UNIVERSITY OF CALIFORNIA

Los Angeles

Security Through Stochasticity

Toward Adversarial Defense using Energy-based Models

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Computer Science

by

Jonathan Craig Mitchell

2020

ABSTRACT OF THE THESIS

Security Through Stochasticity
Toward Adversarial Defense using Energy-based Models

by

Jonathan Craig Mitchell
Master of Science in Computer Science
University of California, Los Angeles, 2020
Professor Song-Chun Zhu, Chair

This paper serves as an investigation in the use of energy-based models for adversarial defense via purification and training. Convergent and non-convergent energy-based models are tasked to remove white-box adversarial signals embedded into images from the CIFAR-10 dataset so that they may be classified correctly. This work presents an analysis behind the stochastic behavior of MCMC sampling for adversarial noise reduction in meta-stable energy basins and the benefits and challenges associated with different regimes of energy-based learning for this task.

The thesis of Jonathan Craig Mitchell is approved.

Cho-Jui Hsieh

Yingnian Wu

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2020

To my mother and father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I extend my deepest gratitude towards my advisor Dr. Song-Chun Zhu for allowing me to work with such a talented group of people at UCLA. To Mitchell Hill for teaching me the underlying methodologies behind MCMC and the principles behind energy-based modeling.

VITA

2017–2018	Computer Vision Engineer, Octi inc. Los Angeles, California
2015	B.S. (Electrical Engineering), UC Davis, Davis, California.
Present	M.S. Candidate (Computer Science), UCLA., Los Angeles, California
2019–present	Research Assistant, Computer Science Department, UCLA.
2019–present	Teaching Assistant, Computer Science Department, UCLA.

CHAPTER 1

Introduction

Security and safety of machine learning systems is paramount due to their increased adoption in modern society. Deep neural networks have a variety of use cases and trained DNN models are being used in autonomous vehicle perception, person identification, fraud detection, and Natural Language Processing.

The goal of this work is to create an adversarial robust purification method to remove adversarial signals from a perturbed image for the task of image classification. This work leverages the use of an MCMC-based energy model as an auxillary purification tool to remove adversarial signals.

This work explores the use of both convergent and non convergent energy based models (EBM's). The difference between these models is in the way they are trained as well as their probability densities. The non convergent models are trained from noise initialization whereas the convergent models are trained using persistent chains. Non convergent models are great for image synthesis but lack steady state sampling, a feature of the convergent models. In abusing synthesis of the non convergent model, we attempt to reconstruct the adversarial signal in a stochastic fashion in order to remove the noise as well as prevent an attacker from differentiating through the add-on purification unit to create strong attacks.

We will also explore the use of adversarial-based classifier training that encompasses the purification unit in order to create models robust both PGD attack and BPDA attacks.

CHAPTER 2

Background

2.1 Adversarial attacks

A seemingly benign change to an input of a trained state of the art classifier can cause the classifier to be fooled. The adversary targets the input to create an adversarial sample indistinguishable from the original input as shown in figure 2.1. This is known as an adversarial attack; an algorithm that perturbs an image to fool a classifier. This may cause serious security issues as vision algorithms integrate into our daily lives.

2.1.1 What are adversarial attacks

There are three categories of adversarial attacks with respect to vision and images. In increasing order of strength these methods are widely known as "black box", "transfer" and "white box" attacks respectively. Black box attacks are aware of a targets task, dataset, and training environment (hyper parameters and tuning variables), but they are not aware of the model parameters (weights). Transfer attacks utilize gradient-based information of a surrogate model trained in the same environment as the target model and attempt to transfer the attack to the target. A white box attack has direct access to the model's parameters and utilizes gradient based information to uniquely target each specific model. This work is primarily concerned with white box attacks, specifically FGSM, PGD, and BPDA attacks.

