Advancing Adversarial Robustness in GNeRFs: The IL2-NeRF Attack

Anonymous CVPR submission

Paper ID 17334

Abstract

Generalizable Neural Radiance Fields (GNeRF) are recognized as one of the most promising techniques for novel view synthesis and 3D model generation in real-world applications. However, like other generative models in computer vision, ensuring their adversarial robustness against various threat models is essential for practical use. The pioneering work in this area, NeRFool, introduced a state-of-the-art attack that targets GNeRFs by manipulating source views before feature extraction, successfully disrupting the color and density results of the constructed views. Building on this foundation, we propose IL2-NeRF (Iterative L_2 NeRF Attack), a novel adversarial attack method that explores a new threat model (in the L_2 domain) for attacking GNeRFs. We evaluated IL2-NeRF against two standard GNeRF models across three benchmark datasets, demonstrating similar performance compared to NeRFool, based on the same evaluation metrics proposed by NeRFool. Our results establish IL2-NeRF as the first adversarial method for GNeRFs under the L_2 norm. We establish a foundational L_2 threat model for future research, enabling direct performance comparisons while introducing a smoother, image-wide perturbation approach in Adversarial 3D Reconstruction.

1. Introduction

In recent years, machine learning has significantly advanced the field of computer vision, with numerous state-of-the-art (SOTA) models pushing the boundaries of 2D and 3D representation. Among these models, **Ne**ural **R**adiance **F**ields (NeRF) has emerged as a powerful method for reconstructing highly detailed 3D scenes from 2D images [9, 21, 33]. NeRF utilizes a deep learning model to represent a 3D scene as a continuous volumetric field, generating realistic views from any camera angle based on its learned representation of light and color radiance.

As machine learning vision models see greater in-field deployment, security concerns around these models increase. Specifically, adversarial attacks can perturb images, in turn producing adversarial examples that machine learning models misclassify. The first proposed successful attack for GNeRF models was the NeRFool attack under the L_∞ norm [8].

Being the first adversarial attack proposed on GNeRF models, NeRFool was the advent for analyzing robustness for a novel type of vision model. With this in mind, we are keen to present new attacks across new threat norms. Our main contributions can be summarized as follows:

1. We introduce IL2-NeRF, the **first adversarial attack on GNeRF models in the L_2 domain**, providing a novel threat model that applies uniform perturbations across the entire image and setting a foundation for future L_2-based attacks.
2. **Technical Contribution:** We utilize several unique factors, such as an independent perturbation variable and a weighted loss term, that presents a unique perspective on iterative attack algorithms. The traits we present here provide a framework for future algorithms to exploit GNeRF robustness.
3. **Comprehensive Experimental Validation:** We demonstrated the effectiveness of IL2-NeRF through rigorous experiments across multiple datasets (*LLFF*, *DeepVoxels*, *Synthetic*) and GNeRF models (*IBRNet*, *GNT*). The results showcase IL2-NeRF's comparable performance with existing methods like NeRFool in degrading 3D model outputs under varied conditions.

2. Preliminaries

In this section, we will formalize the loss objective of the NeRF and GNeRF pipelines. This allows us to define the NeRFool attack. Lastly, we propose the threat model we operate under. All of this serves as a preliminary to establish our new attack algorithm.

2.1. NeRF Pipeline

In computer vision, the 3D coordinate of the camera is stored in the form of the location (x, y, z) and direction (θ, ϕ). The rays are formed and broken into $r_o \in \mathbb{R}^3$ (ray origin/camera center) and $r_d \in \mathbb{R}^3$ (ray direction) based on the image size.

Each chunk's color and density can later be synthesized into the color along the ray. ie, the ray segment becomes $r_t = r_o + t_i r_d$, where $t_i \sim \mathcal{U}[t_n + \frac{i}{N}(t_f - t_n), t_n + \frac{i+1}{N}(t_f - t_n)]$ [21].

A rendering model is trained using a loss function comprised of multiple functions aggregated together to generate our final scene. The predicted color is a function of the camera's ray $r(t)$ that is input into volume density function σ. The function $T(t)$ denotes the likelihood that the ray will be transmitted from t_n to t without colliding with another rendered particle. Like the ray, the transmittance T is broken down into N evenly spaced bins partitioned from $[t_n, t_f]$. The function then aggregates these partitioned bins back into the full transmittance. The function then uses the sum to estimate the continuous integral $C(r)$.

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt, \tag{1}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right)$.

The estimated color takes in the continuous integral while using δ_i, the distance between consecutive samples along the ray, in the exponential term exp($-\sigma_i\delta_i$) to model then calculate the attenuation of the ray as it travels, which can be given by

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \left(1 - \exp(-\sigma_i \delta_i)\right) c_i \tag{2}$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$.

After the volumetric rendering, we receive the learned image from that particular coordinate. We then use the generated 2D image against the ground truth image to calculate the Mean Squared Error (MSE) Loss from which the model learns and updates its weights. formally, the loss function is as follows

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \left[\|\hat{C}_c(r) - C(r)\|_2^2 + \|\hat{C}_f(r) - C(r)\|_2^2 \right] \tag{3}$$

for all the accumulation of rays as $\mathcal{R}$ where $C_c(r)$ is the output from the coarsely ray-sampled NeRF model and $C_f(r)$ is for the finely ray-sampled NeRF model.

2.2. GNeRF Adaptations

To enable cross-scene generalizations, subsequent work adopt CNN encoders to extract features $\{E(\mathbf{I_i})\}$ from the source views $\{\mathbf{I_i}\}$ [3, 17, 27, 31, 32]. Each sampled point x is transformed by some function π_i, producing feature vectors $e = \{E(\mathbf{I}_i)[\pi_i(x)]\}$ that the GNeRF model f can use to produce the density and color $f(x, r_d, e) = (\sigma, c)$.

