53] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. *arXiv preprint arXiv:2401.01339*, 2024.

[54] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.

[55] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *Proceedings of the european conference on computer vision (ECCV)*, pages 654–669, 2018.

[56] Yang Yang, Wenye Ma, Yin Zheng, Jian-Feng Cai, and Weiyu Xu. Fast single image reflection suppression via convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8141–8149, 2019.

[57] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023.

[58] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

[59] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[60] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–416, 2018.

[61] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[62] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[63] Xiaoshuai Zhang, Abhijit Kundu, Thomas Funkhouser, Leonidas Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8274–8284, 2023.

[64] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4786–4794, 2018.

[65] MI Zhenxing and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *The Eleventh International Conference on Learning Representations*, 2022.

[66] Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. *arXiv preprint arXiv:2403.12722*, 2024.

[67] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*, 2023.

[68] Chengxuan Zhu, Renjie Wan, and Boxin Shi. Neural transmitted radiance fields. *Advances in Neural Information Processing Systems*, 35:38994–39006, 2022.

[69] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20722–20731, 2023.

[70] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.

A Appendix / supplemental material

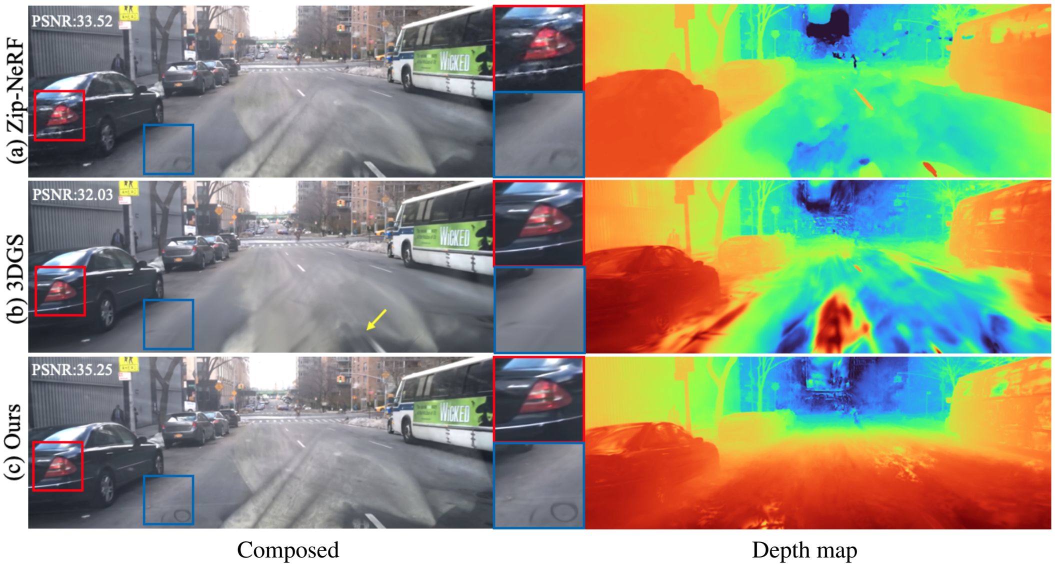

Figure 9: Comparisons with 3DGS and Zip-NeRF [4] on novel view synthesis show that obstructions violate multi-view consistency, leading to erroneous geometry in 3DGS and Zip-NeRF, as evident in the depth maps. This results in blurry renderings and artifacts. In contrast, our method effectively addresses the ambiguity introduced by obstructions and learns physically reasonable geometry, achieving renderings with fine details.

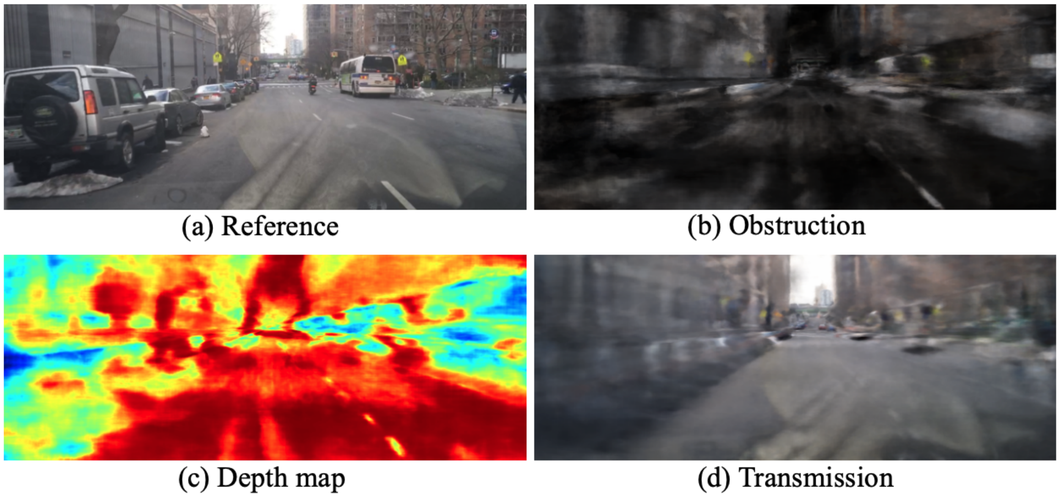
(a) Reference
(b) Obstruction
(c) Depth map
(d) Transmission

Figure 10: We evaluate NeRFRen [16] on our curated dataset. The suboptimal results of NeRFRen are caused by two factors. First, its obstruction modeling cannot address the ambiguity between obstructions and transmission, leading to a failure in image decomposition. Second, its backbone, NeRF, cannot handle large-scale driving scenes, resulting in blurry outputs.

