Unifying the Harmonic Analysis of Adversarial Attacks and Robustness: Supplementary Material

Shishira R Maiya[1]
shishira@umd.edu
Max Ehrlich[1]
maxehr@umd.edu
Vatsal Agarwal[1]
vatsalag@umd.edu
Ser-Nam Lim[2]
sernamlim@meta.com
Tom Goldstein[1]
tomg@umd.edu
Abhinav Shrivastava[1]
abhinav@cs.umd.edu

[1] University of Maryland
[2] Meta

Note: We also include perturbation gradient results across datasets and architectures in the supplementary zip file, which can be easily navigated using index.html file provided.

A Appendix

A.1 Proofs

Here are the proofs for some results from above. In equation (13) we mentioned

$$\nabla_{\delta} Y = \nabla_{x} Y = \nabla_{\hat{x}} Y \tag{1}$$

Consider a neural network $y = h(x; \theta)$. Let the adversarial sample be $\hat{x} = x + \delta$, where δ is the additive adversarial noise.

$$y = h(\hat{x}) = h(x+\delta) \tag{2}$$

$$\frac{dy}{dx} = h(x+\delta)' \cdot 1 = \frac{dy}{d\delta} \tag{3}$$

$$\frac{dy}{d\hat{x}} = h(\hat{x})' = h(x+\delta)' \text{ hence} \tag{4}$$

$$\frac{dy}{d\hat{x}} = \frac{dy}{d\delta} = \frac{dy}{dx} \text{ or } \nabla_{\delta} Y = \nabla_{x} Y = \nabla_{\hat{x}} Y \tag{5}$$

In the same section's equation (14) we also mentioned $\nabla_x L \propto \nabla_x Y$.

$$\text{Let } L = \frac{1}{2}\left(h(x;\theta) - \hat{y}\right)^2 \text{ be the loss.} \tag{6}$$

$$\frac{dL}{dx} = \left(h(x;\theta) - \hat{y}\right) \cdot h(x;\theta)' \tag{7}$$

$$\text{here } h(x;\theta)' = \frac{dy}{dx} \text{ and } \left(h(x;\theta) - \hat{y}\right) \text{ is a constant} \tag{8}$$

$$\frac{dL}{dx} = K \cdot \frac{dy}{dx} \text{ which implies} \tag{9}$$

$$\nabla_x L \propto \nabla_x Y \tag{10}$$

A.2 Training Details

We utilize ResNet-18 in all our experiments (unless stated otherwise). For ImageNet and TinyImageNet datasets, we train for a total of 100 epochs, with an initial learning rate of 0.1 decayed every 30 epochs, momentum of 0.9 and a weight decay of 5e-4. In Madry adversarial training for the same, we use an ε value of 4/255. Under adversarial training for free setting, we train both models for 25 epochs with learning rate decayed every 8 epochs and the m (repeat step) set to 4.

For CIFAR-10, we train the model for total of 350 epochs, starting with a learning rate of 0.1, decayed at 150 and 250 epochs and use the same setting with an ε of 8/255 for Madry training. In adversarial training for free setting, we train the model for 100 epochs with learning rate decay every 30 epochs and the m value set to 8.

We utilize the pretrained models provided by PyTorch for ImageNet normal models. All experiments involving ImageNet-based adversarial training were done using Adversarial training for free method, with total epochs of 25 and m value set to 4.

A.3 Frequency Range-Based Perturbations

We revisit the results shown in Figure 4 and show the same in a broader sense by attacking different frequency ranges. The results under DCT-PGD based Auto-Attack are shown in Figure **??**. We can see that the trends which were observed and discussed in earlier sections remain unchanged.

A.4 What do frequency attacks target ?

A natural question that might arise with respect to DCT-PGD paradigm is how can we be sure that there is proportionate distortion in the frequency space as well. We can visualize this using simple properties of the DCT. Consider the 1-D DCT from above. Since it is a linear transform, we can rewrite it as :

$$D(z) = WZ \text{ where W is the linear DCT transform on the input tensor Z} \tag{11}$$

