Robustness with Query-efficient Adversarial Attack using Reinforcement Learning

Soumyendu Sarkar†* Ashwin Ramesh Babu † Sajad Mousavi † Sahand Ghorbanpour
Vineet Gundecha Antonio Guillen Ricardo Luna Avisek Naug
Hewlett Packard Enterprise
USA

Abstract

A measure of robustness against naturally occurring distortions is key to the safety, success, and trustworthiness of machine learning models on deployment. We propose an adversarial black-box attack that adds minimum Gaussian noise distortions to input images to make machine learning models misclassify. We used a Reinforcement Learning (RL) agent as a smart hacker to explore the input images to add minimum distortions to the most sensitive regions to induce misclassification. The agent employs a smart policy also to remove noises introduced earlier, which has less impact on the trained model at a given state. This novel approach is equivalent to doing a deep tree search to add noises without an exhaustive search, leading to faster and optimal convergence. Also, this adversarial attack method effectively measures the robustness of image classification models with the misclassification inducing minimum $L_2$ distortion of Gaussian noise similar to many naturally occurring distortions. Furthermore, the proposed black-box $L_2$ adversarial attack tool beats state-of-the-art competitors in terms of the average number of queries by a significant margin with a 100% success rate while maintaining a very competitive $L_2$ score, despite limiting distortions to Gaussian noise. For the ImageNet dataset, the average number of queries achieved by the proposed method for ResNet-50, Inception-V3, and VGG-16 models are 42%, 32%, and 31% better than the state-of-the-art "Square-Attack" approach while maintaining a competitive $L_2$.

Demo: https://tinyurl.com/yr8f7x9t

1. Introduction

Deep learning models have yielded impressive results in numerous applications, but research on adversarial attacks has shown that these models suffer from a vulnerability where small distortions could lead to wrong predictions. Specifically, naturally occurring distortions that affect the inputs are of greater concern in safety-critical applications such as self-driving cars, facial recognition, and image-based authorization [15] [21]. Measuring robustness, i.e., how resilient these machine learning models are against distortions, is key to discovering vulnerabilities of poorly trained models.

Literature has provided us with two major paths to identify the sensitivity of the deep learning models, White box attacks [32] [10] and Black box attacks [1] [30]. Even though recent works have introduced efficient white-box approaches targeting a specific region or very minimum distortion to fool the Convolutional Neural Network (CNN) models, it requires complete visibility of the network architecture and the parameters. In general, visibility into the models is not practical in many real-world applications for intellectual property (IP) concerns and support issues. However, black box attacks are inefficient and require too many queries to create the adversarial sample that could break the evaluated model.

In this paper, we propose a black-box approach using a
Figure 2. Average number of queries in un-targeted $L_2$-attacks for ImageNet datasets of 3 CNN models for black-box attacks. RLAB outperforms all other attacks by a large margin. **Note:** There is no official results for pixle with Inception-V3.

Reinforcement Learning (RL) agent (RLAB) that can learn an optimum policy to make an adversarial attack with fewer queries and with a 100% success rate while maintaining other metrics like distortion at a minimum. This is unlike the hand-crafted heuristics that are used in State-of-the-art adversarial attacks. Our method includes a dual action RL agent, which makes parallel addition and removal of distortions to image regions, based on the image region sensitivity at the current state and the history of progression of added distortion as shown in Figure 1. The goal is to cause a misclassification with a minimum number of queries. In an extensive evaluation of un-targeted attacks with ImageNet and CIFAR-10 datasets on CNN architectures such as ResNet-50, Inception-V3, and VGG-16, RLAB outperforms the state-of-the-art methods for $L_2$ threat model on the number of queries while achieving competitive $L_2$-norm as shown in figure 2. The main contribution of the work can be summarized as follows.

1. A novel Reinforcement Learning agent, that beats the state-of-the-art un-targeted black-box $L_2$ attack models in terms of an average number of queries by a wide margin with a 100% success rate while keeping the $L_2$-norm minimum.

2. This RL approach learns a policy to form an optimum adversarial attack agent that can outperform the engineered heuristic approach of the prevailing SOTA adversarial attacks by the above metrics.

3. A high-performance adversarial attack agent that limits the distortions to Gaussian noise, one of the naturally occurring real-life non-malicious distortions, unlike most adversarial attacks.

