Supplementary Material for Advancing Adversarial Robustness in GNeRFs: The IL2-NeRF Attack

Nicole Meng[1,2] Caleb Manicke[2] Ronak Sahu[2] Caiwen Ding[3] Yingjie Lao[1]

1. NeRF Pipeline

This section provides an in-depth, comprehensive review of the NeRF training pipeline. We provide a detailed explanation of how machine learning models can optimize a continuous volumentric scene function to synthesize novel views of complex scenes.

NeRF take multiple 2D images as input with their respective locations from where the image was taken from the camera (hence, 5D data) to render a 3D volumetric model. The model is trained using the following pipeline, which we break into five steps:

1. **Fetching the Light Rays:** In computer vision, the 3D coordinate of a camera is stored in the form of the location (x, y, z) and direction (θ, ϕ). Then, rays are formed based on the image size, denoted $r_o \in \mathbb{R}^3$ (ray origin/camera center) and $r_d \in \mathbb{R}^3$ (ray direction). This is done in 2 stages:
   (a) **Conversion from pixel to camera:** Every image pixel indices (i, j) is converted into $(\frac{i-w/2}{f}, \frac{i-h/2}{f}, -1)$, where w and h are the width and height of the image respectively and f is the focal length of the camera. In the z-axis, we use -1 to denote the OpenGL convention.
   (b) **Conversion from camera to world:** Following the pixel-to-camera transformation, all pixel values undergo a linear transformation using the rotation component R of the extrinsic matrix to compute r_d, while r_o is derived from the translation component t of the extrinsic matrix, represented as $[R \,|\, t]$.
2. **Sampling the Rays:** After the rays are generated, they are broken down into small chunks. The model learns the color and density of each small chunk rather than the whole ray. Each chunk's color and density can later be synthesized into the color along the ray. I.e., the ray segment becomes $r_t = r_o + t_i r_d$, where $t_i \sim \mathcal{U}[t_n + \frac{i}{N}(t_f - t_n), t_n + \frac{i+1}{N}(t_f - t_n)]$.
3. **Positional Encoding:** The individual ray chunk coordinates are encoded into an accumulation of $[\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^L\pi p), \cos(2^L\pi p)]$, where p is the value given to the ray chunk after segmenting the ray. Experimental results revealed that the best values for location coordinates are $L = 4$ and $L = 10$ for directional coordinates. This helps the model to uniquely represent the color and density based on the input chunk value the model was queried with.
4. **Deep Learning Model:** The model consists of 8 fully connected layers, each with ReLU activation and 256 channels, where the positionally encoded 3D coordinate (x, y, z) is processed. This generates a volume density, σ, and a 256-dimensional feature vector. This feature vector is then concatenated with the positionally encoded camera ray's viewing direction and passed through an additional 128 channel fully connected layer with ReLU activation, producing the view-dependent RGB color, c.
5. **Volumetric Rendering:** The rendering model comprises 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 the 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 calculate the attenuation of the ray as it travels.

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

[1] Tufts University,[2] University of Connecticut, [3] University of Minnesota

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

Using the following pipeline, two different NeRF models are trained—a coarse and a fine network—which is distinguishable based on the amount of sampling done on the rays. The coarse model captures general color and intensity across broader regions, providing a rough approximation. In contrast, the fine model focuses on a refined subset of samples, allowing it to learn finer details and subtle variations in color and intensity within the smaller chunks. This combination enhances the model's ability to represent complex scene details with high fidelity. Finally, after the volumetric rendering, we get 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. Mathematically, the loss function is as follows

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

for all the accumulation of rays as $\mathcal{R}$ where $C_c(r)$ is the output from the coarse NeRF model and $C_f(r)$ is the output for the fine NeRF model.

2. Additional Experiments

Here, we present additional experimental results that expand on our findings presented in the main experiments section. Our findings here feature results on the GNT model [14] from varying the perturbation factor ϵ on the LLFF dataset [10], varying ϵ for IBRNet on DeepVoxels [13], and numerous figures.

Variable Epsilon GNT, LLFF In the main experiments section, we report metrics from running NeRFool and IL2-NeRF on LLFF for both the IBRNet and GNT models. On IL2-NeRF, we vary the perturbation factor ϵ from 8 to 256 for IBRNet and fix ϵ to 256 for GNT. Here, Tables 1, 2, and 3 present results for varying ϵ on GNT.

Table 1 shows the PSNR value of NeRFool against IL2-NeRF on all eight scenes from the LLFF dataset. Once we reach $\epsilon = 256$, the PSNR of IL2-NeRF is lower than NeRFool on all eight scenes for GNT, achieving a PSNR that is on average 1.378 lower. This outperforms results on IBRNet, where IL2-NeRF at $\epsilon = 256$ achieved a lower PSNR on five out of eight scenes for IBRNet.

