NeRF Robustness Study Against Adversarial Bit Flip Attack

by

Zhou Yu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2023 by the
Graduate Supervisory Committee:

Deliang Fan, Chair
Chaitali Chakrabarti
Yanchao Zhang

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

Recently, there has been a notable surge in the development of generative models dedicated to synthesizing 3D scenes. In these research works, Neural Radiance Fields(NeRF) is one of the most popular AI approaches due to its outstanding performance with relatively smaller model size and fast training/ rendering time. Owing to its popularity, it is important to investigate the NeRF model security concern. If it is widely used for different applications with some fatal security issues would cause some serious problems. Meanwhile, as for AI security and model robustness research, an emerging adversarial Bit Flip Attack (BFA) is demonstrated to be able to greatly reduce AI model accuracy by flipping several bits out of millions of weight parameters stored in the computer's main memory. Such malicious fault injection attack brings emerging model robustness concern for the widely used NeRF-based 3D modeling. This master thesis is targeting to study the NeRF model robustness against the adversarial bit flip attack. Based on the research works the fact can be discovered that the NeRF model is highly vulnerable to BFA, where the rendered image quality will have great degradation with only several bit flips in the model parameters.

DEDICATION

This thesis is dedicated to my father and mother.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Deliang Fan at Arizona State University for his advising and mentoring. I am grateful for his remarkable suggestions and ideas, which helped form my research structure and directions. In addition, I sincerely thank the graduate advisory committee Dr. Chaitali Chakrabarti, Dr. Yanchao Zhang, for their support and valuable suggestions.

Also, thanks to all of my lab partners and their contributions to forming this thesis. I want to thank Zheng Zhang for introducing me to this nice lab, and I want to thank Yongjae Lee for the guide in explaining the NeRF details. And I want to thank Li Yang, Jintao Li, Xing Chen, and my other colleagues who have given their valuable expertise and suggestions in helping me throughout my master

Last but not least, I want to thank my parents for their continuous support during my master's studies.

This work is supported in part by the National Science Foundation under Grant No. 2019548 and No. 2342618

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

Deep Neural Networks (DNNs) have contributed great success in the computer vision field, like image classification, object detection, and speech recognition. Besides those areas, 3D scene modeling has become more and more important to attract researchers' interest. Those 3D scene modeling techniques can create entire scenes for imagery or animation for Automotive, Aerospace industries and Virtual reality (VR) and augmented reality (AR) use. On those 3D scene modeling techniques, Neural Radiance Fields (NeRF)(1) is one of the most efficient methods to generate a new view based on limited quantity images. With its increasing popularity, the model robustness study of Neural Radiance Fields(1) is quite important to ensure the security of NeRF. This thesis highlights the potential safety concern of NeRF against Adversarial Bit Flip Attack(2). The reason why chose the Adversarial Bit Flip Attack as the target attack method is its outstanding performance on the hamper DNN accuracy with extremely small model parameter noise injection. In the previous works, A. Rakin, et al. have shown that Bit Flip Attack can degrade ResNet 18 top-1 accuracy from 69.8% to 0.1% by flipping 13 bits out of 93 million bits stored in the computer's main memory. Although 3D scene modeling uses the Peak signal-to-noise ratio(PSNR) of rendering images to evaluate performance instead of classification accuracy, it is worth investigating the NeRF model's robustness against Adversarial Bit Flip Attack.