2.1.1.1 Formulation

. Consider the saddle point optimization formulation from [MMS17]. Given a dataset $\{X_i\}_{i=1}^n$ where $X_i \in R^D$ with underlying data distribution q. Natural image training based on empirical risk minimization seeks to minimize $E_q[L(x, y, \theta)]$ where $x \in X$

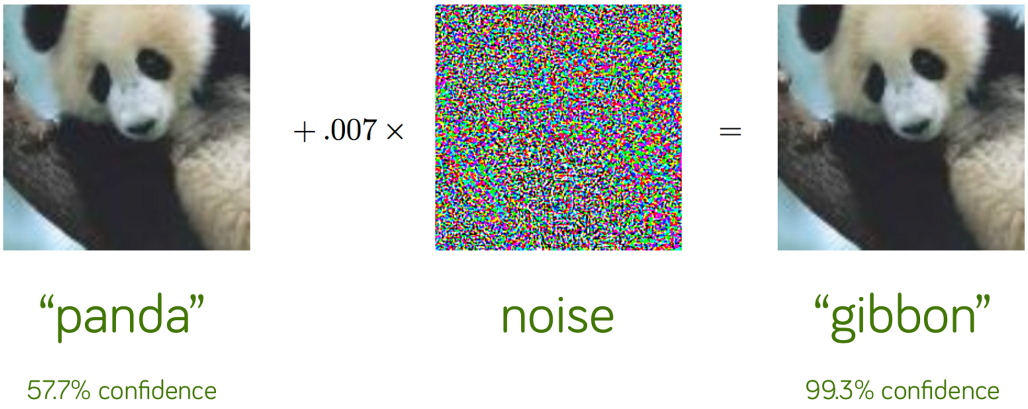

Figure 2.1: Adversarial signal added to a simple panda image using the FGSM attack (equation 2.2) applied to GoogLeNet[SLJ14] trained on Imagenet [RDS14]

and $y \in R^k$ are labels with k classes and θ are the classifiers trainable parameters. However, simple empirical risk minimization will not provide an adversarially robust classifier (as shown in fig 2.1). Therefore our goal is to train the classifier under the following optimization criteria:

$$\min_{\theta} E_{(x,y)\in q}[\max_{\delta \in S} L(x+\delta, y, \theta)] \tag{2.1}$$

Where $S \in R^D$ is the set of allowed pixel perturbation around the original image constraints by the l_∞ norm which is considered to be an $\epsilon - ball$ around x. δ is one of those perturbations such that $x + \delta \in S$. $L(x + \delta, y, \theta)$ is the classifiers loss function. This saddle point formulation can be decomposed into the inner maximization portion; whose goal is to create "true adversaries" that are able to fool the classifier. The outer minimization portion is tasked to limit the amount of these adversaries and create a robust classifier that can't be fooled. Robustness is a heuristic used to measure the accuracy of a classifier with respect to adversarial samples.

2.1.1.2 Targeted vs Untargeted attacks

There is also a distinction among adversarial attack destinations. Given an input x_i with label y_i, a targeted attack is one where the adversary attempts to perturb the input x_i so that the classifier predicts class y_j st $i \neq j$ where j is a specific class target. An untargeted attack creates an adversary to increase $L(x, y, \theta)$ solely to cause misclassification of y_i without any specific ”targeted” class in mind. In both cases the perturbation should be ”imperceptible” such that the original image and the adversarial image can not be distinguished by humans.

2.1.1.3 Specific attacks

In this work we describe three different untargeted attacks. The Fast Sign Gradient Method (FGSM) attack is an l_∞ bounded adversarial algorithm from [MMS17] that computes adversarial examples using eq (2.2)

$$\hat{x} = x + \epsilon sgn(\nabla_x L(\theta, x, y)) \tag{2.2}$$

where L represents the loss function after a forward pass of the network and $\hat{x} = x + \delta$ is the adversarially perturbed image, x corresponds to the original image, y the class label, θ the model parameters, and ϵ is the constraint of allowed perturbation of each pixel with respect to the l_∞ norm. We also borrow a variant of FSGM from [MMS17] known as Projected Gradient Descent (PGD), that iteratively attacks each newly formed adversarial image and projects it back to the l_∞ constrained ϵ-ball around the original image x