Table 2 gives 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 on GNT across five scenes than NeRFool $\epsilon = 8$. Again, IL2-NeRF performs better on GNT than IBRNet, as IL2-NeRF on IBRNet at $\epsilon = 256$ achieves a lower SSIM on only two scenes.

Table 3 reports 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 on GNT when compared to NeRFool $\epsilon = 8$. This trend remains for our different model, where IL2-NeRF $\epsilon = 128$ again achieves a higher LPIPS on all scenes on IBRNet.

Overall, **for LLFF, IL2-NeRF performs better on GNT than on IBRNet**. Our conclusion is similar: IL2-NeRF $\epsilon = 128$ performs comparably to NeRFool $\epsilon = 8$ on GNT, with worse average PSNR and better LPIPS. IL2-NeRF $\epsilon = 256$ on GNT outperforms on IBRNet, achieving lower PSNR and SSIM than NeRFool $\epsilon = 8$ on more scenes.

Variable Epsilon IBRNet, DeepVoxels The main experiment section shows attack metrics from running NeRFool $\epsilon = 8$ and IL2-NeRF $\epsilon = 256$ on both IBRNet and GNT on the DeepVoxels dataset from fixing epsilon. Here, Tables 7, 8 and 9 expand these results by varying ϵ on IBRNet.

Table 7 reports the PSNR value from running IL2-NeRF on all four scenes on DeepVoxels for IBRNet across five values of ϵ. At $\epsilon = 256$, IL2-NeRF achieves a lower PSNR on three out of four scenes than NeRFool $\epsilon = 8$ with an average PSNR lower by 0.952. At $\epsilon = 128$, IL2-NeRF is on average still lower than NeRFool $\epsilon = 8$ for PSNR by 0.164.

Table 8 reports SSIM for IL2-NeRF on all five values of ϵ against NeRFool $\epsilon = 8$ on IBRNet, DeepVoxels. At $\epsilon = 128$, the average IL2-NeRF SSIM is lower than the average NeRFool SSIM by a slight 0.017. IL2-NeRF outperforms NeRFool at $\epsilon = 256$ on all scenes, with an average SSIM lower by 0.054.

Table 9 reports LPIPS for IL2-NeRF on all five values of ϵ against NeRFool $\epsilon = 8$ on IBRNet, DeepVoxels. IL2-NeRF receives a greater or larger LPIPS across all scenes at $\epsilon = 128$, with an average that is greater by 0.021. This difference becomes 0.057 at $\epsilon = 256$.

Variable Loss Weights IBRNet, LLFF We used a weighted sum of eight total losses: coarse RGB, fine RGB, density, depth variable, depth difference, depth consistency, depth smoothness, and multi-view consistency, which provides gradient direction for the perturbation term to target all factors in reconstructive accuracy.

In the main experiments section, we report attack metrics for running IL2-NeRF on variable ϵ and a loss combining 8 different loss functions with a fixed weight. We used the same weights in NeRFool to maintain consistency. Here, we include results for varying weights from IL2-NeRF on IBRNet, LLFF outlined below:

- **RGB-Loss**: This is the original loss featured in the main experiments section and in NeRFool. RGB-Loss gives the coarse RGB and fine RGB a weight term of 1 and sets everything else to 0.
- **Density & Depth:** Here, we give density and depth variable loss a weight of 0.5 and maintain a weight of 1 for both RGB losses.
- **Diff & Smooth:** This combination gives the depth dif-

GNT LLFF PSNR

	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	8	14.930	15.453	14.088	14.127	13.946	12.341	13.173	12.725	13.848
IL2-NeRF	8	21.715	23.467	25.413	23.345	18.276	17.171	23.802	20.700	21.736
	16	20.966	22.114	22.600	22.052	17.703	16.690	21.260	19.090	20.309
	64	17.637	18.470	17.558	17.277	15.555	14.573	16.330	15.085	16.561
	128	15.645	15.920	15.809	14.939	14.023	12.421	14.587	13.362	14.588
	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 GNT model, LLFF dataset. Note that a lower PSNR indicates a more successful attack.

GNT LLFF SSIM

	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	8	0.470	0.513	0.461	0.540	0.390	0.328	0.623	0.520	0.481
IL2-NeRF	8	0.731	0.797	0.825	0.842	0.672	0.574	0.884	0.801	0.766
	16	0.707	0.762	0.790	0.812	0.642	0.552	0.855	0.772	0.737
	64	0.590	0.616	0.621	0.660	0.503	0.434	0.742	0.647	0.602
	128	**0.385**	**0.481**	0.504	**0.538**	**0.370**	**0.311**	0.656	0.538	**0.473**
	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 GNT model, LLFF dataset. Note that a lower SSIM indicates a more successful attack.