Chapter 2

BACKGROUND

2.1 Neural Radiance Fields (NeRF)

The 5D(x,y,z,θ, ϕ) neural radiance field encapsulates a scene's characteristics by detailing the density and emitted radiance in a particular direction at any given spatial point (1). When we compute the color of a ray traversing through the scene, we apply principles derived from traditional volume rendering techniques.

$$(RGB, \sigma) = F(x, y, z, \theta, \phi)$$

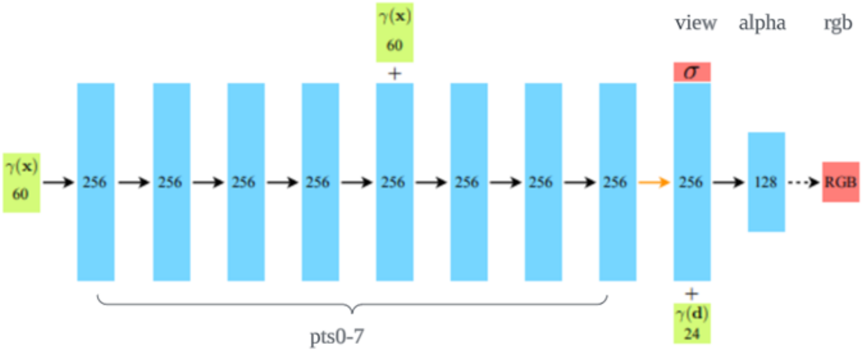


Figure 2.1: The Visualization of NeRF Architecture

The Nerf multi-layer perceptron contains 11 layers, taking the input direction y(x). It undergoes four fully connected layers and incorporates a skip connection that concatenates this input with the activation of the fifth layer. The first eight layers are dedicated to executing positional encoding information, denoted as points layer 0-7. Succeeding the points layers is the view layer, which concatenated input information on the viewing direction y(d) for training. This layer, characterized by the symbol σ, represents volume density and is processed by the alpha layer for density parameters. All the previously mentioned layers use the Rectified Linear Unit (ReLU) activation function. In contrast, the final layer, the RGB layer, uses the sigmoid activation function to generate the emitted RGB radiance at position x, as perceived by a ray with direction d.

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

The volume density can be conceptualized as the infinitesimal probability of a ray concluding its trajectory at a particle positioned at location x. The anticipated color of a camera ray is constrained in the near and far limits. The function T(t) represents the cumulative transmittance along the ray, extending from tn to tf. The volume density, denoted as $\sigma(x)$, can be construed as the infinitesimal probability of a ray terminating at a particle located at the position x. The camera ray function is denoted as r(t), where C(r) signifies the expected color, and d is the viewing direction. This function represents the cumulative transmittance along the ray's path from tn to t. To generate a view from our continuous neural radiance field, it is necessary to estimate this integral for a camera ray that traces through each pixel of the virtual camera as per the desired perspective.

2.2 Bit Flipped Attack (BFA)

The Bit-Flip Attack (BFA) is an adversarial technique designed to disrupt a Deep Neural Network (DNN) system by altering a small fraction of the weight parameters stored in Dynamic Random-Access Memory (DRAM) (3). This method leverages the row-hammer security exploit to manipulate specific bits in the DRAM, subsequently flipping their binary values by the gradient ranking. The main idea of Bit-Flip Attack (BFA) is flipping bits based on their gradients through the ascending rank of the DNN loss. Subsequently, this operation executes bits flipping by leveraging the inference loss of the Deep Neural Network (DNN) $\mathcal{L}$, resulting in the identification of perturbed bits. The operation can be simply written as follows:

$$\hat{b} = b + sign(\nabla_b \mathcal{L})$$

Since previous research(2) presents, deep neural networks exhibit vulnerability to adversarial examples due to their extreme linearity(4). Applying the Bit-Flip Attack can lead to significant misclassification of the model with a small perturbation of the binary representation. Previous experimental results(2) demonstrate its potency, where the top-1 accuracy of ResNet 18 drops to 0.1% by flipping a 13 out of 93 million bits in the ImageNet dataset(5).