$$\hat{x} = x + \delta \text{ in DCT space becomes} \tag{12}$$

$$W \cdot \hat{x} = W \cdot x + W \cdot \delta \tag{13}$$

$$\tag{14}$$

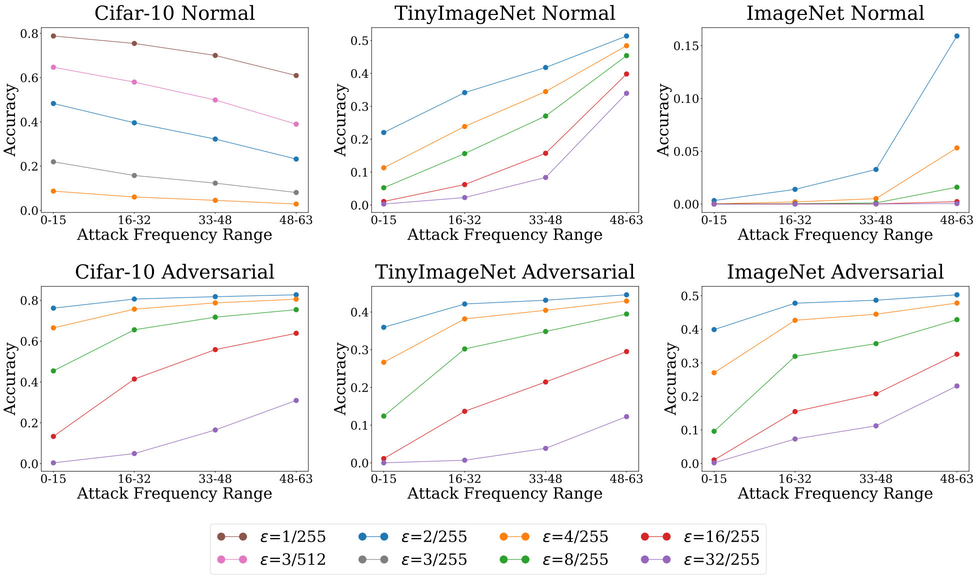


Figure A.1: Extension to experiments shown in Figure 4. DCT-PGD Auto-Attack across different frequency ranges. Note that for CIFAR-10 Normally trained model, we have shown the results with slightly lower epsilons.

The elements of W represent different standard DCT basis functions, such that the lower frequencies are in upper left corner and the higher frequencies are towards the lower right corner. For any element i that also represents a frequency component, we can say that:

$$W_i \cdot \hat{x}_i = W_i \cdot x_i + W_i \cdot \delta_i \tag{15}$$

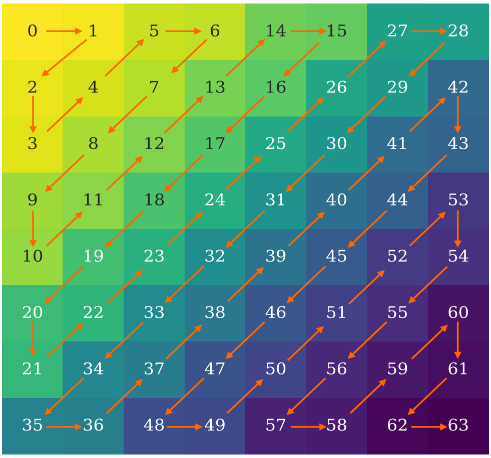
Essentially, we see that in the frequency space, each component of the resulting adversarial example $\hat{x}$ is linearly distorted by the corresponding frequency component of noise δ.

A.5 Does model architecture influence perturbation gradients?

The input images or the dataset is still only one part of $D(\nabla_\delta Y)$, with the model being the other part. We analyzed the effects of different datasets with the measure, but there is a possibility that differing model architecture can also influence it. We ran the same experiments using non-ResNet style architectures like DenseNet-121, ViT and VGG-16 on ImageNet. The results are shown in Figure **??**, and we can see that there is no deviation in trends.

A.6 Does Image Size Matter?

To confirm that the anomalies of adversarial examples are indeed due the underlying dataset and not just the size, we repeat the experiment by training models where ImageNet and Tiny-Imagenet images are resized to smaller sizes using bicubic filter. The average Perturbation gradients calculated from these models are shown in Figure **??**. The trends tell us that the anomalous properties exhibited by CIFAR are not merely due to their size.



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure A.2: (b) shows the standard 8×8 DCT block with the all 64 frequencies arranged in zigzag order.

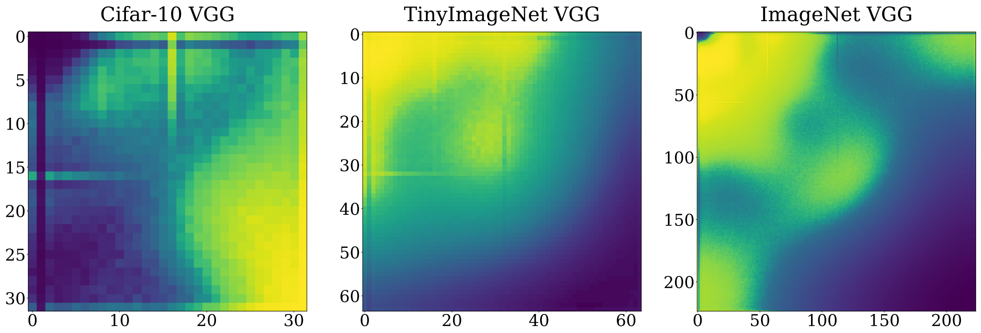


Figure A.3: Average Perturbation gradients of VGG-16 models, across datasets

A.7 Extending to more datasets

We also repeat the experiments across datasets, including non-ImageNet derived datasets like MNIST, Fashion-MNIST and CIFAR-100.The results are shown in Figure **??**.