GNT LLFF LPIPS

	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
NeRFool	8	0.374	0.339	0.388	0.359	0.369	0.421	0.352	0.388	0.374
IL2-NeRF	8	0.203	0.145	0.150	0.150	0.217	0.268	0.141	0.191	0.183
	16	0.221	0.170	0.177	0.174	0.234	0.281	0.170	0.214	0.205
	64	0.306	0.271	0.319	0.289	0.314	0.359	0.274	0.311	0.305
	128	**0.488**	**0.366**	**0.407**	**0.377**	**0.395**	**0.448**	**0.353**	**0.392**	**0.403**
	256	**0.483**	**0.484**	**0.469**	**0.500**	**0.499**	**0.560**	**0.448**	**0.506**	**0.494**

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

IBRNet LLFF PSNR

Weights	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
RGB-Loss	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	**13.020**	**13.193**	**13.462**	**12.028**	**12.212**	**9.920**	**12.518**	**11.485**	**12.230**
Density & Depth	8	22.113	25.805	27.834	24.325	18.999	18.343	28.296	22.701	23.552
	16	21.996	25.654	27.498	24.179	18.945	18.326	27.947	22.432	23.372
	64	21.431	25.128	26.456	23.448	18.856	18.044	26.731	21.884	22.747
	128	20.764	24.434	25.450	22.660	18.543	17.500	25.081	21.127	21.945
	256	19.846	22.212	23.158	20.772	17.495	16.016	23.337	19.560	20.300
Diff & Smooth	8	21.939	25.610	26.982	24.090	18.980	18.255	27.518	22.042	23.177
	16	21.600	25.149	25.888	23.537	18.893	18.050	26.522	21.624	22.658
	64	19.869	22.347	23.898	20.974	17.873	16.307	23.866	20.244	20.672
	128	18.996	19.913	22.084	20.117	17.267	14.857	22.226	19.155	19.327
	256	18.247	18.120	20.767	18.643	16.741	13.614	21.123	18.108	18.170

Table 4. PSNR of IL2-NeRF on IBRNet model, LLFF dataset on different combinations of summed loss weights. Note that a lower PSNR indicates a more successful attack.

ference, depth smoothness and depth variable losses a weight of 0.2, density loss weight 0.4, and both RGB losses 1.

Table 4 reports the PSNR value from running IL2-NeRF on variable ϵ across all three weight combinations. On all scenes, RGB-Loss obtains a lower PSNR than Density & Depth and Diff & Smooth. On average, RGB-Loss outperforms both Density & Depth and Diff & Smooth at every

IBRNet LLFF SSIM

Weights	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
RGB-Loss	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**
Density & Depth	8	0.706	0.843	0.819	0.824	0.650	0.576	0.929	0.812	0.770
	16	0.700	0.840	0.801	0.817	0.649	0.574	0.924	0.805	0.764
	64	0.659	0.814	0.708	0.758	0.630	0.545	0.896	0.765	0.722
	128	0.618	0.777	0.643	0.700	0.588	0.497	0.855	0.715	0.674
	256	0.563	0.691	0.578	0.602	0.488	0.398	0.805	0.623	0.593
Diff & Smooth	8	0.698	0.840	**0.785**	0.817	0.650	0.571	0.919	0.801	0.760
	16	0.682	0.828	**0.739**	0.794	0.642	0.558	0.905	0.784	0.742
	64	0.599	0.748	0.628	0.677	0.555	0.459	0.833	0.688	0.648
	128	0.547	0.663	0.569	0.602	0.493	0.368	0.784	0.613	0.580
	256	0.496	0.564	0.527	0.520	0.432	0.277	0.736	0.532	0.510

Table 5. SSIM of IL2-NeRF on IBRNet model, LLFF dataset on different combinations of summed loss weights. Note that a lower SSIM indicates a more successful attack.

IBRNet LLFF LPIPS

Weights	$\epsilon =$	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-Rex	Avg.
RGB-Loss	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**
Density & Depth	8	0.292	0.175	0.214	0.224	0.267	0.340	0.179	0.287	0.247
	16	0.303	0.181	0.237	0.235	0.268	0.345	0.191	0.296	0.257
	64	0.370	0.220	0.339	0.310	0.292	0.382	0.255	0.349	0.315
	128	0.426	0.267	0.404	0.376	0.338	0.436	0.331	0.405	0.373
	256	0.488	0.360	0.461	0.470	0.433	0.534	0.409	0.490	0.455
Diff & Smooth	8	**0.306**	**0.181**	**0.250**	**0.237**	0.269	**0.349**	**0.200**	0.304	**0.262**
	16	**0.335**	**0.198**	**0.297**	**0.266**	0.280	**0.364**	0.231	0.328	**0.287**
	64	**0.434**	0.287	0.391	0.384	0.363	0.467	0.349	0.421	0.387
	128	0.485	0.373	0.432	0.453	0.418	0.549	0.412	0.481	0.450
	256	0.523	0.456	0.462	0.516	0.467	0.622	0.467	0.537	0.506

Table 6. LPIPS of IL2-NeRF on IBRNet model, LLFF dataset on different combinations of summed loss weights. Note that a higher LPIPS indicates a more successful attack.

