Appendices

A. Constraint-based attack

Constraint-based technique is designed based on equation (3). This technique tries to minimize the distortion while constraints are responsible for increasing FLOps. Constraint-based technique can be time or resource consuming because it is an optimization technique with non-linear constraints. Because of that reason, we use a smaller input size. We send perturbation values as the input of the optimization technique. We only perturb selected pixels to keep the input size low. Constraint-based technique only perturbs pixels in the border of the image. Changing only border pixels makes sure that the main object of the image does not get perturbed. We divide the technique into four steps.

Initializing Output. This step represents initializing the output of the optimization function *i.e.,* resultant perturbation. The optimization with non-linear constraints may find itself in a local minima without proper Initialization. Initialization is done by randomly generating 100 values for distortion and selecting the value which generates the highest FLOPs increase and has the lowest absolute value.

Creating the constraints. In the equation (3), the constraint is designed to keep all α values $True$. But with our observations from results, we notice that all the blocks can not be made active simultaneously for every image Specially for a few images, the behaviours of two blocks are totally opposite, *i.e.,* making one block active makes other inactive. Because of this reason, instead of making all the blocks active, we have tried to make a certain number of blocks active. If N_G represents the number of active blocks in the model, then the constraint will be $N_G >= G_{Max}$ where G_{Max} is the predefined desired number of active blocks.

Creating Minimization Function. Minimization function has been designed to minimize the difference between the original and attacked image. For Constraint-based attack, we minimize the difference between pixel values of the original and distorted image.

Calling Optimization Function. The optimization function for Constraint-based perturbation attack can be represented as,

$$minimize(||x - x'||) \tag{6}$$

such that,

$$(N(x') > G_{Max})$$

Where, N is a method that returns number of active gates, G_{Max} is desired active number of gates, x' is the attacked image with changed borderline pixel values, x is image input. We have used Sequential Least Squares Programming (SLSQP) as the optimization algorithm.

B. Quality Measurement Algorithms

Canny Edge Detection (CED) is a method to detect edges in an image. Pixels whose value is significantly different from their neighbors are called edges. We use edge detection as a metric to measure quality by analyzing the change in the number of edges in the image after the attack. Canny edge detection follows certain steps to detect edges. After reducing the noise from the image, CED calculates intensity gradients by convolving the image with predefined Sobel filters. We get the information about changes in values over pixels. Next, edge thinning is performed by suppressing the gradient, and finally, strong edges (higher gradient value) are selected to represent the edges of the image.

Fourier Transform is used to convert signals from spatial or temporal domain to the frequency domain. Fast Fourier Transform (FFT) is an algorithm to perform Fourier Transform with lower time complexity. Converted spatial frequency domain can be useful in giving useful information regarding image features. The rough shape structure of the image generates low spatial frequency after FFT. High spatial frequency corresponds to detailed features [17].

C. Euclidean distances between FFTs

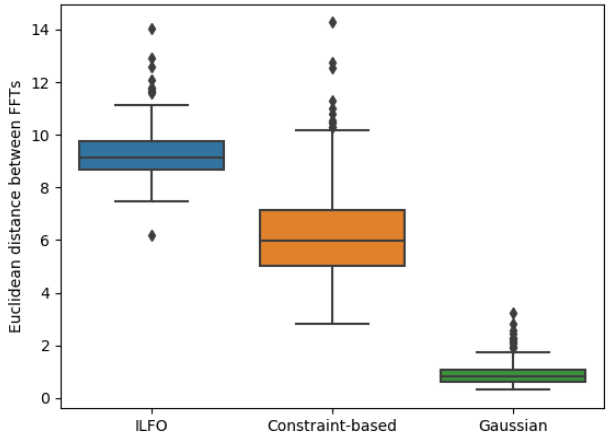
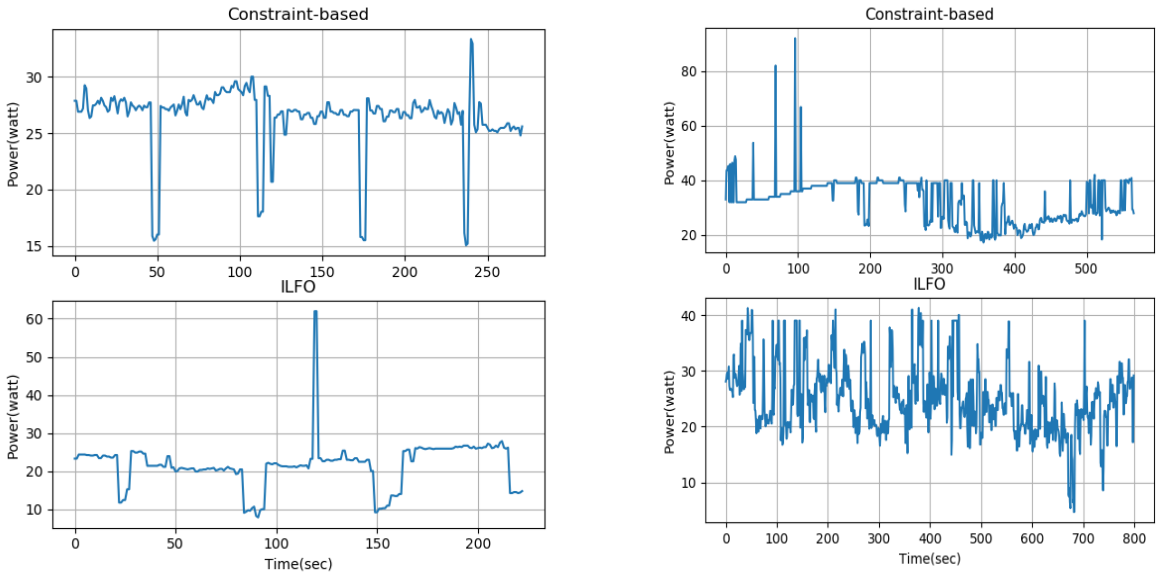


