[Re] Diffusion-Based Adversarial Sample Generation for Improved Stealthiness and Controllability

William Kang,
wkang01@student.ubc.ca

Christina Yang
chryang@student.ubc.ca

Abstract

We are doing a reproducibility report based on the paper "Diffusion-Based Adversarial Sample Generation for Improved Stealthiness and Controllability" by Haotian Xue, Alexandre Araujo, Bin Hu, Yongxin Chen. Their Github Repo is here: `https://github.com/xavihart/Diff-PGD/tree/main`.

The paper uses a novel framework to generate adversarial samples. They use a gradient based method guided by a pre-trained diffusion model to try to generate images that appear realistic to the human eye, can fool a wide range of models, and is easy to control how certain regions are modified.

In particular, we are evaluating the claims that Diff-PGD outperform baseline methods such as PGD, AdvPatch, and AdvCam in physical-world attacks and style-based attacks specifically. Finally, we are evaluating the claim that Diff-PGD generates adversarial samples with higher stealthiness.

The following section formatting is optional, you can also define sections as you deem fit. Focus on what future researchers or practitioners would find useful for reproducing or building upon the paper you choose.

1 Introduction

Neural networks are remarkably good at tasks such as image classification. However, they are susceptible to attacks that perturb images in subtle ways, causing the model to classify one image as something else entirely. These perturbed samples are called adversarial samples, and are a major security concern for systems that use neural networks. There are various types of adversarial attacks, including global attacks, which perturb any part of the image, regional attacks, which change a specified region of an image, style-based attacks, which change an image region based on a reference image, and physical world attacks, which generates images you can print out and place on a physical object which cause the physical object to be misclassified by an image classifier.

The authors of the original paper propose a new method for generating adversarial models, that uses an off-the-shelf diffusion model to help guide the generation of the perturbations in the image, called Diff-PGD. This method aims to create adversarial samples that are more realistic, and harder to detect by the human eye, compared to baseline methods, and can be applied to various types of attacks, such as digital attacks, physical-world attacks, and style-based attacks.

2 Scope of reproducibility

The main claims from the original paper are as follows:

1. Diff-PGD can be applied to specific tasks such as digital attacks, physical-world attacks, and style-based attacks, outperforming baseline methods such as PGD, AdvPatch, and AdvCam.

Course project for UBC CPSC 440/550, April 2024.

2. Diff-PGD is more stable and controllable compared to existing methods for generating natural-style adversarial samples.

3. Diff-PGD surpasses the original PGD in Transferability and Purification power

4. Diff-PGD generates adversarial samples with higher stealthiness

We will be exploring the first claim in regards to physical-world 4 and style-based attacks 4 specifically.

3 Methodology

3.1 Model descriptions

We used ResNet-50, ResNet-101, ResNet-18, Wide-ResNet-50, and Wide-ResNet-101 to classify images. The parameter ‘weights‘ was set to the default weight for each model.

3.2 Datasets

The original paper used ImageNet, but due to limitations in compute and memory, we used a smaller subset of ImageNet with 1000 samples: `https://www.kaggle.com/datasets/ifigotin/imagenetmini-1000`

3.3 Experimental setup and code

We used the authors' code, however, since not all parts of the authors' code have been released, we wrote our own code to evaluate a lot of the experiments. We also expanded on some of the authors' experiments with our own code, and reduced the number of iterations in a number of cases, due to time and compute constraints. We also added our own code for evaluating the success rates of the authors' code.

3.3.1 Success Attack Rate

We wrote our own code to test success rate in `attack_global.py`. We used the same hyperparameters for running Attack_Global as the authors, except that we chose to use skip=20, to use 78 images per iteration to test success rate. If the adversarial sample caused the model to classify the image as anything other than it's original classification, we considered it a successful attack.

3.3.2 Physical-World Attacks

We tried our own physical world attack using an image patch of a digitally drawn panda head and a laptop as our target object.

We use a Galaxy A53 5G to take images from the real world and used a MP C4505ex Color Laser Multifunction Printer to print the images in color.