2.3. NeRFool

The original NeRF paper presents two attacks - one that samples a subset of rays, and one that perturbs all rays. We will only explore the first NeRFool attack known as the "View Specific" attack [8].

The view-specific attack adds perturbations $\Delta = \{\delta_i\}$ to the RGB pixels of selected source view images ($\mathbf{I}_i$). To minimize perturbations while maximizing loss, Δ is optimized with a perturbation budget of ϵ over a L_∞ norm constraint ($\|\delta_i\|_\infty \le \epsilon$). This gives us the attack objective

$$\max_{\forall \delta_i \in \Delta : \|\delta_i\|_\infty \le \epsilon} \hat{\mathcal{L}}_{rgb}(\mathcal{R}_{target}, f, \Delta) \tag{4}$$

where f is the GNeRF model and R_{target} are the randomly sampled rays for the source view image. $\hat{\mathcal{L}}_{rgb}$ is a modified loss function that uses a pseud-ground truth since the ground truth image might not be available. Formally, it is expressed as follows,

$$\hat{\mathcal{L}}_{rgb}(\mathcal{R}, f, \Delta) = \sum_{r \in \mathcal{R}} \|\hat{C}(r, f_\Delta^{adv}) - \hat{C}(r, f^{clean})\|_2^2 \tag{5}$$

where f_Δ^{adv} is the output from the model with adversarial image and f^{clean} is the output from the model with clean image. Lastly, δ_i is iteratively optimizing by updating at each step via gradient descent. An Adam Optimizer is used as follows:

$$\delta_i^{(t+1)} = \text{clip}(\delta_i^t + \eta \cdot \text{Adam}(\nabla_{\delta_i^t} \hat{\mathcal{L}}_{rgb}), -\epsilon, \epsilon) \tag{6}$$

while keeping the sum between our source image and perturbation $\mathbf{I}_i + \delta_i \in [0, 1]$ within an expected pixel bound.

2.4. Threat Model

All adversarial attacks can be categorized as black-box or white-box attacks [19]. White-box attacks assume the attackers have full access to the target victim model, such as model parameters or the training data set. In contrast, black-box attacks are implemented without prior knowledge of the target victim model, except for the model output [15].

While most existing GNeRF attacks operate under the L_∞ norm, focusing on individual pixel perturbations, the L_2 norm offers a more uniform and holistic approach to perturbations across the entire image, which can be more realistic and challenging for generative models to counter [1, 4, 5, 18, 23]. By introducing the L_2 norm in this setting, we provide a new threat model that broadens the scope for evaluating adversarial robustness in GNeRF. Both types of attacks work by applying perturbations under a specified norm. Most popular attacks work under the L_∞ norm, where updates are performed coordinate-wise and bounded by a specified ϵ term [6, 7, 10, 11, 14, 16]. However, attacks that work exclusively in the L_2 norm, or Euclidean distance, have gained notoriety, especially in the black-box domain [22, 24, 26, 29, 35].

The NeRFool attack requires the model loss and the rotation and translation matrix, making it a white-box attack. We work under the same assumption that we have access to rotation and translation matrices and can compute model loss. Unlike NerFool, which works under the L_∞ norm, we introduce an iterative algorithm that works under the L_2 norm. The following sections explain our threat model's formalism and efficiency.

3. IL2-NeRF: NeRFool in L2 Domain

In this section, we present IL2-NeRF, an iterative attack that perturbs NeRF source images in Euclidean distance. We draw insight from both the NeRFool and PGD attacks.

3.1. Motivation

Current adversarial approaches to GNeRF only focus on the L_∞ norm, which targets maximal per-pixel perturbations [12, 13]. While effective, this approach tends to introduce highly localized perturbations, emphasizing individual pixel changes without a unified impact across the image [1, 4]. In contrast, the L_2 norm threat model offers a different perspective, allowing for smoother, more evenly distributed perturbations across all pixels [4, 23]. This uniformity in perturbation can be particularly advantageous for generative models, where coherent transformations across the image may be more perceptually relevant than isolated pixel deviations [5, 18, 23].

From this, we derive that L_2 perturbations can provide a realistic setting for GNeRF adversarial attacks, as they resemble noise patterns that affect the entire image uniformly, mimicking real-world imaging artifacts. Exploring the L_2 norm thus enables us to assess GNeRF models' robustness in an alternative threat model that may reflect practical adversarial scenarios more accurately than L_∞. This work introduces the first attack under the L_2 norm for GNeRF models, establishing a baseline for future explorations in this domain and enabling direct comparisons in the L_2 threat space.

3.2. Objective

We work under a similar objective as the original NeRFool attack, but we rework the formalism to include minimizing adversarial perturbation as an objective. For an image-set $\{x_i\}_{i=1}^N$, we want to find $\{\delta_i\}_{i=1}^N$ such that $\forall \delta_i \in \Delta$,

$$\min_{\|\delta_i\|_2} \left(\max \hat{\mathcal{L}}_{rgb}(\mathcal{R}_{target}, f, \Delta) \right) \text{ s.t. } \|\delta_i\|_2 \leq \epsilon \qquad (7)$$

where f is the GNeRF model, R_{target} are the randomly sampled rays for the source view image, and $\hat{\mathcal{L}}_{rgb}$ is a modified loss function defined in Eq. (5).