IBRNet DeepVoxels PSNR

	$\epsilon =$	Armchair	Cube	Greek	Vase	Avg.
NeRFool	8	9.500	13.982	**11.688**	11.437	11.652
IL2-NeRF	8	26.021	24.129	24.177	24.053	24.595
	16	16.594	20.286	17.644	20.118	18.661
	64	12.276	14.846	13.449	12.167	13.185
	128	9.754	12.968	12.391	**10.840**	**11.488**
	256	**8.660**	**11.829**	12.067	**10.235**	**10.700**

Table 7. PSNR of NeRFool vs. IL2-NeRF on IBRNet, DeepVoxels dataset. Note that a lower PSNR indicates a more successful attack.

IBRNet DeepVoxels SSIM

	$\epsilon =$	Armchair	Cube	Greek	Vase	Avg.
NeRFool	8	0.760	0.668	0.772	0.761	0.745
IL2-NeRF	8	0.971	0.936	0.953	0.933	0.948
	16	0.938	0.892	0.914	0.902	0.912
	64	0.855	0.729	0.820	0.790	0.799
	128	0.779	**0.626**	0.781	**0.725**	**0.728**
	256	**0.728**	**0.591**	**0.760**	**0.684**	**0.691**

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

ϵ.

Table 5 reports the SSIM value from running IL2-NeRF on variable ϵ across all three weight combinations. RGB-Loss achieves a lower average SSIM than both Density & Depth and Diff & Smooth at every ϵ. RGB-Loss also outperforms Density & Depth and Diff & Smooth on all scenes except Fortress at $\epsilon = 8, 16$.