2. Related Works

Some of the established metrics to evaluate the performance of a machine learning model include accuracy, precision, recall, and F1 score. With the recent advances in adversarial attacks, the models that showed excellent performance on static test sets with the above metrics were easily misclassified with adversarial examples. For example, work done by Szegedy et al. [32] was one of the first works to introduce adversarial attacks. White-box attacks showed great results with one of the initial works from Goodfellow et al. in their work [10] introducing Fast Gradient Sign Method (FGSM) based attack where a small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input changed the classification outcomes. Following this work, there were other incremental works based on gradients-based distortion that could flip the model [17] [16] [6]. Papernot et al. [22] generated an indication map representing the right area on the input that can be attacked. Similarly, DeepFool by Moosavi et al. [20] proposed a simple yet effective approach to add perturbations to the input to fool the machine learning models.

2.1. Black-box attacks

In Black-box attacks, there is only partial visibility to no visibility into the model. In a partially visible black-box attack, information about the loss function, the prediction probabilities, or top-K sorted labels could be available based on which the attack is executed in a query access approach. Work done by Michel et al. [18] and Chakraborty et al. [2] provides a detailed survey on the current trends in adversarial attacks on neural networks. Further, Ilyas et al. [13] in their early work approached this problem with multiple level of restrictions including limited visibility, limited query access and so on. Some of the most popular black-box attack in recent times that has been acknowledged by the research community include Square attack [1], SimBA [11], and LeBA [33], which achieved significant results in breaking Convolutional Neural Network based models. Guo et al. [11] in their work proposed a simple approach where they iteratively and randomly sample a vector from a predefined orthonormal basis such that it can be added or subtracted from the target image. Similarly, Andriushchenko et al. [1] proposed an approach where square-shaped updates are added at random positions such that at each iteration, the total budget constraint is still preserved. Furthermore, some of the most recent works in the black-box attack include EigenBA [34], Pixle [23], Querynet [4], advFlow [19], and CG attack [9] producing state-of-the-art results.

2.2. Reinforcement learning for adversarial attacks

Reinforcement Learning has solved problems that classic machine learning struggles in various domains and applications such as healthcare, energy [26–29], medical imaging, etc. Their unique ability to learn a policy for action...
is a key attribute of their success. Reinforcement learning for adversarial attacks has not been explored much. Sun et al. [31] in their work use reinforcement learning to target graph neural networks via node injections. Similarly, work done by Yang et al. [33] (Patch Attack) applies reinforcement learning to attack CNN models by superimposing textured patches on the input image. Unlike the previous approach, our RL agent uses a comprehensive state representation that captures the model’s sensitivity to various image regions and implements a patch-based process with natural distortions. This enables our approach to significantly outperform state-of-the-art adversarial attacks, including RL-based methods in terms of minimum distortion measured by L2-norm, query efficiency, and success rate.

3. Proposed Method

3.1. Reinforcement Learning/problem formulation

The Deep Neural Network (DNN) model under test/evaluation can be represented as \( y = f(x; \theta) \), where \( x \) denotes the input image, \( y \) represents the prediction and \( \theta \) represents the model parameters. The motivation is to generate a perturbation \( \delta \) such that, \( y \neq f(x + \delta; \theta) \). The objective is to minimize \( \delta \) which represents a measure of robustness.

3.2. RLAB Overview

In our approach, the image is divided into squared patches and sensitivity of the ground truth probability \( P_{GT} \), to addition and removal of distortion, is computed for each patch. Based on the sensitivity information, the RL decides the patches to which Gaussian noise is added or removed at every step. This process is done iteratively until the model misclassifies the image. To further reduce \( L_2 \), we perform an iterative image cleanup as a post-processing step while maintaining the misclassification. The overall flow of the proposed method is represented in the figure 3.

3.3. Image Sensitivity Analysis

In our proposed approach, we limit all distortions to Gaussian noise, as it is a commonly encountered and naturally occurring distortion. During the image sensitivity analysis, we generate a fixed number of noise masks of same noise level, of size \( n \times n \) sampled from a normal distribution as represented in the equation 1.

\[
\text{NoiseMask}(n \times n) = \text{NormalDistribution}(0, \text{Noise level})
\]

At every step during the training and validation, one mask is randomly chosen from the generated noise masks and applied across all image patches to evaluate the drift in the ground truth classification probability \( P_{GT} \). A lower noise level is chosen as it helps more granular addition of noise in successive steps to specific regions that create maximum drift with the \( P_{GT} \), while keeping \( L_2 \) minimum. The noise mask is generated such that they have the same effect on change in \( L_2 \) distance. The perturbations \( \hat{x} - x \) are constrained to the values \([0, 1]^d\). Note that the size of the patch is fixed throughout the experiment and is chosen as a hyperparameter based on the performance-cost trade-off. Table ?? provides detailed experiments on different patch sizes.