Due to time and compute constraints, we chose to run 1500 iterations for each method instead of 4000 like the original model. We chose 1500 because after running one of the original paper's physical world attacks, the loss converged after around 1000 iterations.

We took the adversarial samples generated using each method (AdvCam, AdvPatch, and Diff-PDG) and stuck them on a laptop. We then took multiple pictures of the laptop with the sample on top, with the sample in various different locations and rotations, and tested for success rate on the these photos by classifying them with ResNet-50.

3.3.3 Style-Based Attacks

3.4 Computational requirements

We ran it on either CPU or MPS. For used MPS for style-based attacks and global attacks, and CPU for physical-world attacks. For each model, include a measure of the average runtime (e.g. average time to predict labels for a given validation set with a particular batch size).

Global attack success rate took between 5-6 hours to run, while physical-world attacks for one target object and patch took around 1-2 days to run for 1500 iterations.

Style-based attack took around 15s per iteration to run.

4 Results

Start with a high-level overview of your results. Do your results support the main claims of the original paper? Keep this section as factual and precise as possible, reserve your judgement and discussion points for the next "Discussion" section.

4.1 Results reproducing original paper

For each experiment, say 1) which claim in Section 2 it supports, and 2) if it successfully reproduced the associated experiment in the original paper. For example, an experiment training and evaluating a model on a dataset may support a claim that that model outperforms some baseline. Logically group related results into sections.

4.1.1 Attack success rate

We got a very different result for success rate from that of the original paper (fig. 2). In the original paper, the success rate of all of the attacks reached 100% after 5 iterations. However, in our case the success rate reached only 60%-80% after 15 iterations.

4.1.2 Physical-World Attacks

None of the adversarial samples generated by AdvCam, AdvPatch, or Diff-PGD were able to perturb the classification of the image. This was likely due to the fact that we did not adjust the scale properly. This likely means that hyperparameter tuning is very important for physical-world attacks.

Because of this, we failed to support or reject the claim that Diff-PGD can be applied to physical-world attacks. However, from the samples generated, the Diff-PGD samples looked less stealthy and more perturbed than those of AdvCam and AdvPatch, which does not support the paper's claim that Diff-PGD generates samples with higher stealthiness. However this is likely unimportant in regards to physical-world samples, since the samples generated through every method were quite obviously perturbed.

4.1.3 Style-Based Attacks

4.2 Experiment Setup

For each iteration, we take a random base image from the test set of the ImageNetMini dataset. Then for each image we compare the success rate for creating adversarial samples under 3 different attacks:

1. Style-based attack with a different target image from ImageNetMini. Along with a mask generated based on the base image
2. The same as the above, except no mask is applied.