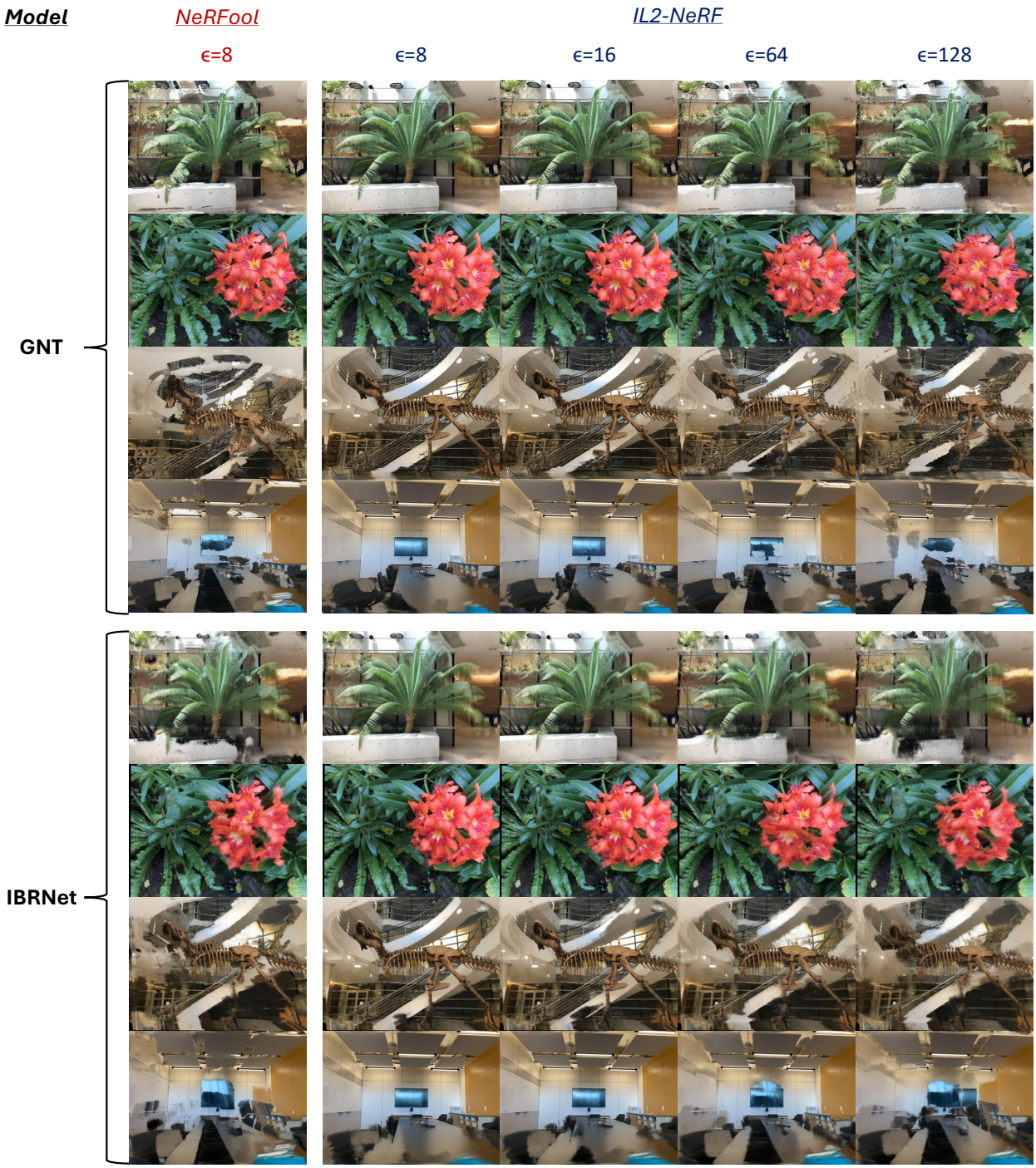

Figure 1. Visual comparing predicted images on four LLFF scenes from IBRNet and GNT on NeRFool and IL2-NeRF perturbed images on varying perturbation factors ϵ.

Table 6 reports the LPIPS value from running IL2-NeRF on variable ϵ across all three weight combinations. Here, Diff & Smooth achieves the highest average LPIPS across all scenes on $\epsilon = 8, 16$ and RGB-Loss achieves the highest average LPIPS on $\epsilon = 64, 128, 256$.

Overall, the original RGB-Loss results in the best attack metrics. We acknowledge there is future work in experimenting further with these loss weights and designing a loss that outperforms RGB-Loss for attacks on GNeRFs.

Visualizing Perturbations under L_∞ and L_2 Norms

IBRNet DeepVoxels LPIPS						
	$\epsilon =$	Armchair	Cube	Greek	Vase	Avg.
NeRFool	8	0.303	0.285	0.291	0.231	0.278
IL2-NeRF	8	0.072	0.054	0.068	0.084	0.070
	16	0.124	0.095	0.121	0.111	0.113
	64	0.226	0.234	0.251	0.217	0.232
	128	0.303	**0.332**	**0.296**	**0.265**	**0.299**
	256	**0.360**	**0.373**	**0.314**	**0.291**	**0.335**

Table 9. LPIPS of NeRFool vs. IL2-NeRF on IBRNet, DeepVoxels dataset. Note that a higher LPIPS indicates a more successful attack.

We provide a visual for comparing the same perturbation factor under the L_∞ and L_2 norm in Figure 2. L_2 attacks ensure uniform perturbation with less sharp neighboring-pixel differences, better resembling natural imaging artifacts. This is seen here, as under the same level of perturbation, IL2-NeRF presents smoother visuals perturbations across the pixels. Zooming in on the upper-right corner, we see how distortions between the two attacks differ, with IL2-NeRF exhibiting significantly less perceptible perturbations.

Visualizing Perturbations in LLFF Predictions We consider the immediate affects our perturbations have on GNeRF scene generation. Figure 1 compares the predicted images produced by IBRNet from varying ϵ. Here, we compare perturbations across different architectures by placing predicted images for the same four LLFF scenes (Fern, Flower, T-Rex, Room) for GNT and IBRNet.

Across model lines, one trend is obvious: as ϵ increases, the number of visible distortions in our generated images becomes more visible. However, the intensity of these distortions varies for different scenes and models. Notably, in the Flower scene in the second row, for both NeRFool $\epsilon = 8$ and IL2-NeRF $\epsilon = 128$ the image is more perturbed for IBRNet than GNT.

Visualizing Perturbations in LLFF Depth Masks We wish to compare differences in degradations across model lines by considering the GNT model. Figure 3 shows the depth mask for predicted scenes on the LLFF Orchids scene for both models.

Interestingly, across all values of ϵ, all GNT depth masks feature heavy artifacts on the edge of the image. This clearly contrasts IBRNet, whose perturbations are concentrated around the main object. When we increase ϵ for both models, more artifacts are added and intensified in the middle of the image. For both models, the depth mask for NeRFool $\epsilon = 8$ is most similar to IL2-NeRF at $\epsilon = 128$.

Visualizing Perturbations in DeepVoxels Predictions

We extend our analysis on visual perturbations by considering predicted images on the DeepVoxels dataset. Figure 4 compares the predicted images produced by IBRNet from varying ϵ across all four scenes of DeepVoxels. Each scene is depicted from top to bottom as follows: Armchair, Cube, Greek, and Vase. Unlike LLFF, DeepVoxels validation images consist of objects on a blank background. The lack of background makes perturbations more apparent.