$$\hat{x}_{i+1} = \Pi_{x+S}(\hat{x_i} + \alpha sgn(\nabla_x L(\theta, x, y))) \tag{2.3}$$

where α is the learning rate and where the space of allowed pixel perturbations on x is S specified by the aforementioned l_∞-ball around x. This ensures that the difference between x and the adversarial image $\hat{x}_{i+1}$ (which has gone through multiple attacks) is imperceptible. Both of these attacks are considered white box because they utilize gradient information of the model and untargeted because they are not being pushed towards a specific class.

Currently, the most robust form of adversarial defense against equationss 2.2, 2.3 is to train a classifier on adversarial samples, as shown in [MMS17]. However, this detracts from the original task of the model in that it does not increase task performance (natural image classification) and increases both training time and computational load. For this purpose, auxillary white box defense methods that do not require classifiers to undergo adversarial training have been explored such as [SKN17], [SKC18], etc. We will refer to these methods as "add-on purification" and their defense algorithms as purifiers. These purifiers proved hopeful until further analysis by the authors of [ACW18] revealed that the majority of these methods were simply adding non-differentiable components/layers to existing classifiers which caused them to "obfuscate" their gradients and create weak adversarial samples during testing.

To combat this "false sense of security" the authors of [ACW18] created a Backwards Pass Differentiable Attack (BPDA) which is a straight-through attack algorithm that is able to differentiate through add-on purifiers to the core network in order to create adversarial samples [ACW18]. The approach consists of performing a forward pass on the network in standard fashion and simply replacing the purifier with the identity on backwards pass differentiation.

$$\hat{x}_{i+1} = \Pi_{x+S}(\hat{x}_i + \alpha sgn(\nabla_x L(\theta, f(g(x)), y)) \tag{2.4}$$

Eq 2.4 provides an approximation of the true gradient because on average $g(x) \approx x$. However, this also requires that more iterations of the attack are performed because $g(x)$ is treated as an approximation of the true gradient on each step. The function $g(x)$ is the purifier in this case. We can treat the output as $x_p \leftarrow g(x)$ where x_p is a purified image. We perform the same projection as in eq 2.3 after the perturbation.

2.2 DeepFrame

The use of MCMC based models for adversarial defense is based on the MCMC property that after an infinite number of steps the distribution $p_\theta(x)$ should not depend on the initial state. Therefore, if the initial state contains adversarial noise, after enough MCMC steps, that noise should disappear. However, full MCMC mixing behavior is not desired because sampling would occur between different modes of the distribution. Thus, we rely on a "meta-stable" region around the adversarial signal that would prevent mixing between modes and enable us to purify an image within the same local energy basin. Additionally, the stochastic nature of the DeepFrame model [LZW15], [WXZ18], when used for LMC sampling, makes it tough for an attacker to backpropagate through the purification network to create effective adversarial samples, thus we utilize security through stochasticity.

2.2.1 Formulation

To present a formal definition of the models used herein, we begin with an energy-based Gibbs-Boltzmann density and propose the formulation as seen in [NHH19], [BZ19].

$$p_\theta(x) = \frac{1}{Z(\theta)} exp\{-U(x;\theta)\}$$

where x is an image signal and $U(x;\theta)$ is an energy potential that belongs to a family of distributions $P = \{p_\theta\}_{\theta \in \Theta}$