3.3. Algorithm & Explanation

Algorithm 1 IL2-NeRF

Input: Set of sampled rays $\mathcal{R}_{tar}$, GNeRF model f, Input images $\{x\}_{i=1}^N$, Number of Steps T, Step Size α, Perturbation Limit ϵ

Output: δ_i^T for each x_i

for $i \leftarrow 1$ to N **do**
  $\delta_0^t \in \mathcal{U}[-\epsilon, \epsilon]$
  for $t \leftarrow 1$ to T **do**
    $\delta_i'^t = \nabla_{\delta_{i-1}^t} \tilde{\mathcal{L}}_{rgb}(\mathcal{R}_{rgb}, f, \delta_{i-1}^t)$
    $\tilde{\delta_i^t} = \delta_{i-1}^t + \alpha \cdot \text{sign}\left(\frac{\delta_i'^t}{||\delta_i'^t||_2}\right)$
    $\delta_i^t = \delta_{i-1}^t + ||\tilde{\delta_i^t} - \delta_{i-1}^t||_2$
    $\delta_i^t = \text{clip}(\delta_i^t, -\epsilon, \epsilon)$
  end for
end for
Return $\{\delta_i^T\}_{i=1}^N$;

The IL2-NeRF algorithm is an iterative adversarial attack method specifically designed for GNeRF models. Unlike traditional 2D adversarial attack methods, our method targets the volumetric representation of scenes within GNeRF, aiming to produce perturbations along rays that impact the rendered output views. Our algorithm uses gradient-based optimization, customizing the loss function to target the RGB and density of the final output. Specifically, in our implementation, we are taking the gradient over 8 different loss functions, each crafted upon GNeRF's unique 3D perspective.

The pseudo-code of our algorithm is provided in Algorithm (1). We iterate through each source image and sample the initial perturbation from a Uniform distribution with upper and lower bounds set at our perturbation limit.

At the i-th step, we take the gradient of our GNeRF model f when we add the last δ_{i-1} perturbation to our source images and subset of rays, denoting this change in loss as δ'_{i-1}. From here, we add the product of the sign of L_2 normalized δ_{i-1} grad with a specified step size α to δ_{i-1}. We denote this as $\tilde{\delta_i}$. Note that Eq. (6) uses η to denote its learning rate. α is a normalized learning rate determined by $\alpha = \frac{\eta}{255}$.

To ensure that our final perturbation is added with respect to the L_2 norm, we take the difference between $\tilde{\delta_i}$ and δ_{i-1}, normalize this in L_2, and add this back δ_{i-1} to receive our current δ_i term. To ensure our final perturbation stays within our perturbation limit ϵ, we clip in the ϵ ball.

3.4. Key Technical Differences from PGD

1. **Ray-Based Perturbation vs. Pixel-Level Perturbation**: Traditional PGD focuses on perturbing individual pixels in 2D images only[19]. In contrast, IL2-NeRF ap-

plies perturbations along both the *sampled rays in 3D space ($\mathcal{R}_{tar}$)* and at a pixel-level, leveraging the GNeRF model's structure and targeting the volumetric rendering process. This combination of perturbing pixels and rays fundamentally changes how adversarial effects are achieved, as the perturbations influence the entire 3D scene rather than a single 2D representation.

2. **Customized Gradient-Based Optimization for RGB and Density Losses**: Our algorithm uses gradient-based optimization, customizing the loss function to target both the RGB values and the density of the final output. Specifically, in our implementation, we compute the gradient over a weighted loss of 8 different loss functions, each designed to exploit GNeRF's unique 3D perspective. In contrast, traditional PGD typically uses only a single basic loss function, such as Cross Entropy, to guide perturbations in 2D adversarial attacks [2, 19]. These tailored loss functions in IL2-NeRF enable precise control over how the adversarial perturbations affect the rendered output in terms of both color and depth.
3. **Direct Perturbation Optimization**: Most adversarial attacks operate by creating an initial adversarial image and then directly optimizing the loss and perturbation of the adversarial image with respect to its clean counterpart [6, 7, 10, 11, 14, 16]. Our algorithm is unique in that we maintain our perturbation as a separate variable and update this using the gradient loss that our perturbed rays and images create.

4. Experiments & Results

To properly compare the performance of NeRFool with IL2-NeRF, we maintain a controlled environment where only one parameter is variable and the rest are fixed. We compare attack performance and pixel-wide perturbations between ground truth source images and the predicted outputs from feeding the perturbed source images.

4.1. Experiment Setup

Models We run both NeRFool and IL2-NeRF on two SOTA GNeRF methods: IBRNet [28] and GNT [27]. We use the pre-trained weights provided by their corresponding implementations.

Datasets On IBRNet and GNT, we run experiments across three different datasets: LLFF [21], DeepVoxels [25], and Synthetic [20]. We present results for GNT on LLFF and DeepVoxels here. We run attacks on evaluation images from eight objects and scenes from LLFF, four objects in DeepVoxels, and eight objects from Synthetic.

Attack Parameters To ensure a controlled experiment environment, we fix all hyperparameters besides our maximum perturbation factor ϵ. For both NeRFool and IL2-NeRF, we fix the number of steps T to 1000. Both attacks use a learning rate of 1 (so $\eta = 1$, $\alpha = 1/255$) and four source views. The initial adversarial perturbation δ_0 is sampled uniformly from $\mathcal{U}[-\epsilon, \epsilon]$. Both attacks are set to perturb both color and density.

Evaluating Attack Performance There are three metrics we use to evaluate attack performance. PSNR, or Peak Signal-to-Noise Ratio, represents the reconstruction accuracy and is our main metric [34]. A lower PSNR indicates a poor scene generation and thus a more successful attack.

SSIM, or structural similarity, compares the local patterns of pixel intensities by normalizing using the mean intensity and taking luminance as a contrast comparison [30]. A lower SSIM score indicates a more successful attack.