At $\epsilon = 8$ and $\epsilon = 16$, IL2-NeRF, the scene predictions produce minimal distortions. These distortions are amplified at $\epsilon = 64$: most notably, the headrest on Armchair becomes larger, the two sides of Cube do not touch on their corners anymore, and Vase starts to move to the left. IL2-NeRF $\epsilon = 128$ and 256 are most similar to NeRFool $\epsilon = 8$, but we notice that there are larger white blotches for NeRFool, specifically in Armchair and Greek.

3. Ethical Consideration

In this section, we address any concerns about the negative impact our work creates. As our work explores a new threat model for GNeRF models, this paper is crucial in expanding a discussion for producing more robust GNeRFs and defensive techniques.

Neural radiance has seen success across many interdisciplinary fields. In robotics, NeRF models improve navigation and localization capabilities [1, 9, 11]. In autonomous driving, NeRF models have added a new dimension to panoptic image segmentation for 3D object tracking [4, 6, 12]. In Virtual Reality (VR), NeRF allows for real-time high-fidelity renderings of remote environments [2, 3, 7].

As in-field systems employ NeRF models, understanding the adversarial robustness of GNeRF models becomes crucial to understanding the potential of system performance being compromised. In general, machine learning models are vulnerable to perturbations injected into the source image [5, 8]. Discussing types of adversarial perturbations and their effects on GNeRF robustness is necessary to create secure scene-generation systems.

Our work is the first to address the performance of adversarial attacks on GNeRFs in the L_2 domain. The inception of IL2-NeRF elevates the discussion of GNeRF robustness to L_p perturbations, where $p \neq \infty$. This opens GNeRF security research to explore defensive techniques across different norms, which in turn produces resilient systems across different threat models and domains.

4. Open Science

We have made our repository available here. This repository contains the pipeline to run both NeRFool and IL2-NeRF attacks on both IBRNet and GNT provided their respective weights. IBRNet weights can be found here, and GNT weights can be found here. Both the IBRNet model code and the GNT model code are provided in our repository [14, 15].