Stochastic gradients are useful in cases where the partition function $Z(\theta) = \int_X exp\{-U(x;\theta)\}dx$ is intractable. Our goal in using this energy-based model is to synthesize realistic images $x \sim p_\theta(x)$ to be as close as possible to the true data distribution $q(x)$. In doing so we formulate our loss function as

$$\min_\theta KL(q||p_\theta) = \min_\theta E_q \left[log \frac{q(x)}{p_\theta(x)} \right] \tag{2.5}$$

$$\min_\theta E_q[\log q(x)] - E_q[\log p_\theta(x)] \tag{2.6}$$

where E_q does not depend on θ. Additionally, for an i.i.d dataset $\{X_i\}_{i=1}^n$, using the law of large numbers we can approximate the expectation of the true underlying distribution $E_q[\log p_\theta(x)] \approx \frac{1}{n}\sum_{i=1}^n \log p_\theta(X_i)$ therefore

$$= \min_\theta -E_q[\log p_\theta(x)] \tag{2.7}$$

$$= \max_\theta E_q[\log p_\theta(x)] \tag{2.8}$$

and therefore minimizing $KL(q||p)$ also maximizes the log likelihood of $p_\theta(x)$ which is equivalent to minimizing the negative log likelihood. The likelihood $l(x|\theta)$:

$$\min_\theta l(x|\theta) = \min_\theta \log(Z(\theta) + E_q[U(X;\theta)] \tag{2.9}$$

We can approximate the intractable partition function using the gradient of $\log Z(\theta)$ which can be expressed in closed form as $\nabla_\theta \log Z(\theta) = -E_{p_\theta}[\nabla_\theta U(X;\theta)]$ [GBC16], thus we can minimize $l(\theta)$ by taking the derivative

$$\frac{\partial}{\partial_\theta} l(x|\theta) = \frac{\partial}{\partial_\theta} E_q[U(X;\theta)] - E_{p_\theta}[\frac{\partial}{\partial\theta} U(X;\theta)] \tag{2.10}$$

$$\approx \frac{\partial}{\partial_\theta}\big(\frac{1}{n}\sum_{i=1}^{n} U(X_i^+;\theta) - \frac{1}{m}\sum_{i=1}^{m} U(X_i^-;\theta)\big) \tag{2.11}$$

where $U(X_i^+;\theta)$ are known as positive samples that follow the true underlying distribution of the data q and where $U(X_i^-;\theta)$ are known as negative samples obtained using MCMC from the models currently learned distribution $p_\theta(x)$. [NHH19]. Positive samples are simply randomly sampled training images while the negative (MCMC) samples are obtained using Langevin dynamics. The advantage of using an energy-based model is that it does not have to approximate the partition function because it simply tries to create "realistic" synthesized images from our model and compare them to the data itself. Thus MLE forces the MCMC samples from the model $p_\theta(x)$ to be as close to q as possible.

For this application we consider our model $p_\theta(x)$ to be a lightweight Convolutional Neural Network (CNN) where a forward pass of the network $f(x;\theta) = -U(x;\theta)$ and where $U(x;\theta) \in R$. Moreover, there are two regimes for training the DeepFrame model; convergent and non-convergent. Convergent models have a probability density function that closely approximates the steady state of the model's distribution p_θ (converges to steady state). Non-convergent models have a probability density function near p_θ but only as a crude approximation. The non-convergent models have great synthesis as shown in [NHZ19] and don't need as many Langevin updates (eq 2.12) to purify adversarial signals making them ideal for purification over a large number of adversarial attacks. The non-convergent models can be initialized from noise whereas the convergent models need many more updates to approximate the steady state distribution and must be initialized from persistent chains. Explicit details on how to train these auxillary networks can be found in [NHZ19], [NHH19].