LPIPS, or Learned Perceptual Image Patch Similarity, takes the L_2 distance between averaged unit-normalized channels for an image [34]. Unlike PSNR and SSIM, a higher LPIPS score means a more successful attack.

Comparing Epsilon We vary ϵ starting at 8 for IL2-NeRF and consider powers of two until we perform similarly to NeRFool. As a benchmark, we fix the perturbation factor ϵ to 8 for NeRFool. It is important to note that the ϵ values imply different interpretations of perturbation magnitudes when working under different norms (such as L_∞ and L_2).

For an L_∞ attack like NeRFool with $\epsilon = 8$, the perturbation constraint allows each pixel in the input to be independently perturbed by up to 8 units, which can lead to a large and consistent maximum distortion across all pixels.

In contrast, for an L_2 attack like ours, $\epsilon = 8$ signifies that the total perturbation energy (i.e., the sum of squared perturbations across all pixels) must remain within 8. This constraint in the L_2 norm distributes the perturbation across multiple dimensions, resulting in smaller individual pixel changes than the maximum possible under the L_∞ constraint.

Thus, even though a large perturbation factor like $\epsilon = 64$ in L_2 may appear large, it actually enforces a more dispersed, lower-magnitude perturbation at the pixel level when compared to $\epsilon = 8$ in L_∞. This explains why the L_2 norm attack with a higher ϵ results in a smaller perceptible perturbation than the L_∞ attack with a smaller ϵ.

4.2. Experiment Evaluation

LLFF Attack Results Tables 1, 2, 3 and 4 showcase results from varying ϵ on IL2-NeRF when compared to NeRFool on $\epsilon = 8$ across all scenes for the LLFF dataset on IBRNet. We interpret the IBRNet as our baseline GNeRF model and likewise LLFF as our most standard dataset for GNeRFs.

Table 1 reports the PSNR value of NeRFool against IL2-NeRF on all eight scenes from the LLFF dataset. We compare NeRFool on $\epsilon = 8$ to IL2-NeRF on five values of ϵ from 8 to 256. For each scene, as our perturbation factor ϵ increases, we notice the PSNR that IL2-NeRF achieves decreases monotonically, closer to NeRFool. Once we reach

LLFF PSNR

	Model	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	**IBRNet**	8	13.145	14.428	**12.944**	**11.682**	14.045	11.042	**12.091**	11.526	12.613
	GNT	8	14.921	15.428	**14.165**	14.134	13.946	12.348	13.253	12.773	13.871
IL2-NeRF	**IBRNet**	8	21.581	25.214	24.605	22.967	18.696	17.960	24.172	20.188	21.923
		16	20.590	24.296	21.652	21.458	18.450	17.438	21.071	18.355	20.414
		64	16.825	17.852	15.862	15.742	16.740	14.030	16.120	14.240	15.926
		128	14.900	15.220	14.299	13.945	14.708	11.950	13.837	12.731	13.949
		256	**13.02**	**13.193**	13.462	12.028	**12.212**	**9.920**	12.518	**11.485**	**12.230**
	GNT	256	**13.381**	**13.367**	14.449	**12.394**	**11.912**	**10.151**	**12.649**	**11.453**	**12.470**

Table 1. PSNR of NeRFool vs. IL2-NeRF on IBRNet model, LLFF dataset. Note that a lower PSNR indicates a more successful attack.

LLFF SSIM

	Model	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	**IBRNet**	8	0.473	0.594	0.539	0.513	0.442	0.311	0.658	0.549	0.510
	GNT	8	0.470	0.515	0.462	0.541	0.391	0.327	0.626	0.520	0.482
IL2-NeRF	**IBRNet**	8	0.694	0.836	0.790	0.809	0.641	0.565	0.908	0.792	0.754
		16	0.670	0.825	0.740	0.784	0.631	0.548	0.881	0.767	0.731
		64	0.564	0.705	0.570	0.635	0.551	0.426	0.764	0.656	0.609
		128	0.485	**0.579**	**0.487**	0.516	0.442	0.320	0.686	0.567	0.510
		256	**0.405**	**0.435**	**0.451**	**0.390**	**0.271**	**0.184**	**0.616**	**0.437**	**0.399**
	GNT	256	**0.353**	**0.307**	**0.388**	**0.356**	**0.180**	**0.144**	**0.545**	**0.365**	**0.330**

Table 2. SSIM of NeRFool vs. IL2-NeRF on IBRNet model, LLFF dataset. Note that a lower SSIM indicates a more successful attack.

LLFF LPIPS

	Model	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	**IBRNet**	8	0.477	0.407	0.471	0.479	0.409	0.566	0.447	0.484	0.468
	GNT	8	0.375	0.338	0.391	0.358	0.369	0.421	0.351	0.388	0.374
IL2-NeRF	**IBRNet**	8	0.299	0.180	0.232	0.236	0.272	0.347	0.199	0.305	0.259
		16	0.326	0.193	0.281	0.263	0.280	0.360	0.233	0.330	0.283
		64	0.430	0.322	0.444	0.404	0.346	0.476	0.374	0.425	0.403
		128	**0.510**	**0.440**	**0.510**	**0.505**	**0.432**	**0.576**	**0.466**	**0.499**	**0.492**
		256	**0.578**	**0.550**	**0.538**	**0.596**	**0.556**	**0.690**	**0.534**	**0.587**	**0.579**
	GNT	256	**0.483**	**0.484**	**0.469**	**0.500**	**0.499**	**0.560**	**0.447**	**0.506**	**0.494**

Table 3. LPIPS of NeRFool vs. IL2-NeRF on IBRNet model, LLFF dataset. Note that a higher LPIPS indicates a more successful attack.