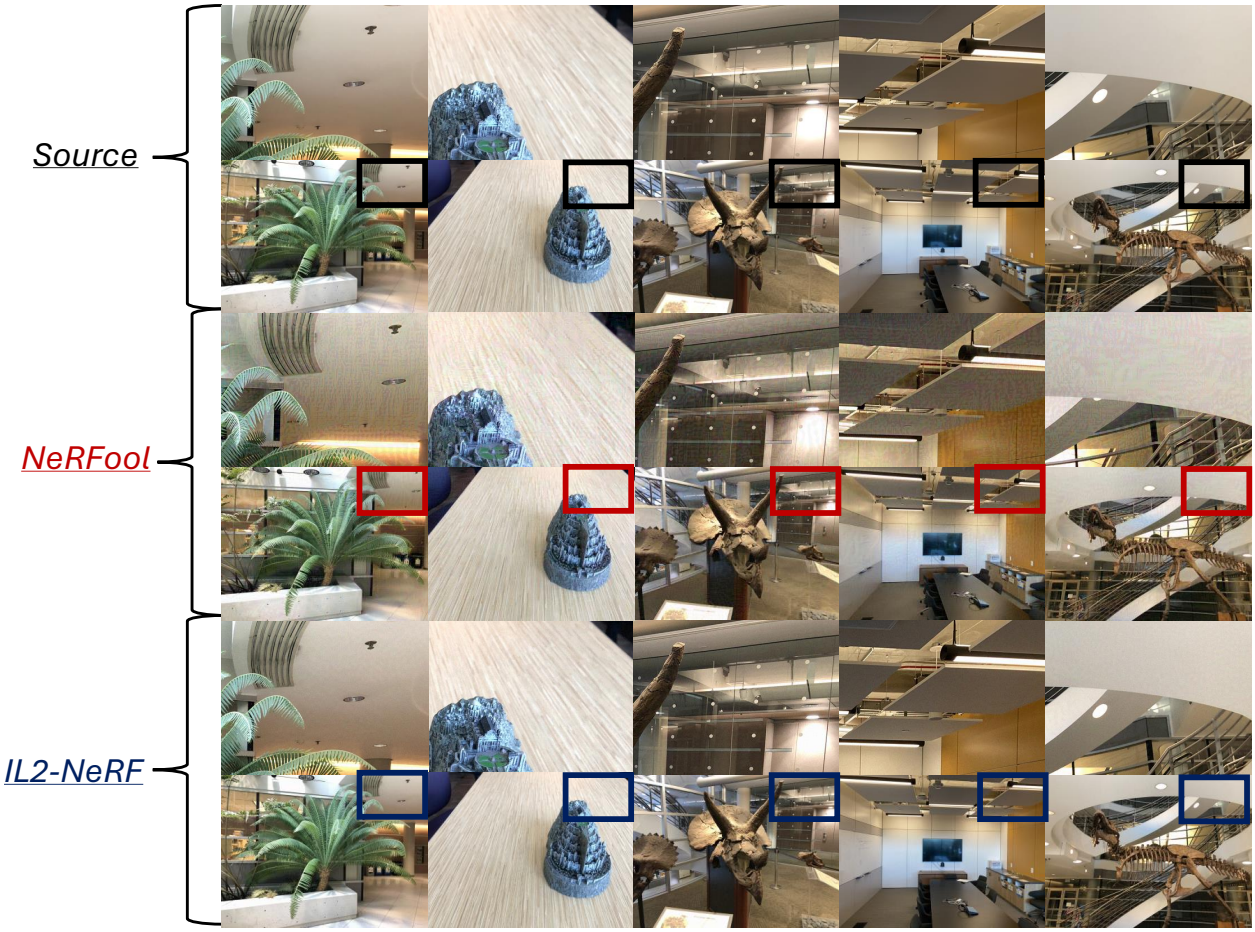

Figure 2. Visual comparing clean and perturbed source images from NeRFool and IL2-NeRFool $\epsilon = 8$ on Fern, Fortress, Horns, Room and T-Rex source images. When zooming into the right corner (these are the segmented 369×275 images from the boxes), L_∞ perturbations produced by NeRFool produce more distorted, jagged perturbations than the L_2 perturbations produced by IL2-NeRF under the same ϵ.

References

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2): 4606–4613, 2022. 6

[2] Peng Dai, Feitong Tan, Xin Yu, Yinda Zhang, and Xiaojuan Qi. Go-nerf: Generating virtual objects in neural radiance fields. *arXiv preprint arXiv:2401.05750*, 2024. 6

[3] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022. 6

[4] 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. 6

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 6

[6] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 6

[7] Chaojian Li, Sixu Li, Yang Zhao, Wenbo Zhu, and Yingyan Lin. Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2022. 6

[8] Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 6

[9] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo localization using neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*,

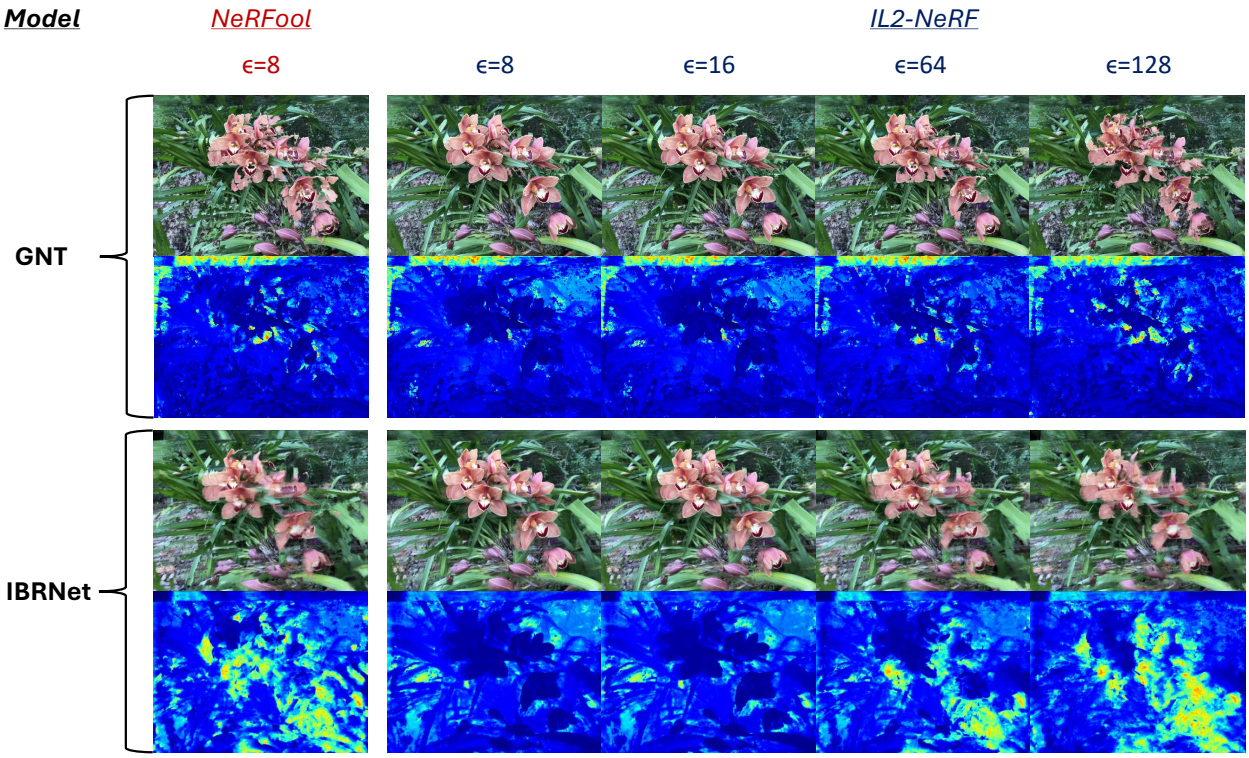

Figure 3. Visual comparing depth masks of Orchids predicted image from IBRNet and GNT on NeRFool and IL2-NeRF perturbed images on varying perturbation factors ϵ.