Chapter 3

METHODOLOGY

3.1 Overview

This subsection introduces the methodology of how to perform Bit Flip Attack on Neural Radiance Fields (NeRF) multi-layer perceptron. Fig below shows the overview of the NeRF MLP attack frame. We quantize floating point NeRF MLP parameters to int8 format. The reason why we perform the attack on the quantized model, not the float model is that previous research(6) shows quantized models are robust to single-bit corruption. This phenomenon arises due to model quantization, a technique in which full-precision model parameters are substituted with low-bit-width integers or binary representations. This replacement substantially constrains the potential range of parameter values (60; 61).

After the quantized parameters were stored in DRAM. We can perform fault injection attacks to investigate the vulnerability of NeRF MLP parameters. This fault injection attack BFA is based on rowhammer(7) to flip the bits of the parameters stored in memory.

3.2 On Training Quantization

This subsection introduces the quantization method we used for the experiments. First, we use bi-linear layers to replace the original linear layers in MLP. The bi-linear layer can store the necessary information to do further steps. Then, train the new substitute MLP with bi-linear layers. At last, model quantization and weight quantization can be applied to substitute MLP. After those steps, we would get a

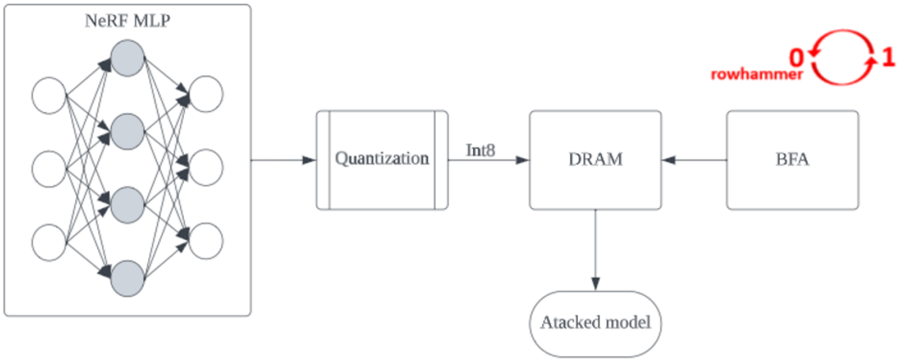


Figure 3.1: Overview of Applying Quantization and Perform Bit Flip Attack on DRAM

quantized NeRF model with a small performance deduction.

3.3 Adversarial Bit Flip Attack

This subsection introduces the algorithm of progressive bit search to locate the most vulnerable bits in the NeRF MLP and flip them in DRAM. Figure 3.2 shows the flowchart of this procedure. After the quantization conversion, the model is stored in DRAM in int8 format. The progressive bit search can be separated into two parts: in-layer search, and cross-layer search.

In layer search, this part of the algorithm is searching for n most vulnerable bits from $\mathbf{B}_l^{k-1}$ in target layers. The algorithm uses the input x and target t to calculate the gradients of bits and the DNN loss. Then rank those bits according to their gradients in descending order, the process can be written as:

$$\hat{b_l}^{k-1} = Top_{nb} \left| \nabla_{\hat{\mathbf{B}_l}^{k-1}} \mathcal{L}(f(x; \left\{ \hat{\mathbf{B}}_l^{k-1} \right\}_{l=1}^{L}), t) \right|$$

The next step involves applying Bit-Flip Attacks (BFA) to the bits that have been previously identified, resulting in new gradients and a modified loss. This altered gradient and loss serve as the evaluation metrics to gauge the increase in loss due to the BFA. Once the evaluation is complete, the information about the vulnerable bits is stored in the data profile, then the attacked bits are reverted to their original values for further calculations.

The cross-layer search conducts BFA across the entire network, particularly during the k-th iteration of the progressive bit search process. During this phase, the cross-layer search commences by independently executing in-layer searches on each layer, subsequently yielding a set of losses. The bits exhibiting the maximum loss in each layer are then identified. The ranking of this loss set reflects the order of the largest gradient changes resulting from the application (or non-application) of BFA. Succeeding this stage, the progressive bit search proceeds to the next iteration. Upon the completion of all iterations, the data profile should contain the top-n most vulnerable bits across the target layers.