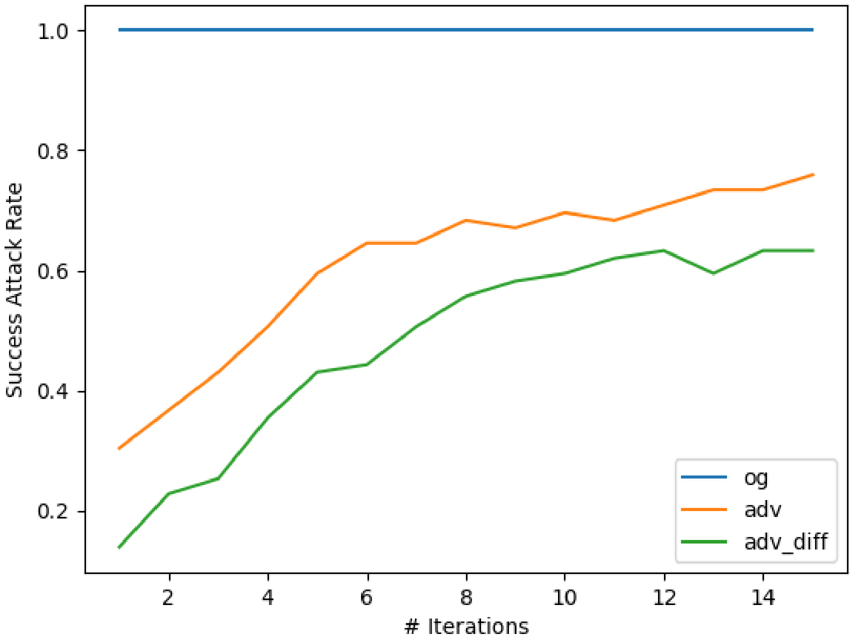

Figure 1: Blue line is success rate of classifying an image not attacked. Orange line is success rate of causing a different classification with Diff-PGD. Green line is success rate of causing a different classification with Diff-PGD after applying sdedit at the end (purification???).

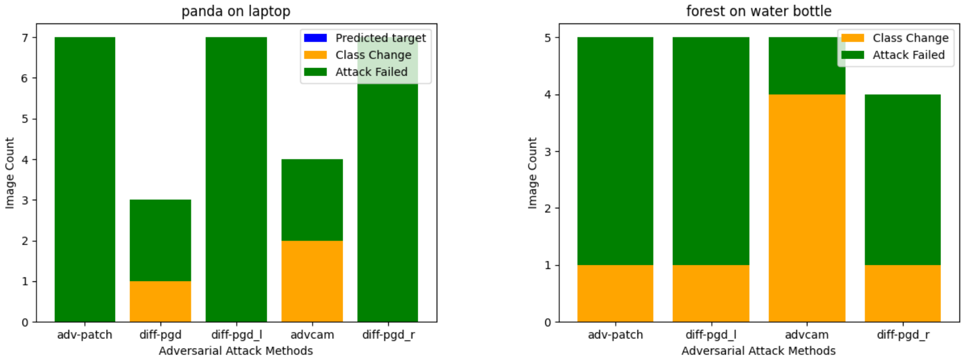

Figure 2: Blue line is the number of images that were classified correctly as the target label. Orange line is the number of images that had a different classification with ResNet-50, but that is not the target class. Green line is the number of images whose classifications were one of the following: ’notebook, notebook computer’ and ’iPod’ for panda on laptop, and ’rain barrel’, ’lighter, light, igniter, ignitor’, ’pill bottle’, ’hand blower, blow dryer, blow drier, hair dryer and hair drier’ for forest on water bottle. These are the classes ResNet-50 gave to images with no adversarial sample (the original image on target object or no image on target object), or classes that we decided had the same meaning as the target object class (ie. laptop vs. notebook computer)