3.4. Alternative to Tree Search

Generating adversarial examples for image classification through multiple steps is similar to board games. For board games, the most effective moves or actions are figured out through a Deep Tree Search (DTS) of multiple layers to determine the effectiveness of an action taken at the current step on a longer time horizon as the game evolves. DTS is computationally expensive, even with approximations like Monte Carlo Tree Search (MCTS). But unlike a board game, in this problem, there is a possibility to reset the earlier moves when we realize that we have made a less optimized move a few steps back. In RLAB this is done by removing distortions from some patches and adding distortions to some other patches, considering the state of the
modified image at any given step (equivalent to position on the board). This is equivalent to replaying all the moves in one step while keeping the sensitivity analysis restricted to the current state of the image without a tree search.

Our method reduces the complexity from $O(N^d)$ to $O(N)$ where $N$ represents the computation complexity of one level of evaluation and corresponds to the image size, and $d$ represents the depth of the tree search, which translates to how many queries and actions we would like to look ahead if we were doing a tree search. $d = [1, \text{max\_steps}]$.

### 3.5. Reinforcement Learning

The decision of which patches to choose for adding or removing distortion has multiple dependencies and needs to be adaptive for the most efficient generation of adversarial examples. Mapping this adversarial sample generation as a Reinforcement Learning (RL) problem requires defining the states, actions, and rewards. The state-space is constructed such that the environment becomes observable in a way it enables the RL agent to learn the optimum policy to take actions while maximizing the reward. We used the Dueling DQN Reinforcement Learning (RL) based agent in RLAB. The Dueling DQN model fits well with the discrete action space. Algorithm 1 represents the overall flow of the proposed method. Figure 7 represents the steps involved in adding and removing distortion by the RL agent.

#### 3.5.1 RL States

We designed a state space that gives required observability to the RL agent but is simple enough and of lower dimension such that the agent could be trained efficiently as shown in Figure 6. The image sensitivity analysis acts as a feature extractor where the top ordered square patch locations are ordered both based on the change in $P_{GT}$ for adding and removing distortion in the state vector. Also included are the classification probabilities and $L_2$ distance progression.

#### 3.5.2 RL Action

To keep the number of actions limited and discrete, we define RL action as the number $N_{\text{ACTION}}$, where RLAB adds distortion to the top ($N_{\text{ACTION}} + 1$) patches from the

### 3.5.3 RL Reward

We define a probability dilution (PD) metric, which measures the extent to which the classification probability shifts
from the ground truth to the other classes. The difference between the PD of the altered image and the original image as a result of an action (ΔPD) is a measure of the effectiveness of the action. Moreover, the change in $L_2$-distance ($\Delta L_2$) as a measure of the distortion added is the cost for an action. The reward is defined by the normalized PD as represented in equation 2.

$$R_t = \Delta PD_{\text{normalized}} = -\Delta PD / \Delta L_2 \quad (2)$$

However, there is a dependence on $LIST_{PROP}$ and $LIST_{L2}$ for the optimum action to achieve the best efficiency in terms of both minimizing the $L_2$ distance and number of steps/queries. Through hyperparameter tuning we obtained a discount factor $\gamma = 0.95$, where $\gamma$ determines how much the RL agent cares about rewards in the distant future relative to those at the current step.

Algorithm 1: RLAB: Reinforcement Learning

```
1 Initialization: Policy parameters
2 Input: Validation set, number of iterations $Max_{iter} = 3500$
3 Output: Optimized policy for Dueling DQN
4 for image in validation set do
5    Load the image;
6    Calculate reward $R_t$ and advantage $A_t$ based on current value function;
7    Calculate sensitivity of ground truth classification probability $P_{GT}$
8    to change in distortion for square patches;
9    $i \leftarrow 0$;
10   $Pred_{fstep} \leftarrow 1 - P_{GT}$;
11   while $Pred_{fstep} \neq Pred_{fstep}$ and $i < Max_{iter}$ do
12      Collect set of trajectories (state, action) by running policy
13      $\pi_k = \pi(\theta_k)$ in the environment $\rightarrow$ action;
14      Calculate reward $R_t$ and TD error;
15      $i \leftarrow i + 1$;
16      Update the DQN policy;
17      Compute/take action and perform prediction $Pred_{fstep}$;
18 end
```