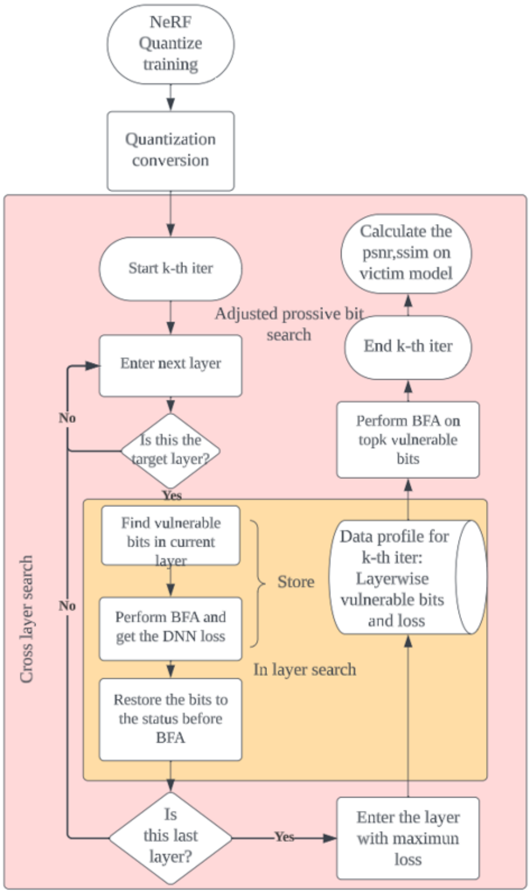

NeRF Quantize training
Quantization conversion
Start k-th iter
Calculate the psnr,ssim on victim model
Adjusted prossive bit search
End k-th iter
Enter next layer
No
Is this the target layer?
Perform BFA on topk vulnerable bits
Yes
Find vulnerable bits in current layer
Store
Data profile for k-th iter: Layerwise vulnerable bits and loss
Cross layer search
No
Perform BFA and get the DNN loss
In layer search
Restore the bits to the status before BFA
Is this last layer?
Yes
Enter the layer with maximun loss

Figure 3.2: Overall Flowchart of Applying Bit Flip Attack and How to Find the Most Vulnerable Bits

Chapter 4

EXPERIEMNT

4.1 Data Set

We are using NeRF Synthetic dataset(1) for training NeRF MLP. NeRF Synthetic dataset contains 400 images in the size of 800*800 for each scene. Those images have 8 scenes with the camera angle and camera transform matrices for each image. The experiment is mainly on the Lego scene, and the PyTorch NeRF code is a reference to Yen and Lin's work(8).

4.2 Experiment Goal

We want to find out if does Adversarial Bit Flip Attack also works on the image synthesis models rather than the classification model. If it works, how many bits are needed to flip to make large degradation? And, which layer is the most vulnerable layer? Those questions are the goals for designing the following experiments.

Chapter 5

RESULT

The results shown in the following tables are the average of 10 separate runs. And each run stops at PSNR around or below 10 or the result is saturated.

5.1 Quantization Result

After applying the quantization method to the NeRF MLP part, the result is shown in Figure 5.1. The PSNR drops 3.53, and the Structural Similarity Index (SSIM) drops 0.0104. The image quality does not have degradation based on evaluating PSNR and SSIM differences. However, the model size is about 1/4 of the original size, which matches the fact that the model is represented from float 32-bit numbers to integer 8-bit numbers.