IBRNet LLFF Average L2 Distance

	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	8	343.268	**288.048**	354.545	393.361	**299.904**	**426.528**	377.955	407.624	**361.404**
IL2-NeRF	8	126.494	85.853	94.222	114.872	176.605	192.003	100.132	148.946	129.891
	16	142.021	95.403	128.997	137.913	181.555	203.898	138.884	184.992	151.708
	64	218.791	195.606	249.270	252.401	220.501	304.433	240.540	298.844	247.548
	128	274.809	264.418	293.947	308.864	278.835	384.506	310.953	352.868	308.650
	256	**339.786**	333.319	**323.590**	**382.289**	372.280	483.862	**360.341**	**405.359**	375.103

Table 4. Average L2 Difference between ground truth source images and predicted outputs when IBRNet is provided perturbed source images as input on LLFF dataset. Note that a lower L2 distance is desired.

IBRNet LLFF PSNR

	$\eta =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
IL2-NeRF	0.001	9.848	11.571	11.231	9.547	10.332	8.179	25.500	9.406	11.952
	0.05	9.849	11.561	10.899	9.587	10.282	8.143	13.932	9.525	10.473
	0.1	9.906	11.558	10.886	9.531	10.326	8.135	13.795	9.612	10.469

Table 5. PSNR of IL2-NeRF on IBRNet and GNT models with variable learning rate, LLFF dataset. Note that a lower PSNR indicates a more successful attack.

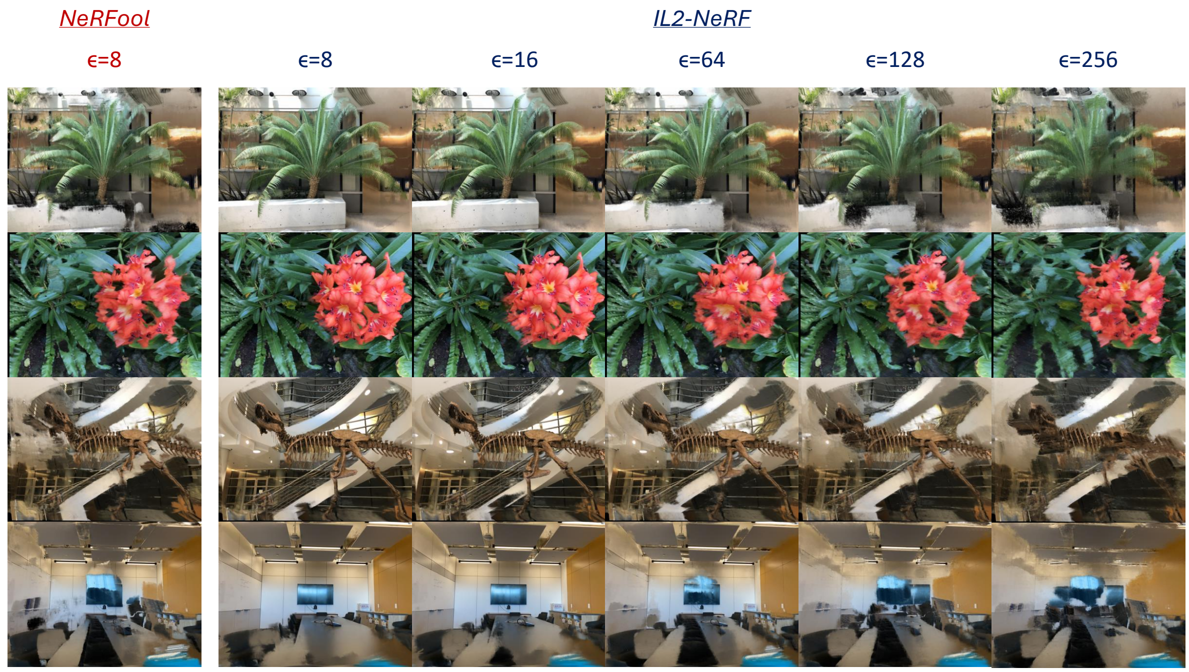

Figure 1. Visual comparing predicted images on four LLFF scenes from IBRNet on NeRFool and IL2-NeRF perturbed images on varying perturbation factors ϵ.

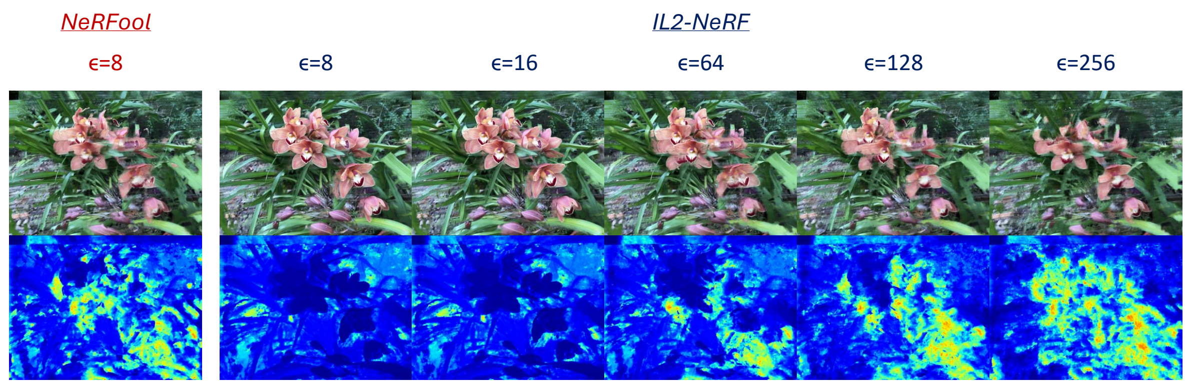

Figure 2. Visual comparing depth masks of Orchids predicted image from IBRNet on NeRFool and IL2-NeRF perturbed images on varying perturbation factors ϵ.