4. Experiments

In this section, we discuss the effectiveness of our proposed method with the same experimental setup as our competitors. We evaluate on two popular image classification datasets ILSVRC2012 [25] and CIFAR-10. 80 percent of the validation set was used to train our RL agents, and 20 percent of the validation set was used for evaluation. We performed our attacks on three major Convolution-based Neural Network architectures: ResNet, Inception-V3, and VGG-16. We used three metrics to evaluate the performance of our approach. $L_2$ distance which is a measure of distortion, the average number of queries to make a model miss-classify a correctly classified sample, and the average success rate.

For validation, we had an overall average $L_2$ of 4.03 with the values of pixels ranging between 0 and 1 and setting a maximum query budget of 3500 evaluated over 1000 samples from imagenet dataset on ResNet-50 architecture. A failure case is when the proposed method could not fool the victim model within the given budget, and failure cases were not included in any of the metrics calculated except for the success rate. All experiments were performed for a patch size of $2 \times 2$ and with the noise level of 0.005 as we got the best results for this configuration.

The computation for the complete pipeline is GPU-dependent and is efficiently batched, and scaled on GPUs. Caching techniques were used for pre-computed information such as the noise masks for improved efficiency. Apollo servers with $8 \times V100$ 32 GB GPUs were used for training and validation. We processed 16(images per GPU) x 8(GPUs) = 128 images in a batch for the complete pipeline.

4.1. Evaluation on Imagenet and CIFAR-10

Table 1 aggregates the proposed method’s results compared to other state-of-the-art black-box algorithms on Imagenet dataset for ResNet-50 architecture. The competitors’ results were generated with the best parameters described in their papers. The average Success Rate (ASR) and Average Query (AVG.Q) were calculated for each victim model while the average $L_2$ for most of the competitors were presented in their paper. It can be observed that our proposed approach beats state-of-the-art algorithms for average queries and success rate by a significant margin while maintaining competitive $L_2$. It is also worth mentioning that the proposed approach was able to achieve 100% success rate for a maximum query set to 3500 while the competitors have experiments performed with a maximum query set to 10000. Similarly, from table 2 our proposed approach outperforms competitors for Inception-v3 for average number

<table>
<thead>
<tr>
<th>Attack</th>
<th>AVG.Q</th>
<th>$L_2$</th>
<th>ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-Fool [3]</td>
<td>5000</td>
<td>7.52</td>
<td>-</td>
</tr>
<tr>
<td>Bandits$_{FD}$ (2018) [14]</td>
<td>5251</td>
<td>5</td>
<td>80.5</td>
</tr>
<tr>
<td>HopSkipJumpAttack [24]</td>
<td>1000</td>
<td>11.76</td>
<td>-</td>
</tr>
<tr>
<td>Subspace (2019) [12]</td>
<td>1078</td>
<td>-</td>
<td>94.4</td>
</tr>
<tr>
<td>P-RGF$_D$ (2019)</td>
<td>270.5</td>
<td>-</td>
<td>99.3</td>
</tr>
<tr>
<td>LeBA (2020) [33]</td>
<td>178.7</td>
<td>-</td>
<td>99.9</td>
</tr>
<tr>
<td>Square (2020) [1]</td>
<td>401</td>
<td>5</td>
<td>99.8</td>
</tr>
<tr>
<td>querynet (2021) [4]</td>
<td>-</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>Eigen BA (2022) [34]</td>
<td>518</td>
<td>3.6</td>
<td>98</td>
</tr>
<tr>
<td>Pixle (2022) [23]</td>
<td>341</td>
<td>-</td>
<td>98</td>
</tr>
<tr>
<td>CG-Attack (2022) [9]</td>
<td>210</td>
<td>-</td>
<td>97.3</td>
</tr>
<tr>
<td>Patch Attack (2022) [33]</td>
<td>983</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RLAB (ours)</td>
<td>169</td>
<td>4.01</td>
<td>100%</td>
</tr>
</tbody>
</table>
Figure 8. RLAB’s distortion comparison with Patch Attack [33] and Square Attack [1]. The distortions are exaggerated for better visibility.