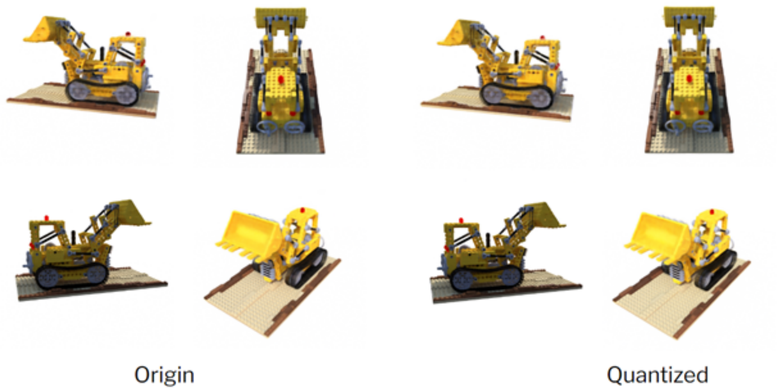
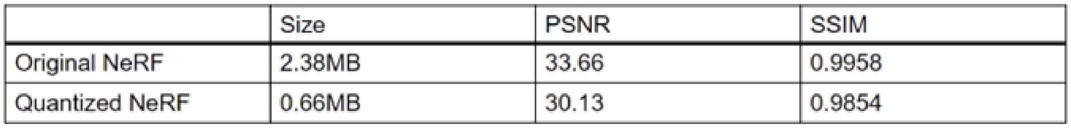


Figure 5.1: Figure of Quantization Result

5.2 Overall Vulnerability Analysis

In this section, we want to analyze the result of performing Bit Flip Attack on the entire Network. By comparing the result with the original output and the output



	Size	PSNR	SSIM
Original NeRF	2.38MB	33.66	0.9958
Quantized NeRF	0.66MB	30.13	0.9854

Table 5.1: Table of Quantization Result

from performing BFA on the entire network,(fig. 5.2 and table 5.2) we can easily find flipping 1 bit suffices to reduce the PSNR to 10 and SSIM to 0.5. It clearly shows NeRF MLP is vulnerable to BFA. Since NeRF MLP only contains linear layers, BFA has better performance when the target has extreme linearity. But when increasing the flipping bit numbers, the result is saturated.

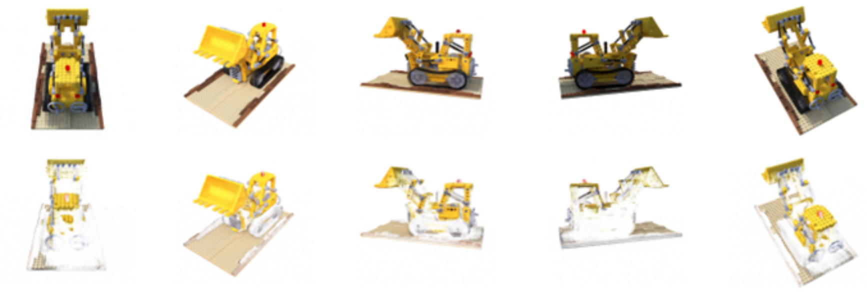
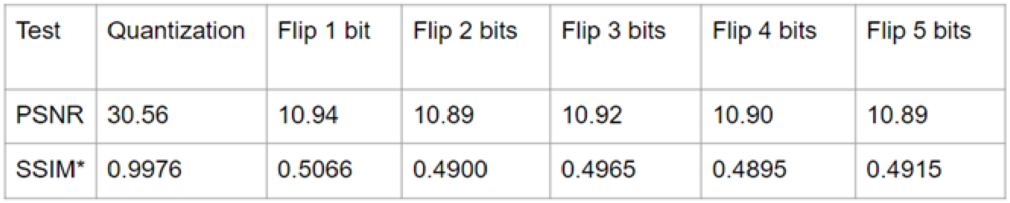


Figure 5.2: Figure of Apply Bit Flip Attack to All Layers



Test	Quantization	Flip 1 bit	Flip 2 bits	Flip 3 bits	Flip 4 bits	Flip 5 bits
PSNR	30.56	10.94	10.89	10.92	10.90	10.89
SSIM*	0.9976	0.5066	0.4900	0.4965	0.4895	0.4915