IBRNet LLFF SSIM

	$\eta =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
IL2-NeRF	0.001	0.296	0.327	0.388	0.291	0.150	0.089	0.876	0.319	0.342
	0.05	0.295	0.330	0.387	0.292	0.150	0.089	0.686	0.323	0.319
	0.1	0.292	0.328	0.385	0.292	0.150	0.090	0.686	0.323	0.318

Table 6. SSIM of IL2-NeRF on IBRNet and GNT models with variable learning rate, LLFF dataset. Note that a lower SSIM indicates a more successful attack.

IBRNet LLFF LPIPS

	$\eta =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
IL2-NeRF	0.001	0.604	0.583	0.571	0.628	0.586	0.711	0.295	0.626	0.576
	0.05	0.603	0.583	0.571	0.626	0.585	0.713	0.468	0.623	0.597
	0.1	0.605	0.586	0.573	0.626	0.584	0.713	0.467	0.623	0.597

Table 7. LPIPS of IL2-NeRF on IBRNet and GNT models with variable learning rate, LLFF dataset. Note that a higher LPIPS indicates a more successful attack.

CVPR #17334

IBRNet Synthetic PSNR

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
NeRFool	**13.330**	14.490	**12.300**	**10.095**	13.510	10.874	10.720	9.824	11.893
IL2-NeRF	13.727	**11.758**	13.327	12.286	**11.104**	**9.024**	**10.450**	**9.780**	**11.432**

Table 8. PSNR of NeRFool vs. IL2-NeRF on IBRNet, Synthetic dataset. Note that a lower PSNR indicates a more successful attack.

IBRNet Synthetic SSIM

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
NeRFool	0.855	0.861	0.818	0.797	0.829	0.802	0.803	0.663	0.804
IL2-NeRF	**0.791**	**0.700**	**0.763**	**0.754**	**0.635**	**0.640**	**0.735**	**0.581**	**0.700**

Table 9. SSIM of NeRFool vs. IL2-NeRF on IBRNet, Synthetic dataset. Note that a lower SSIM indicates a more successful attack.

IBRNet Synthetic LPIPS

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
NeRFool	0.231	0.221	0.236	0.263	0.232	0.252	0.244	0.386	0.258
IL2-NeRF	**0.338**	**0.378**	**0.337**	**0.383**	**0.418**	**0.397**	**0.352**	**0.456**	**0.820**

Table 10. LPIPS of NeRFool vs. IL2-NeRF on IBRNet, Synthetic dataset. Note that a higher LPIPS indicates a more successful attack.

DeepVoxels PSNR

	Model	Armchair	Cube	Greek	Vase	Avg.
NeRFool	**IBRNet**	9.500	13.982	**11.688**	11.437	11.652
	GNT	13.070	17.991	15.532	19.540	16.533
IL2-NeRF	**IBRNet**	**8.660**	**11.829**	12.067	**10.235**	**10.700**
	GNT	**11.959**	**13.874**	**13.414**	**13.312**	**13.140**

Table 11. PSNR of NeRFool vs. IL2-NeRF on IBRNet and GNT models, DeepVoxels dataset. Note that a lower PSNR indicates a more successful attack.

DeepVoxels SSIM

	Model	Armchair	Cube	Greek	Vase	Avg.
NeRFool	**IBRNet**	0.760	0.668	0.772	0.761	0.745
	GNT	0.833	0.826	0.789	0.921	0.842
IL2-NeRF	**IBRNet**	**0.728**	**0.591**	**0.760**	**0.684**	**0.691**
	GNT	**0.796**	**0.253**	**0.684**	**0.781**	**0.629**

Table 12. SSIM of NeRFool vs. IL2-NeRF on IBRNet and GNT models, DeepVoxels dataset. Note that a lower PSNR indicates a more successful attack.

DeepVoxels LPIPS

	Model	Armchair	Cube	Greek	Vase	Avg.
NeRFool	**NeRFool**	0.303	0.285	0.291	0.231	0.278
	GNT	0.188	0.139	0.191	0.076	0.149
IL2-NeRF	**IL2-NeRF**	**0.360**	**0.373**	**0.314**	**0.291**	**0.335**
	GNT	**0.231**	**0.676**	**0.266**	**0.190**	**0.331**

Table 13. LPIPS of NeRFool vs. IL2-NeRF on IBRNet and GNT models, DeepVoxels dataset. Note that a higher LPIPS indicates a more successful attack.

$\epsilon = 256$, the PSNR of IL2-NeRF is lower than NeRFool for five out of eight scenes, achieving a PSNR that is on-average 0.383 lower.

Table 2 shows the SSIM value of NeRFool against IL2-NeRF on all eight scenes from the LLFF dataset. At $\epsilon = 128$, IL2-NeRF achieves a lower SSIM across two scenes than NeRFool $\epsilon = 8$: Flower and Fortress. Furthermore, the average SSIM that IL2-NeRF achieves is exactly the same as the average SSIM that NeRFool achieves at $\epsilon = 8$ of 0.510.

We also see that at $\epsilon = 256$, IL2-NeRF achieves a lower SSIM than NeRFool across all scenes. IL2-NeRF at $\epsilon = 256$ achieves an average SSIM of 0.399, which is a difference of 0.111 lower than the average SSIM of NeRFool $\epsilon = 8$ on LLFF.

Table 3 holds the LPIPS value of NeRFool against IL2-NeRF on all eight scenes from the LLFF dataset. Starting at $\epsilon = 128$, IL2-NeRF achieves a higher LPIPS across all scenes when compared to NeRFool $\epsilon = 8$. NeRFool $\epsilon = 8$ achieves an average LPIPS of 0.468 whereas IL2-NeRF $\epsilon = 128$ achieves an average LPIPS of 0.492, a difference of 0.24. At $\epsilon = 256$, IL2-NeRF achieves an average LPIPS of 0.579, resulting in a difference of 0.111.