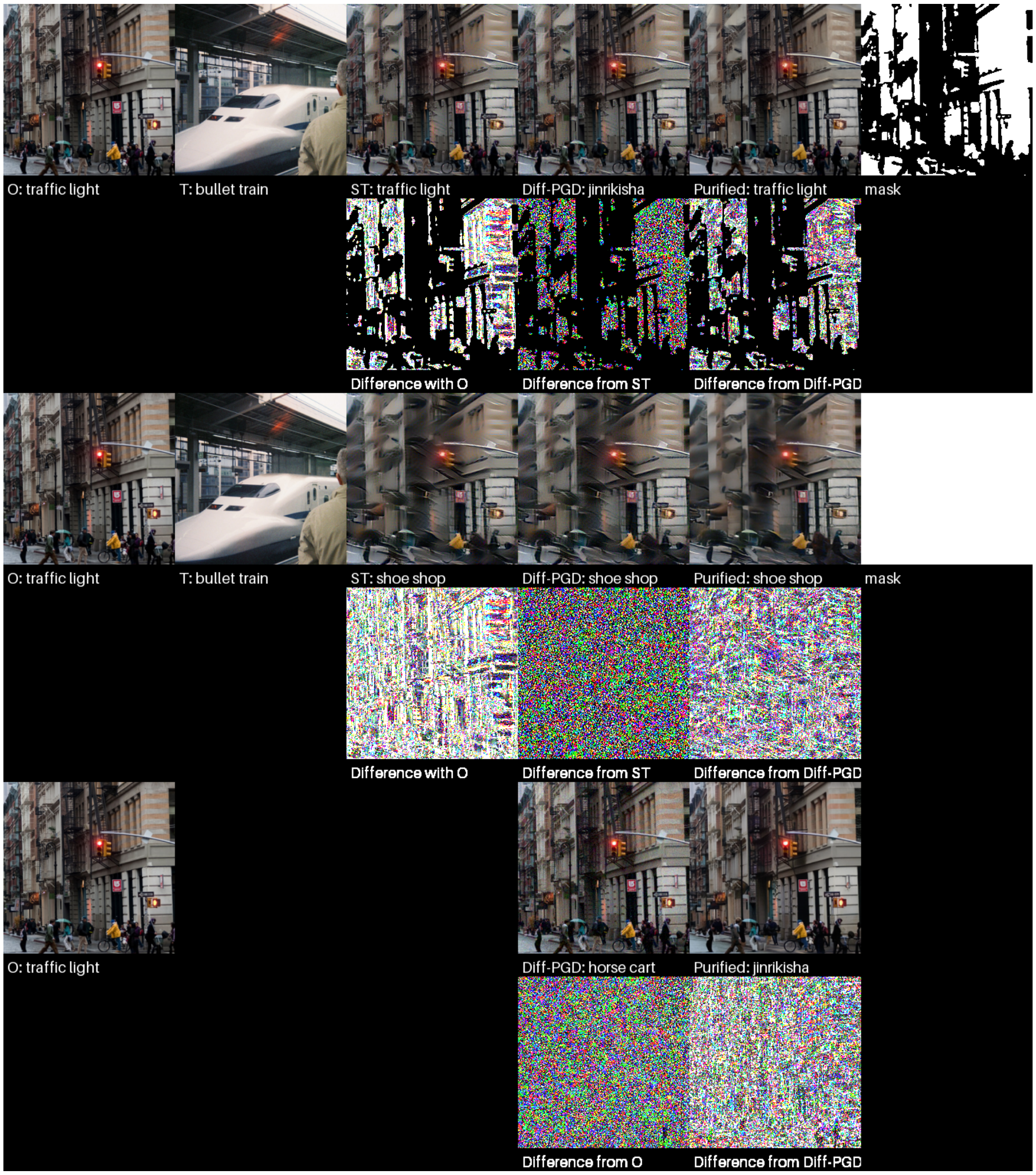

Figure 3: **Visualization of the different attacks**:
There are 6 rows of images. The first two correspond to a style-based attack with a mask applied. The next 2 rows are the same, except there is no mask. The last 2 rows are a global attack where there is no style image. The 6 columns of images are the following: 1: O stands for the original image. 2: T stands for the style image which we try to make the original look more like. 3: ST is the original image with the style applied, but no adversarial properties. 4: Has the Diff-PGD algorithm applied. 5. Is the image after going through purification. 6: Is the mask applied.

3. Global attack which has no target image and no mask

4.3 Results beyond original paper

Often papers don't include enough information to fully specify their experiments, so some additional experimentation may be necessary. For example, it might be the case that batch size was not specified, and so different batch sizes need to be evaluated to reproduce the original results. Include the results of any additional experiments here. Note: this won't be necessary for all reproductions.