Table 5.2: Table of Bit Flip Attack on All Layers

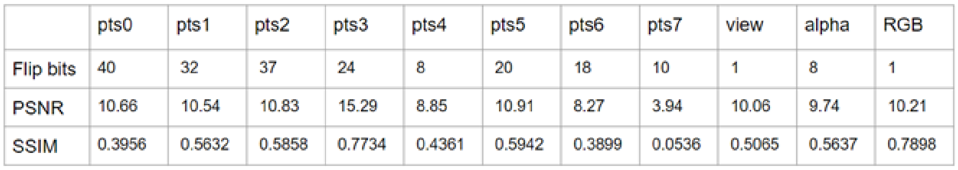
5.3 Layer-wise Vulnerability Analysis

The analysis of the results obtained after executing the Bit Flip Attack (BFA) on each layer yields intriguing insights. On the application of layer-wise BFA on the

NeRF MLP, the findings, as depicted in Figures 5.3 and 5.4, as well as Table 5.3, reveal noteworthy observations. Flipping bits on layers points. from 0 to points.4 manifests as the creation of shadows on the object's side. Despite the uniform reduction of PSNR around 10, the difference in SSIM indicates differing levels of impairment. Specifically, performing BFA on layers points. from 0 to points.4 appears to impact the model in some way.

Flipping bits on layers points.5 to points.7 results in changes to the object's surroundings, particularly affecting the plane on which the object is positioned. Notably, the result for points.7 shows that flipping 10 bits can significantly disrupt structural similarity (SSIM reduced to 0.05) with a PSNR of 3.94.

Figure 5.4 illustrates that flipping just 1 bit on the view layer and RGB layer is sufficient to decrease the PSNR to 10. This result suggests that the view layer and RGB layer exhibit similar vulnerabilities in resisting noise from BFA. However, performing BFA on the RGB layer has a less pronounced impact on SSIM (reduced to 0.79) compared to the view layer results (SSIM reduced to 0.51). The view layer result also matches the overall BFA result, which proves the correctness of the algorithm. These findings underscore the distinct sensitivities of different layers to Bit Flip Attacks and provide valuable insights into their vulnerabilities and implications on model performance metrics.



	pts0	pts1	pts2	pts3	pts4	pts5	pts6	pts7	view	alpha	RGB
Flip bits	40	32	37	24	8	20	18	10	1	8	1
PSNR	10.66	10.54	10.83	15.29	8.85	10.91	8.27	3.94	10.06	9.74	10.21
SSIM	0.3956	0.5632	0.5858	0.7734	0.4361	0.5942	0.3899	0.0536	0.5065	0.5637	0.7898

Table 5.3: Table of Bit Flip Attack on Single Layer

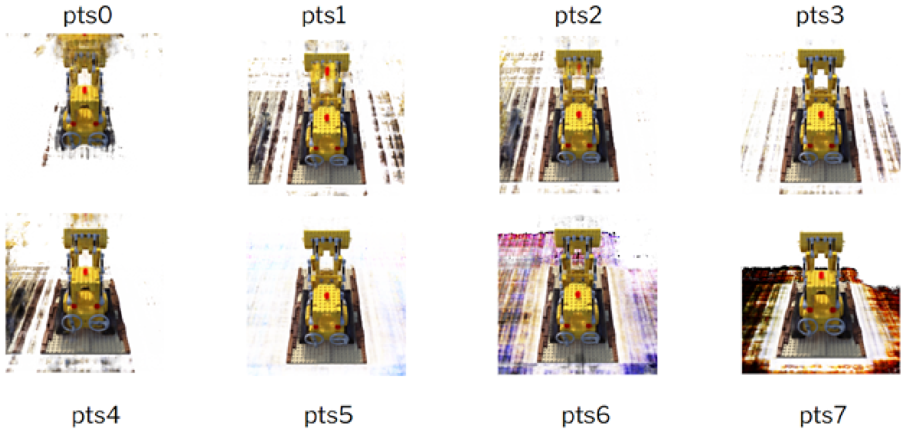


Figure 5.3: Figure of Bit Flip Attack on Single Layer from points 0 to points 7 layer