Table 3: Ablation on threshold τ used in Eq. 7. Our results are not sensitive to the choice of τ.

τ	PSNR ↑	SSIM ↑	LPIPS ↓
0.3	26.27	0.814	0.210
0.5	26.30	0.814	0.210
0.7	26.28	0.814	0.210

Table 4: We first use DSRNet [17] to remove reflections from the input images, and then we train and evaluate 3DGS [19] on these images. The results show that due to the insufficiency of the reflection removal, novel view synthesis performance cannot be improved in this way.

Method	BDD100K			DCVR		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	28.02	0.897	0.188	23.73	0.783	0.248
3DGS + DSRNet	27.99	0.898	0.188	23.72	0.783	0.248

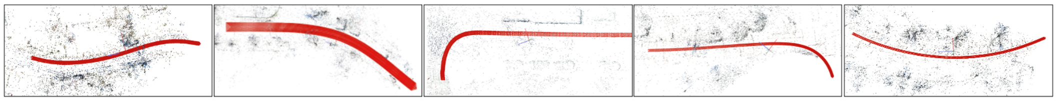

Figure 11: Trajectories of turning cars, which result in diverse illumination changes.

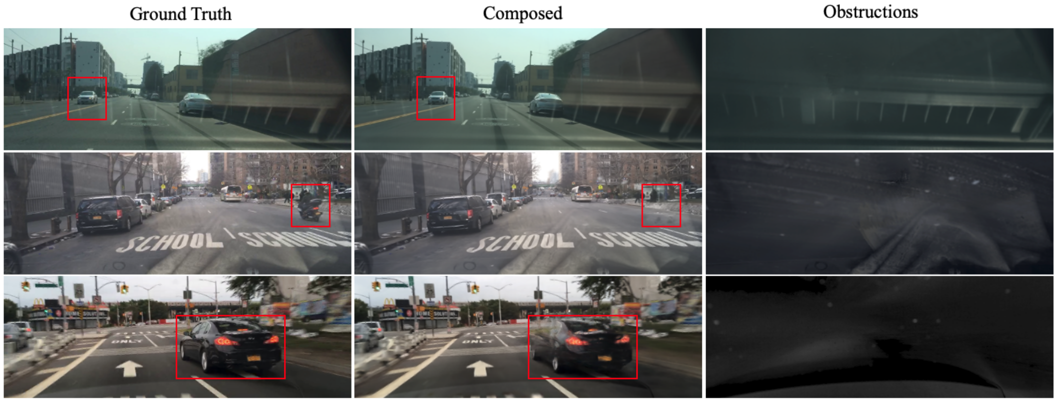

Figure 12: Visual results on dynamic scenes. All three scenes demonstrate that dynamic objects do not significantly impact the decomposition of obstructions. Our method achieves good performance in the scene shown in the first row, where the dynamic objects are moving slowly. However, in the second and third rows, where the dynamic objects are moving at higher speeds, our method shows suboptimal performance.

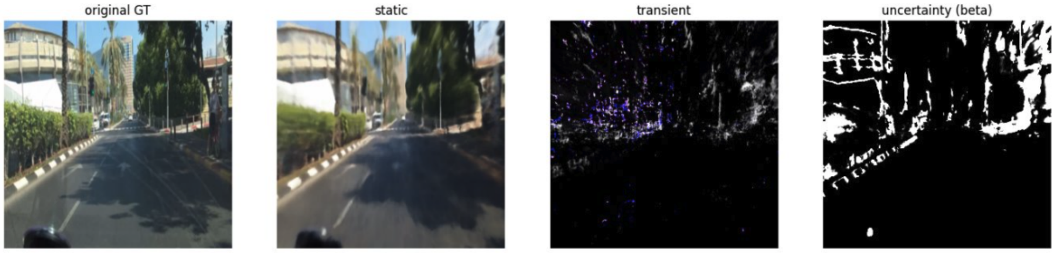

Figure 13: Nerf-in-the-wild fails to separate obstructions from the images. None of the obstructions are accurately represented in the transient image.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We show our method's performance on novel view synthesis and obstruction removal, reflecting our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this paper in conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not present any theoretical results requiring assumptions or proofs, making this criterion not applicable.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We cite used techniques and introduce our method in details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will submit our code and data to github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We document all the training details in the implementation details section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: It's too compute intensive to do so.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the GPU we used in implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We understand and respect the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We discuss the positive societal impacts; however, we do not foresee any potential negative impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, the creators or original owners of the assets used in the paper are properly credited, and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide information about the dataset we use.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work doesn't involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work doesn't involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.