pages 4018–4025. IEEE, 2023. 6

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[11] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022. 6

[12] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 6

[13] 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. 2

[14] 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, 6

[15] 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. 6

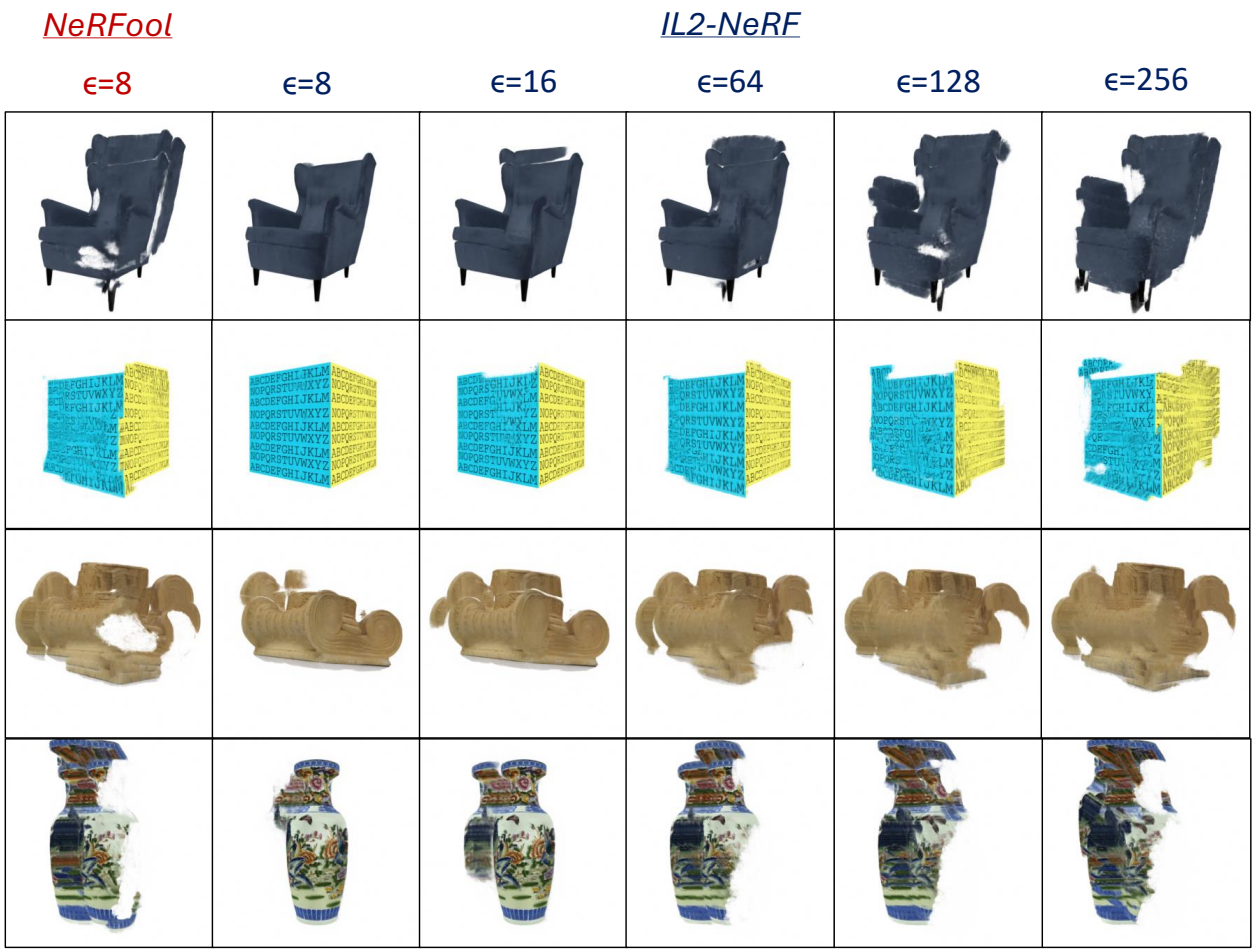

Figure 4. Visual comparing predicted images on all DeepVoxels scenes from IBRNet on NeRFool and IL2-NeRF perturbed images on varying perturbation factors ϵ.