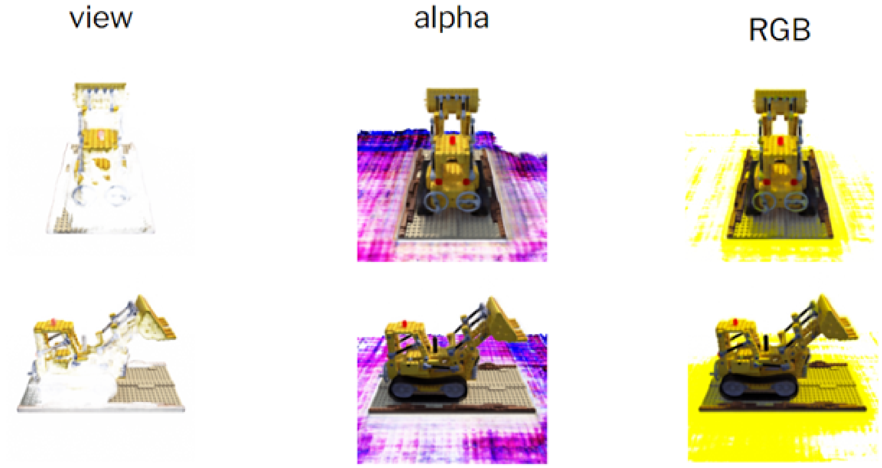


Figure 5.4: Figure of Bit Flip Attack on Single Layer for view, alpha, and RGB layer

Chapter 6

DISCUSSION AND FUTURE

6.1 Conclusion and Discussion

Our experiments provide interesting observations for the vulnerability of the NeRF model. First, we discovered that the view layer is the most vulnerable layer in NeRF MLP. Second, if the attacker wants to focus on reducing the structure similarity, attacking the last layer of the positional encoding layer (points.7) is more effective.

6.2 Future and Extension

Based on this thesis study, there are many interesting findings to extend. While attacking positional coding layers, it would create some shadow objects on the side of scenes. It might be able to design a specific attack creating confusing noise around the object. Flipping one bit in the view layer is enough to make a really large degradation of PSNR. We want to verify how this approach performs in complex NeRF, like pixel NeRF(9), GNeRF(10), etc. If we can find the same kind of vulnerable point in complex NeRF, then it becomes a serious issue.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” 2020.

[2] A. S. Rakin, Z. He, and D. Fan, “Bit-flip attack: Crushing neural network with progressive bit search,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[3] F. Yao, A. S. Rakin, and D. Fan, “{DeepHammer}: Depleting the intelligence of deep neural networks through targeted chain of bit flips,” in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1463–1480, 2020.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” Mar 2015.

[5] A. K. G. Inc, A. Krizhevsky, G. Inc, G. I. Profile, I. S. G. Inc, I. Sutskever, G. E. H. OpenAI, G. E. Hinton, OpenAI, O. Profile, and et al., “Imagenet classification with deep convolutional neural networks,” Jun 2017.

[6] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Quantized neural networks: Training neural networks with low precision weights and activations,” *arXiv.org*, Sep 2016.

[7] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, “Flipping bits in memory without accessing them: An experimental study of dram disturbance errors,” *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, 2014.

[8] L. Yen-Chen, “Nerf-pytorch,” *GitHub repository*, 2020.

[9] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “Pixelnerf: Neural radiance fields from one or few images,” May 2021.

[10] Q. Meng, A. Chen, H. Luo, M. Wu, H. Su, L. Xu, X. He, and J. Yu, “Gnerf: Gan-based neural radiance field without posed camera,” Aug 2021.