A.8 Effect of Auto-Attack

We calculate and plot the Perturbation gradients for all models under Auto-attack setting. In general, there appears to be no significant difference when compared to results from PGD attack. The results are shown in figure **??**.

A.9 Class wise Results

We also investigate if there exists different frequency distribution in perturbation gradients for each class in a dataset. We show these results for CIFAR-10 (Fig **??**), MNIST (Fig **??**) and Fashion-MNIST (Fig **??**) datasets, for both normally trained and adversarially trained

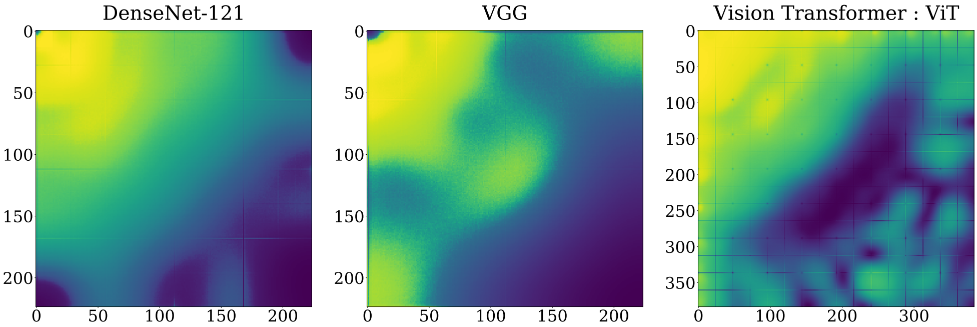
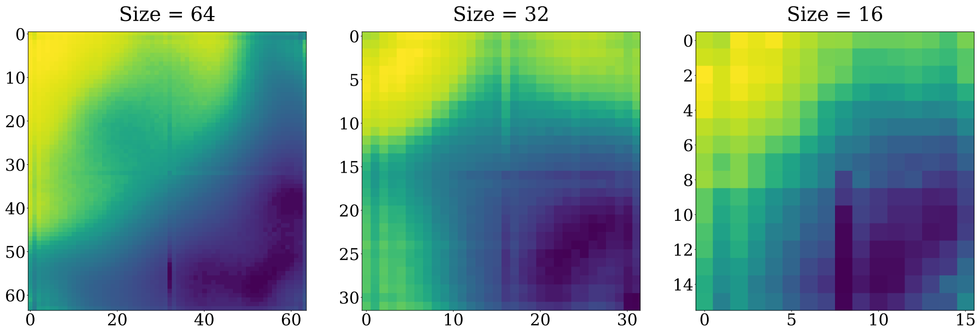


Figure A.4: DCT of Perturbation gradients for different architectures. All models were trained on ImageNet.

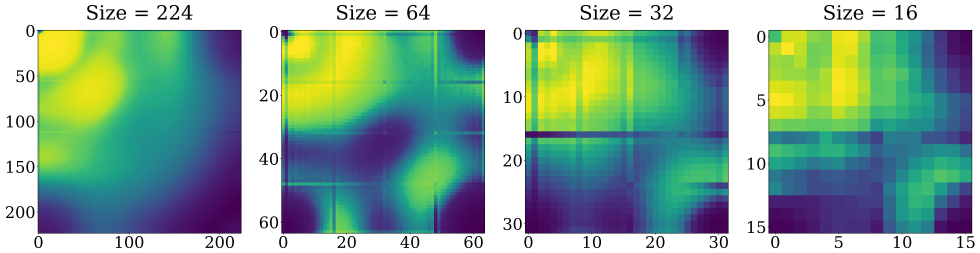
models. Apart from subtle differences, we do not see any general shift in the trends and observations. Results for CIFAR-100 and ImageNet are included along with the supplementary zip folder and can be easily visualized using the “index.html" file.

A.10 Examples of Frequency-Based Perturbations

We show example images under different perturbation budgets of L_∞ norm, across datasets in Figures **??**, **??** and **??**. We also show examples of images when certain frequency bands are dropped **??** and the complementary case of including only specified frequencies **??**.



Resized TinyImageNet



Resized ImageNet

Figure A.5: Effect of resizing the images on TinyImageNet and ImageNet.