Figure 9. Euclidean distances between FFTs of generated and perturbed images

D. CPU power consumption of the techniques

The results for CPU power consumption of different attack techniques is shown in Figure 10.



CIFAR-10 **ImageNet**

Figure 10. CPU power consumption of the attack techniques

Original Gaussian Noise Constraint-Based ILFO

Figure 11. Effect of different attacks on ImageNet dataset images

Figure 12. Effect of different attacks on CIFAR-10 dataset images.

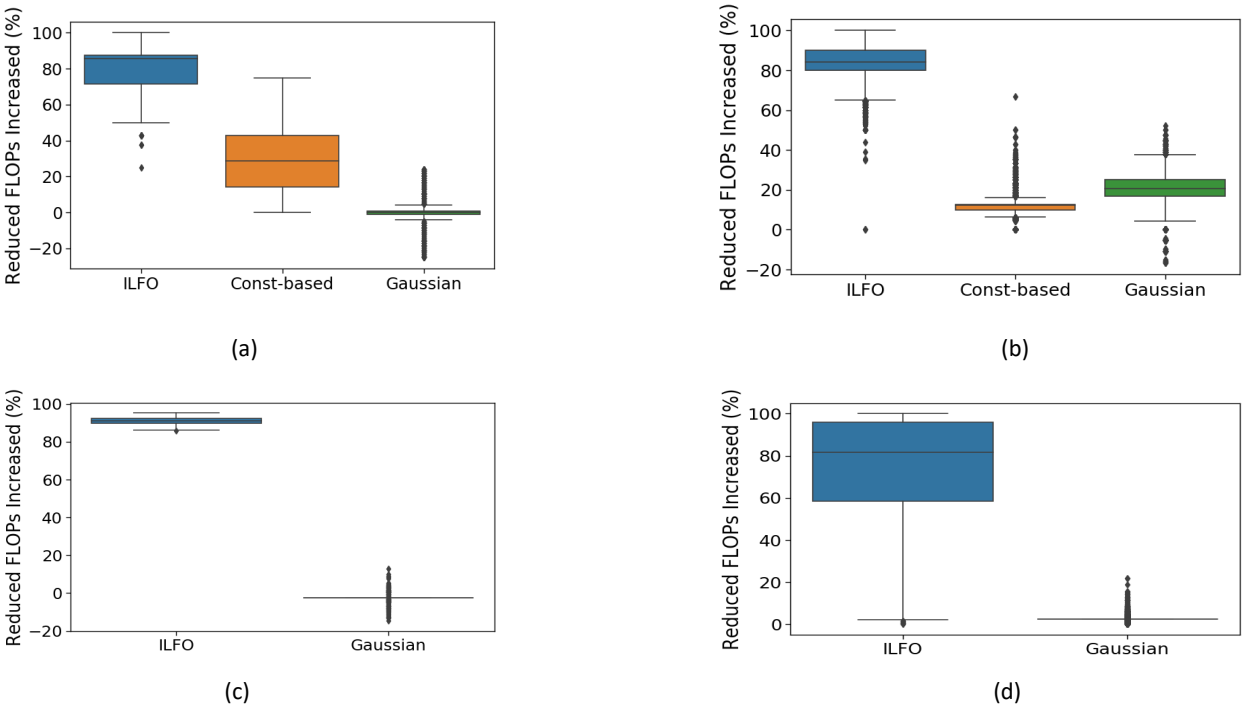


Figure 13. Percentage of reduced FLOPs increased by ILFO. (a) Results after attacking SkipNet using ImageNet images. (b) Results after attacking SkipNet using CIFAR-10 images. (c) Results after attacking SACT using ImageNet images. (d) Results after attacking SACT using CIFAR-10 images.