Table 2. Performance comparison of our approach with State-of-the-art methods. The average number of queries (AVG.Q) and Success Rate (ASR) were evaluated on victim models for Inception-V3, and VGG-16 on ImageNet dataset.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Inception-v3</th>
<th>VGG-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>NES (2018) [13]</td>
<td>88.2</td>
<td>1726.2</td>
</tr>
<tr>
<td>BanditsTD (2018) [14]</td>
<td>97.7</td>
<td>836.1</td>
</tr>
<tr>
<td>Subspace (2019) [12]</td>
<td>96.6</td>
<td>1035.8</td>
</tr>
<tr>
<td>TMII (2019) [7]</td>
<td>49</td>
<td>-</td>
</tr>
<tr>
<td>LeBA (2020) [33]</td>
<td>99.4</td>
<td>243.8</td>
</tr>
<tr>
<td>Sqr. Attack (2020) [1]</td>
<td>99.4</td>
<td>351.9</td>
</tr>
<tr>
<td>EigenBA (2022) [34]</td>
<td>95.7</td>
<td>968</td>
</tr>
<tr>
<td>Pixle (2022) [23]</td>
<td>-</td>
<td>99</td>
</tr>
<tr>
<td>CG-Attack (2022) [9]</td>
<td>100</td>
<td>139</td>
</tr>
<tr>
<td>Patch Attack [33]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RLAB (ours) [33]</td>
<td>100</td>
<td>132</td>
</tr>
</tbody>
</table>

Table 3. Evaluation of the proposed method with competitors on ResNet-50 model trained on CIFAR-10 dataset.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Avg. queries</th>
<th>S. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimBA-DCT [11]</td>
<td>353</td>
<td>100</td>
</tr>
<tr>
<td>AdvFlow [19]</td>
<td>841.4</td>
<td>100</td>
</tr>
<tr>
<td>MetaAttack [8]</td>
<td>363.2</td>
<td>100</td>
</tr>
<tr>
<td>AdvFlow [19]</td>
<td>598</td>
<td>97.2</td>
</tr>
<tr>
<td>CG-Attack [9]</td>
<td>81.6</td>
<td>100</td>
</tr>
<tr>
<td>EigenBA [34]</td>
<td>99</td>
<td>99.0</td>
</tr>
<tr>
<td>RLAB (ours)</td>
<td>60</td>
<td>100</td>
</tr>
</tbody>
</table>

of queries while maintaining competitive queries for VGG-16. Furthermore, we have achieved a 100% success rate for both Inception-v3 and VGG-16 models. Table 3 shows the performance of the proposed method against state-of-the-art attacks on CIFAR-10 dataset.

4.2. Nature of Distortions

Most state-of-the-art competitive solutions use unnatural modifications as shown in Figure 8. The only other RL method used for a similar adversarial attack, Patch Attack, has completely unnatural squared patches placed on the images. Also, as shown in Figure 8, the state-of-the-art high-efficiency Square Attack has unnatural colors of red and green all over the cougar, unlike our RLAB method. In contrast, our proposed method preserves the true nature of the image with barely perceptible Gaussian noise.

5. Conclusion

Reinforcement Learning proved to be very effective in learning the optimum policy to make the complex decision of choosing the square patches for changing distortion, as compared to hand-crafted heuristics. RLAB outperforms the state-of-the-art adversarial attacks in query efficiency by a significant margin and achieves a highly competitive $L_2$-norm indicative of very low distortion with 100% success rate for mis-classification. This RL design can be extended to include other types of distortions as part of future work. Also, this RL approach is generic enough to extend to a wide variety of adversarial attack agents beyond image classifiers. As RLAB only uses Gaussian noise, the distortions are similar to real-life deployment. This makes it valuable for a more appropriate test for non-malicious distortions and an effective measure of robustness, which is a key attribute of trustworthiness with a positive social impact. The adversarial samples generated by RLAB can be used to augment the train data set to retrain the model and enhance its robustness.
References


[27] Soumyendu Sarkar, Vineet Gundecha, Sahand Ghorbanpour, Alexander Shimakov, Ashwin Ramesh Babu, Alexan-
dre Pichard, Mathieu Cocho, and Hewlett Packard Enterprise. Function approximations for reinforcement learning controller for wave energy converters. 2