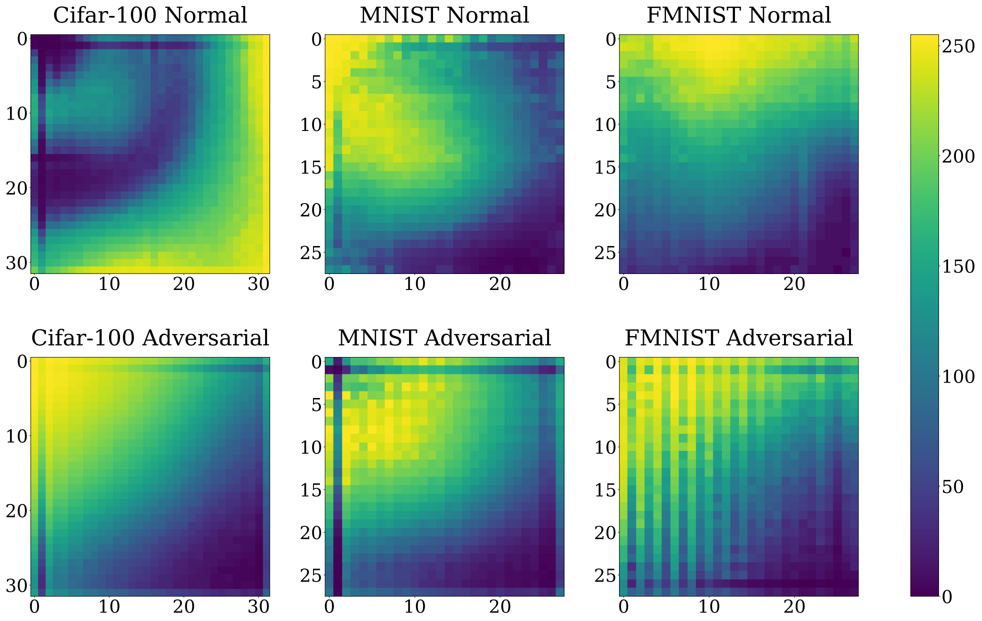


Figure A.6: DCT of Average Perturbation gradients across additional datasets

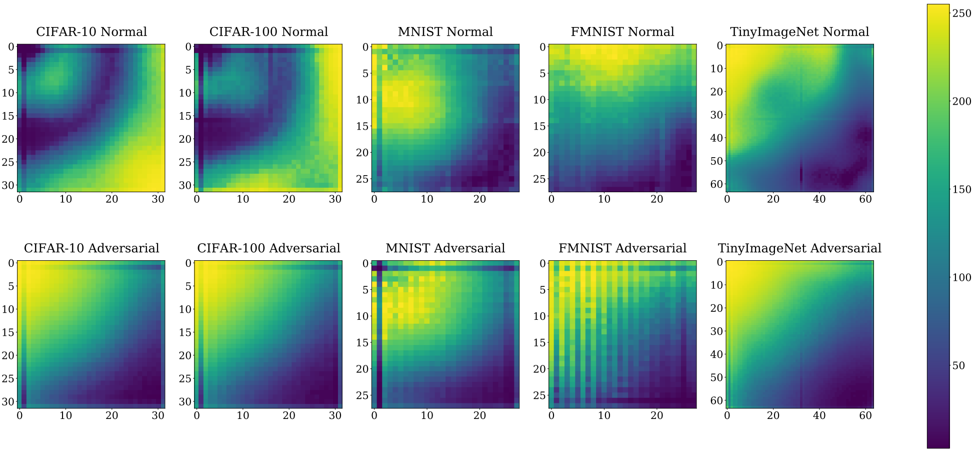


Figure A.7: DCT of Average Perturbation gradients with Auto-Attack

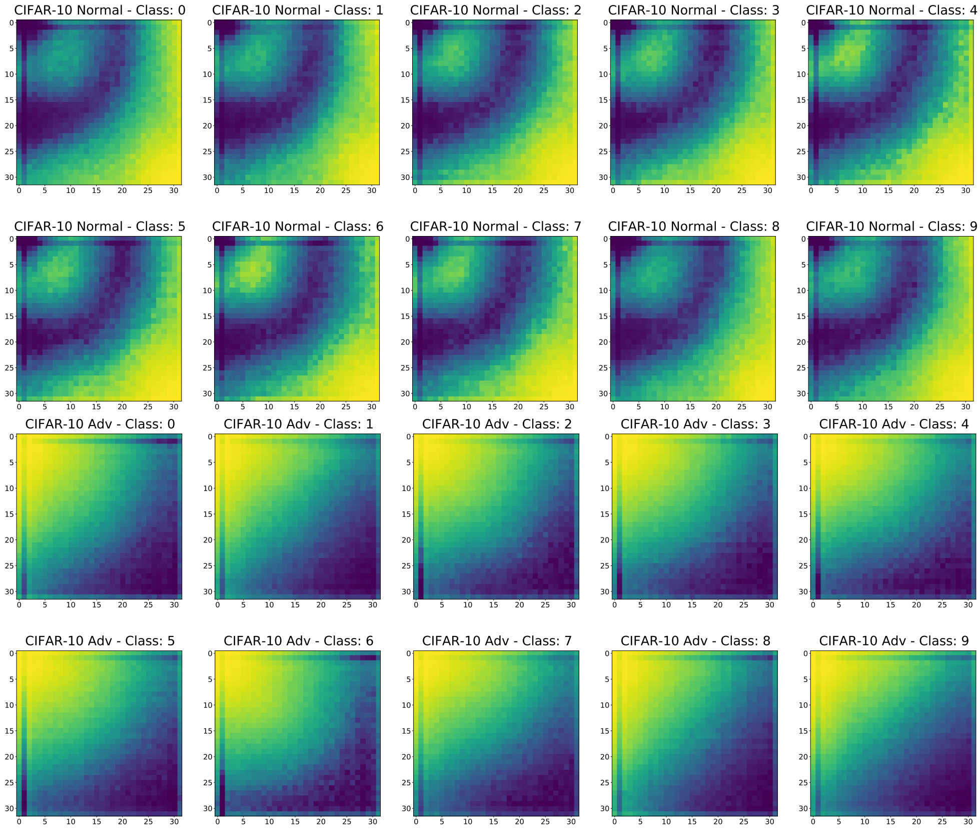