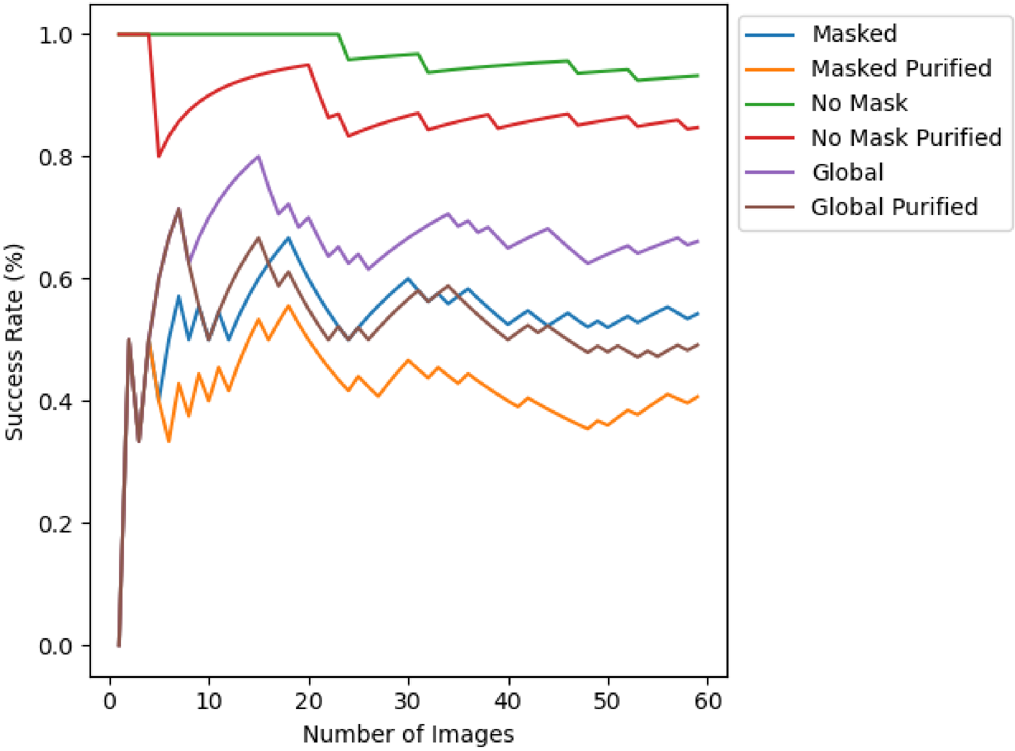

Figure 4: **Plot of the Success Rates**:
We see the best results for applying a style-based attack with no mask. A global attack with no style-image gives the worst results. The success rate for purified images is lower than their non-purified versions.

4.3.1 Additional Result 1

5 Discussion

5.1 Attack Success Rates

Since the authors have yet to release their code for success rate, we are unsure what caused this difference. We did use a different dataset than the authors, this could be a possible reason, since for this calculation, we used 78 images per iteration, while the authors used 250.

5.2 What was easy

Running the code from the original authors to generate adversarial samples was fairly easy, since documentation was included in how to run each attack.

The included code had print statements at each iteration, on the speed each iteration was taking, which was helpful in timing how long the model would take to run.

5.3 What was difficult

List part of the reproduction study that took more time than you anticipated or you felt were difficult.

Be careful to put your discussion in context. For example, don't say "the maths was difficult to follow", say "the math requires advanced knowledge of calculus to follow".

There was little to no documentation on the inner workings of the code, making it more difficult to understand a lot of the attack parameters, and what they did, however the naming was quite clear for the most part, so this was not too difficult to follow.

The physical-world attack was a lot more difficult to reproduce than we anticipated. AdvCam and AdvPatch both ran fairly quickly, averaging around 2-3s per iteration. However, Diff-PGD averaged around 60s per iteration, which took multiple days to run. Because of this, instead of running 4000 iterations as was done in the original paper, we only ran 1500 iterations for each image.
None of the models were able to perturb the classification of the image, which was likely due to the fact that we did not adjust the scale properly. This likely means that hyperparameter tuning is very important for physical-world attacks. However, tuning the model is more challenging for Diff-PGD for physical-world attacks, due to how slowly it runs. Another mistake we made was assuming the the classifier would classify a closed laptop as a laptop, when in reality it classified it as a notebook. We looked into the ImageNet database, and found that most of the images in both laptop and notebook classes were of open laptops, while we used a closed laptop. Since the photo's original label was fed into the model, this could have possibly caused problems with the model.

5.4 Communication with original authors

We did not communicate with the original authors.

References

A Supplementary material

A.1 Physical attacks