To test this trend that IL2-NeRF receives better attack metrics for $\epsilon = 256$ when compared to NeRFool $\epsilon = 8$, we run our attacks on the GNT model as well for LLFF. Table 1 shows us that IL2-NeRF achieves a lower PSNR on seven out of eight scenes for $\epsilon = 256$. Likewise, Tables 2 and 3 tells us IL2-NeRF earns a higher SSIM and lower LPIPS across all scenes.

Overall, **for IBRNet, IL2-NeRF $\epsilon = 128$ performs comparably to NeRFool $\epsilon = 8$, with worse average PSNR and better LPIPS. On both IBRNet and GNT, IL2-NeRF $\epsilon = 256$ outperforms NeRFool $\epsilon = 8$ across most scenes in LLFF for PSNR and all scenes on LLFF for SSIM and LPIPS.** This proves that L_2 attacks can surpass L_∞ attacks on GNeRFs.

LLFF Perturbation Results PSNR, SSIM and LPIPS were all the metrics used to study adversarial perturbations in NeRFool [8]. To complete our analysis, we need to consider the immediate affects that our perturbed images have on GNeRF scene generation. We report these findings in Table 4 and visuals in Figure 1 and 2.

Table 4 shows the average L_2 distance across all our ground-truth source images and the predicted images. These predicted images are produced by the GNeRF model after feeding in the perturbed source images for each scene in LLFF. **Up to $\epsilon = 128$ for IL2-NeRF, all L_2 distances for each scene are smaller than the L_2 distances for each scene for NeRFool $\epsilon = 8$.**

Five out of our eight scenes achieve a lower average L_2 distance for IL2-NeRF at $\epsilon = 256$ than NeRFool on $\epsilon = 8$: Fern, Fortress, Horns, Room, and T-Rex. The average L_2 distance across all scenes for IL2-NeRF is 375.103, which is larger than the averaged L2 distance across all scenes for NeRFool $\epsilon = 8$ of 361.404 by 13.699. Despite this, if we remove the two scenes with the largest difference between NeRFool $\epsilon = 8$ and IL2-NeRF $\epsilon = 256$, Leaves and Orchids, the average for IL2-NeRF becomes 357.447.

We provide a visual for these predicted images in Figure 1. We show the GNeRF outputs for images in four scenes of LLFF from top to bottom: Fern, Flower, T-Rex and Room. At $\epsilon = 8, 16$, IBRNet's output have very minimal degradations. It is not until $\epsilon = 64$ that IBRNet produces noticeable perturbations, but these are not as intense as NeRFool $\epsilon = 8$. At $\epsilon = 128, 256$, the IBRNet outputs appear similarly distorted as from providing NeRFool $\epsilon = 8$ as input.

To better visualize these differences in degradations, we provide another visual in Figure 2. This figure shows the depth mask of our predicted scenes for the Orchid scene in LLFF. These depth masks provide a clear visual for where artifacts are added as we can compare the intensity and addition of said artifacts as we vary ϵ.

Again, we notice that at $\epsilon = 8, 16$, any perturbations added by IL2-NeRF are minimally visible. Artifacts that create clear contrast with the remaining depth mask are noticeable at $\epsilon = 64$. At $\epsilon = 128, 256$, these artifacts intensify and start to consume most of the depth mask, similar to NeRFool $\epsilon = 8$.

From Tables 1, 2 and 3 we have found that IL2-NeRF $\epsilon = 256$ outperforms NeRFool $\epsilon = 8$ by our attack metrics. We wish to study if this trend holds across different datasets. For subsequent experiments on new datasets, we fix $\epsilon = 256$ for IL2-NeRF.

Learning Rate LLFF Results Tables 5, 6 and 7 report our attack metrics that result from varying the learning rate η, where $\alpha = \frac{\eta}{255}$, when we run IL2-NeRF on IBRNet on the LLFF dataset with fixed perturbation factor $\epsilon = 128$. We use three learning rates 0.001, to 0.05, to 0.1 to study the effects of perturbations on a log scale smaller than $\eta = 1$.

Table 5 compares the PSNR of running IL2-NeRF on all scenes of LLFF on IBRNet when η is variable. Interestingly, as η increases, PSNR does not always follow a monotonic behavior for all scenes. Here, the PSNR for Flower, Fortress, Orchids and Room decreases as η increases.

Tables 6 and 7 show SSIM and LPIPS, respectively, when we vary η. We see that across all scenes, as we varied η from 0.001 to 0.1 the change in SSIM and LPIPS is negligible. This validates using a learning rate $\eta \geq 1$ for our attack setting.

DeepVoxels Attack Results Tables 11, 12 and 13 report results from running NeRFool $\epsilon = 8$ and IL2-NeRF $\epsilon = 256$ on all four scenes of the DeepVoxels dataset on both IBRNet and GNT.

Table 11 compares the PSNR that both attacks achieve on DeepVoxels. Here, IL2-NeRF achieves a lower PSNR on three out of four scenes for IBRNet and all scenes for GNT. Likewise, Table 12 and Table 13 shows that IL2-NeRF gives us a lower SSIM and higher LPIPS respective on both IBRNet and GNT.

Synthetic Attack Results Tables 8, 9 and 10 showcase results from running NeRFool $\epsilon = 8$ and IL2-NeRF $\epsilon = 256$ on all eight scenes of the Synthetic dataset.