Figure A.8: DCT of Average Perturbation gradients Classwise for CIFAR-10

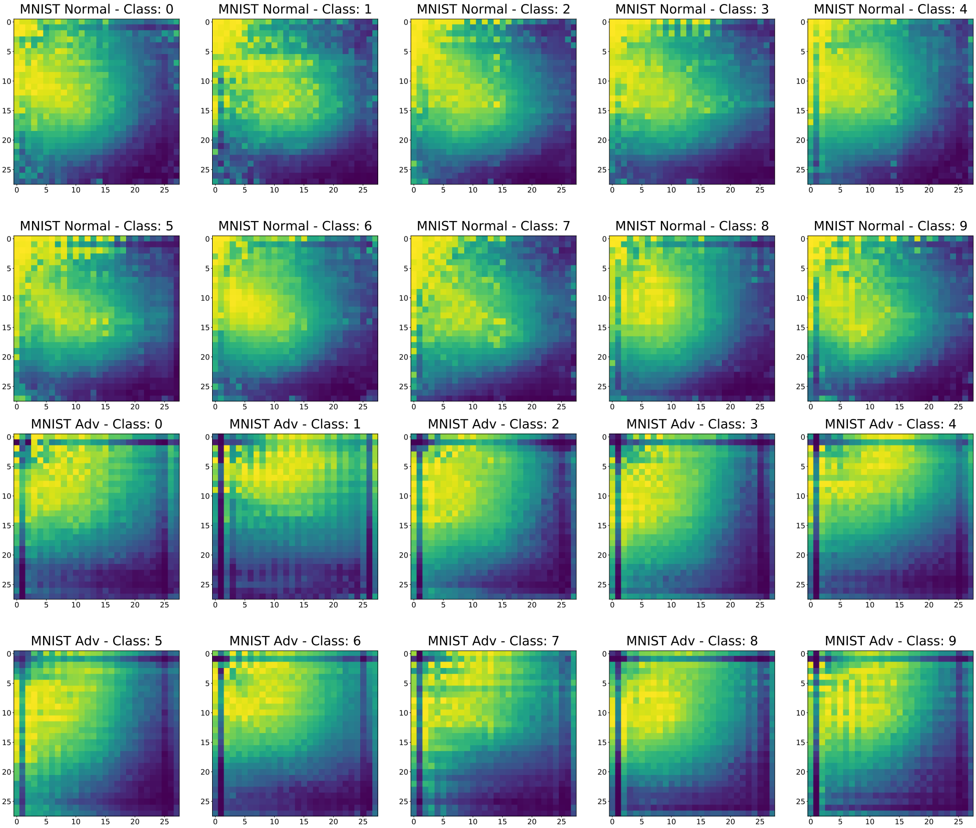


Figure A.9: DCT of Average Perturbation gradients Classwise for MNIST

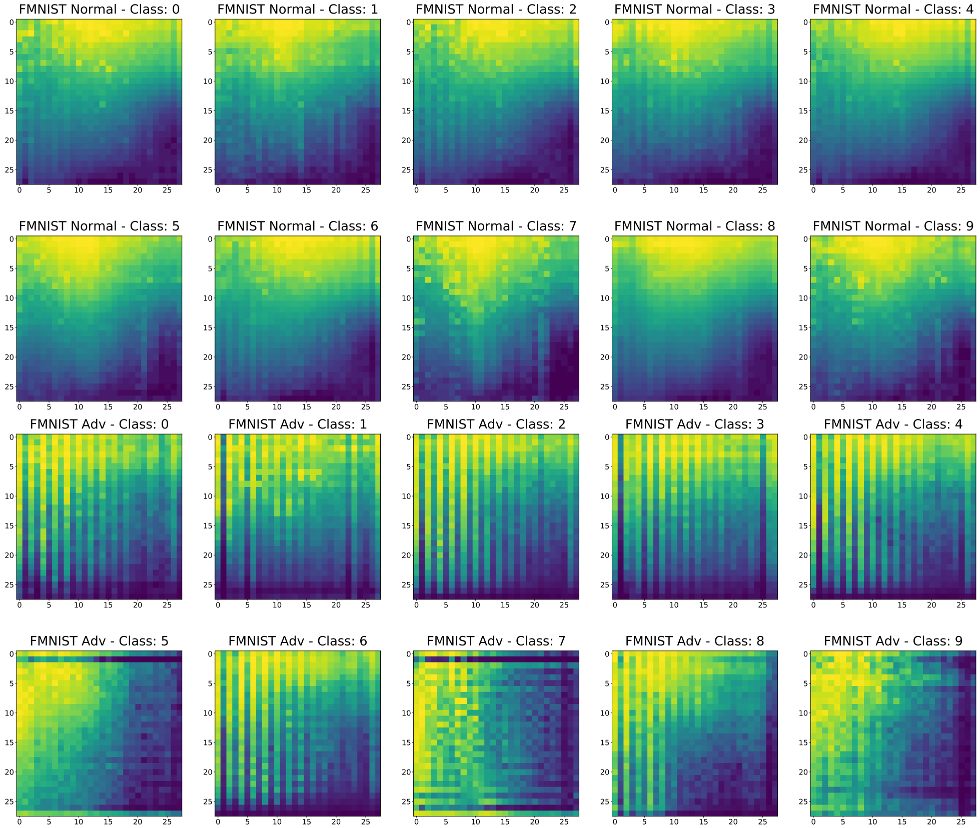


Figure A.10: DCT of Average Perturbation gradients Classwise for Fashion-MNIST

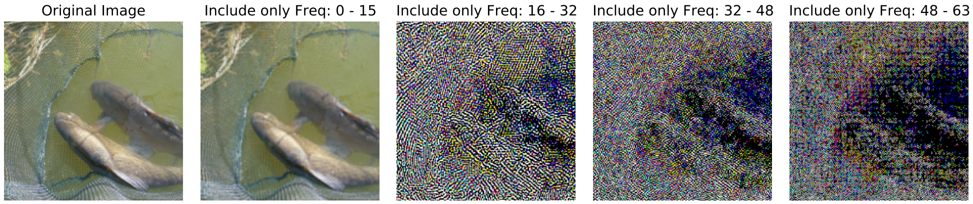

Original Image
Include only Freq: 0 - 15
Include only Freq: 16 - 32
Include only Freq: 32 - 48
Include only Freq: 48 - 63

Figure A.11: ImageNet examples where image is reconstructed using only specified frequency bands

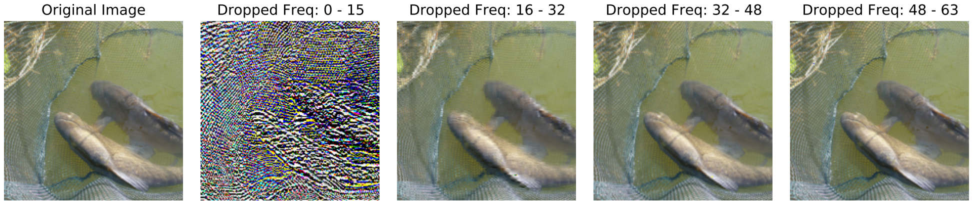
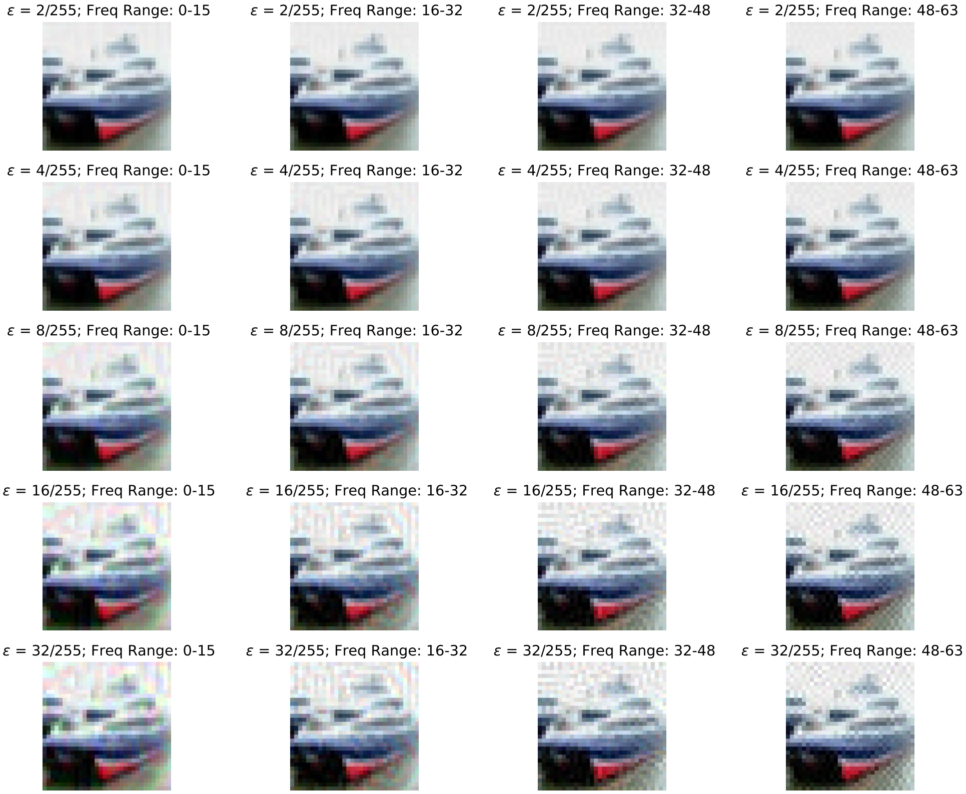
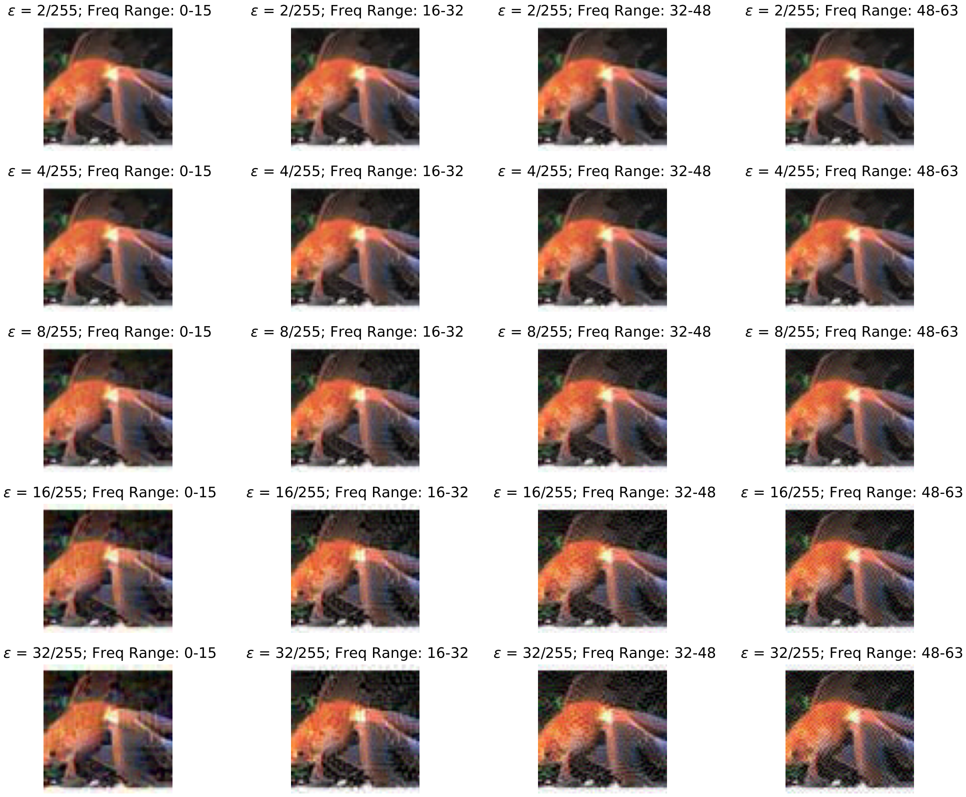


Figure A.12: ImageNet examples where image is reconstructed after dropping (zeroing) certain frequency bands