As shown in Table 11, IL2-NeRF achieves a lower PSNR on five out of eight scenes for IBRNet. Furthermore, Table 12 and Table 13 shows that IL2-NeRF gives us a lower SSIM and higher LPIPS respective on both IBRNet.

Experiment Conclusion We have shown that for our base model IBRNet, on our most standard dataset LLFF that IL2-NeRF at $\epsilon = 128$ produces comparable metrics to NeRFool $\epsilon = 8$. We have further shown that across all three datasets and two models that IL2-NeRF at $\epsilon = 256$ outperforms NeRFool at $\epsilon = 8$.

We acknowledge that there is future work in exploring adversarial methods that produce better metrics under smaller L_2-norm bounds. However, our adversarial algorithm proves that it is possible for L_2 adversarial attacks to achieve success in compromising GNeRF robustness.

5. Conclusion

By introducing IL2-NeRF, we have laid the groundwork for future studies in L_2-based adversarial robustness for GNeRFs. Our baseline threat model and metrics provide a foundation for advancing adversarial 3D reconstruction, offering a new perspective on how L_2 domain attacks can improve robustness testing for neural radiance fields.

As machine learning models see further deployment across social and ethical fields, this research aims to highlight vulnerabilities to drive safer model deployment by examining potential risks. Our work paves the way for future adversarial attacks on GNeRF models that work under the L_2 threat model. The advent of L_2 attacks on GNeRFs will open the door for geometric-based methods and black-box attacks for evaluating GNeRF robustness.

Ethical Considerations of Our Findings: We acknowledge that developing effective and efficient adversarial attacks on generative AI models can be destructive to a certain extent. Attackers could potentially use these algorithms to compromise systems implemented in real-life applications. However, the purpose of inventing such attacks is to expose vulnerabilities and encourage the development of more robust defensive systems.

References

[1] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196, 2021. 2, 3

[2] Christopher M Bishop. Pattern recognition and machine learning. 2006. 4

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 2

[4] Sizhe Chen, Qinghua Tao, Zhixing Ye, and Xiaolin Huang. Measuring the transferability of attacks by the 2 norm. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 3

[5] Celia Cintas, Skyler Speakman, Victor Akinwande, William Ogallo, Komminist Weldemariam, Srihari Sridharan, and Edward McFowland. Detecting adversarial attacks via subset scanning of autoencoder activations and reconstruction error. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 876–882, 2021. 2, 3

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 2, 4

[7] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2, 4

[8] Yonggan Fu, Ye Yuan, Souvik Kundu, Shang Wu, Shunyao Zhang, and Yingyan Celine Lin. Nerfool: Uncovering the vulnerability of generalizable neural radiance fields against adversarial perturbations. In *International Conference on Machine Learning*, pages 10482–10493. PMLR, 2023. 1, 2, 7

[9] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 1

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2, 4

[11] Jindong Gu, Hengshuang Zhao, Volker Tresp, and Philip HS Torr. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, pages 308–325. Springer, 2022. 2, 4

[12] Wenxiang Jiang, Hanwei Zhang, Xi Wang, Zhongwen Guo, and Hao Wang. Nerfail: Neural radiance fields-based multiview adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21197–21205, 2024. 3

[13] Wenxiang Jiang, Hanwei Zhang, Shuo Zhao, Zhongwen Guo, and Hao Wang. Ipa-nerf: Illusory poisoning attack against neural radiance fields. In *ECAI 2024*, pages 513–520. IOS Press, 2024. 3

[14] Qiaoyi Li, Zhengjie Wang, Xiaoning Zhang, and Yang Li. Attack-cosm: attacking the camouflaged object segmentation model through digital world adversarial examples. *Complex & Intelligent Systems*, pages 1–13, 2024. 2, 4

[15] Jing Lin, Long Dang, Mohamed Rahouti, and Kaiqi Xiong. Ml attack models: Adversarial attacks and data poisoning attacks. *arXiv preprint arXiv:2112.02797*, 2021. 2

[16] Renyang Liu, Xin Jin, Dongting Hu, Jinhong Zhang, Yuanyu Wang, Jin Zhang, and Wei Zhou. Dualflow: Generating imperceptible adversarial examples by flow field and normalize flow-based model. *Frontiers in Neurorobotics*, 17:1129720, 2023. 2, 4

[17] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 2

[18] Mingfei Lu and Badong Chen. On the adversarial robustness of generative autoencoders in the latent space. *Neural Computing and Applications*, 36(14):8109–8123, 2024. 2, 3

[19] Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 3, 4

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

[21] 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. 1, 2, 4

[22] Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai. Cgba: curvature-aware geometric black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 124–133, 2023. 3

[23] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4322–4330, 2019. 2, 3

[24] Meng Shen, Changyue Li, Hao Yu, Qi Li, Liehuang Zhu, and Ke Xu. Decision-based query efficient adversarial attack via adaptive boundary learning. *IEEE Transactions on Dependable and Secure Computing*, 2023. 3

[25] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 4

[26] Jie Wan, Jianhao Fu, Lijin Wang, and Ziqi Yang. Bounceattack: A query-efficient decision-based adversarial attack by bouncing into the wild. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1270–1286. IEEE, 2024. 3

[27] Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, et al. Is attention all that nerf needs? *arXiv preprint arXiv:2207.13298*, 2022. 2, 4

[28] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 4

[29] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *European conference on computer vision*, pages 156–174. Springer, 2022. 3

[30] 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. 4

[31] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2

[32] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 2

[33] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

[34] 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. 4

[35] Zhuosheng Zhang, Noor Ahmed, and Shucheng Yu. Qedba: Query-efficient decision-based adversarial attacks via bayesian optimization. In *2024 International Conference on Computing, Networking and Communications (ICNC)*, pages 783–788. IEEE, 2024. 3