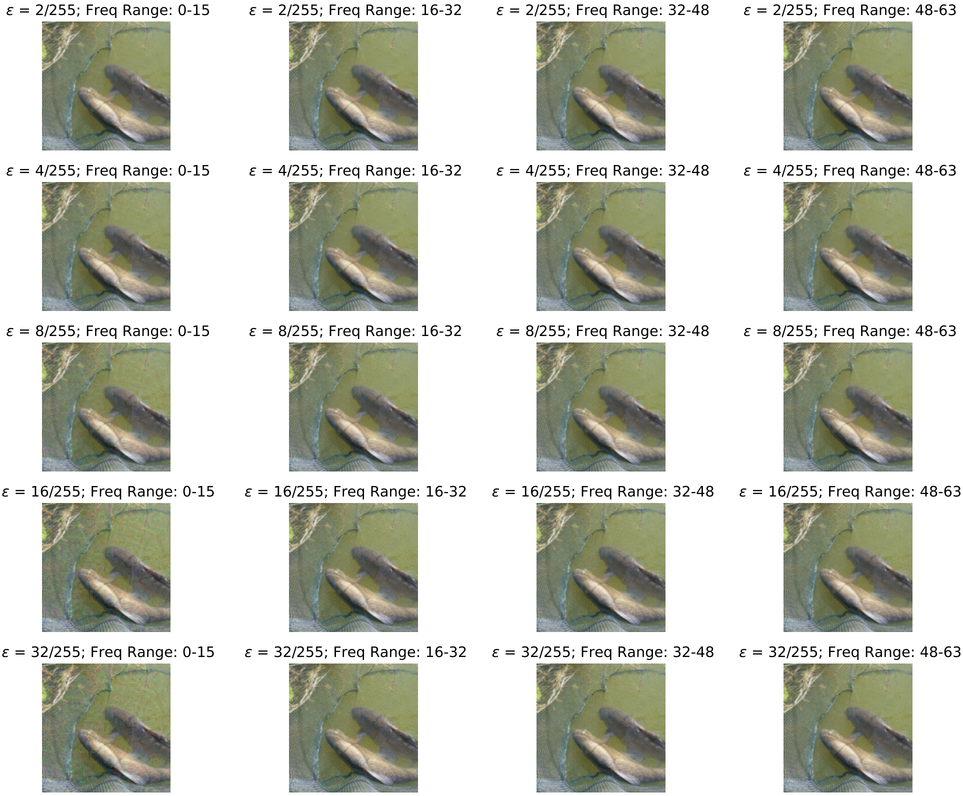
ε = 2/255; Freq Range: 0-15
ε = 2/255; Freq Range: 16-32
ε = 2/255; Freq Range: 32-48
ε = 2/255; Freq Range: 48-63
ε = 4/255; Freq Range: 0-15
ε = 4/255; Freq Range: 16-32
ε = 4/255; Freq Range: 32-48
ε = 4/255; Freq Range: 48-63
ε = 8/255; Freq Range: 0-15
ε = 8/255; Freq Range: 16-32
ε = 8/255; Freq Range: 32-48
ε = 8/255; Freq Range: 48-63
ε = 16/255; Freq Range: 0-15
ε = 16/255; Freq Range: 16-32
ε = 16/255; Freq Range: 32-48
ε = 16/255; Freq Range: 48-63
ε = 32/255; Freq Range: 0-15
ε = 32/255; Freq Range: 16-32
ε = 32/255; Freq Range: 32-48
ε = 32/255; Freq Range: 48-63

Figure A.13: CIFAR-10 example images under different attack settings.



ε = 2/255; Freq Range: 0-15
ε = 2/255; Freq Range: 16-32
ε = 2/255; Freq Range: 32-48
ε = 2/255; Freq Range: 48-63
ε = 4/255; Freq Range: 0-15
ε = 4/255; Freq Range: 16-32
ε = 4/255; Freq Range: 32-48
ε = 4/255; Freq Range: 48-63
ε = 8/255; Freq Range: 0-15
ε = 8/255; Freq Range: 16-32
ε = 8/255; Freq Range: 32-48
ε = 8/255; Freq Range: 48-63
ε = 16/255; Freq Range: 0-15
ε = 16/255; Freq Range: 16-32
ε = 16/255; Freq Range: 32-48
ε = 16/255; Freq Range: 48-63
ε = 32/255; Freq Range: 0-15
ε = 32/255; Freq Range: 16-32
ε = 32/255; Freq Range: 32-48
ε = 32/255; Freq Range: 48-63

Figure A.14: TinyImageNet example images under different attack settings.



ε = 2/255; Freq Range: 0-15
ε = 2/255; Freq Range: 16-32
ε = 2/255; Freq Range: 32-48
ε = 2/255; Freq Range: 48-63
ε = 4/255; Freq Range: 0-15
ε = 4/255; Freq Range: 16-32
ε = 4/255; Freq Range: 32-48
ε = 4/255; Freq Range: 48-63
ε = 8/255; Freq Range: 0-15
ε = 8/255; Freq Range: 16-32
ε = 8/255; Freq Range: 32-48
ε = 8/255; Freq Range: 48-63
ε = 16/255; Freq Range: 0-15
ε = 16/255; Freq Range: 16-32
ε = 16/255; Freq Range: 32-48
ε = 16/255; Freq Range: 48-63
ε = 32/255; Freq Range: 0-15
ε = 32/255; Freq Range: 16-32
ε = 32/255; Freq Range: 32-48
ε = 32/255; Freq Range: 48-63

Figure A.15: ImageNet example images under different attack settings. Notice that the “perceptibility" is not affected due to frequency based attacks.