2.2.1.1 Langevin dynamics

We utilize the Langevin equation to perform Langevin Monte Carlo (LMC) sampling of the model during training to obtain the negative samples. LMC is a special case of HMC that is used when the trajectory to propose a new state consists of one leapfrog step. [Nea12]. In typical LMC we sample momentum variable values from their zero mean and unit variance Gaussian white noise distribution $Z_i \sim N(0, I)$.

$$X_i^* = X_i - \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}U(X_i;\theta) + \varepsilon Z_i \tag{2.12}$$

Where $\varepsilon > 0$ is a constant noise factor. According to the work of [CFG14], [NHH19], the momentum update of the second HMC variable as well as the Metropolis-Hastings update step can be ignored in practice. For this application it is useful to observe that the added noise introduces stochastic behavior into the algorithm which contributes to the stochastic vector in the purification triangle 3.7. This should halt backpropagation through the purifier (DeepFrame) and prevent the BPDA attack from creating strong adversaries. Once trained, the Energy-based model (EBM) is able to sample realistic images from the data.

CHAPTER 3

Purification

3.1 Add-on purification

We can consider the result of eq 2.12 as extracted MCMC (negative) samples from the model $p_\theta(x)$. The EBM is trained using natural images and therefore knows nothing about adversarial signals. The EBM has shown to be proficient in sampling realistic images from noise and persistent chains. Therefore, if we initialize the MCMC sampling process from adversarial images, and iteratively sample using equation 2.12 we can get rid of the adversarial signal. We demonstrate the effectiveness of this method as a defense against the BPDA attack mentioned earlier by purifying adversarial signals that result from BPDA.

Figure 3.1 is provided as a visual representation of the BPDA algorithm flow.

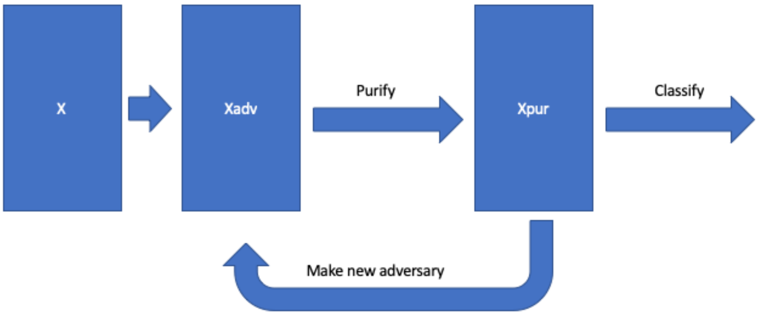

Figure 3.1: Visualization of the BPDA pipeline

Algorithm 1 Adversarial defense against BPDA attack using purification

```
Require: $X_{i=1}^{m}$ = mini batch of natural images of size m, $K = 15$ langevin updates,
  $T = 100$ adversarial attack iterations, $\alpha = 1$ adv step size, $\epsilon = \frac{8}{255}$ maximum allowed
  per-pixel deviation, $error = 0$,
  for t=1:T do
    if $t = 1$ then
      $\tilde{X}_{t-1} = X$
    end if
    $X_p \leftarrow g(\tilde{X}_{t-1})$ where $g(.)$ is eq 2.12 (purification)
    $\hat{X}_t \leftarrow \hat{X}_{t-1} + \alpha \nabla_X L(X_p, Y, \Theta)$ using eq 2.4 (make new adv)
    $\tilde{X}_t \leftarrow \max(\min(\hat{X}_t, x + \epsilon), x - \epsilon)$ enforces $l_\infty$ norm by projecting into $\epsilon$-ball of $X$
    if $argmax(\sigma(f(X_p))) \neq Y \forall i \in X_i$ then
      $error \leftarrow error + 1$
      breaks classifier
    end if
  end for
```

Where $X_{i=1}^{m}$ is a mini batch of natural images of size m and $\tilde{X}$, $\hat{X}$ are its adversarial counter-parts. On the first iteration $t = 1$, we purify the original image X but on iterations $t > 1$ we purify the adversarial images $\tilde{X}$ in order to remain consistent with BPDA. Classifier f is evaluated on "purified" samples and not on the original images. Therefore adversarial attacks that utilize gradient-based information of the original image data will be less successful because they lack knowledge of the purified samples.

$L(X_p, Y, \Theta)$ represents the loss function of a classifier f where Y are the data labels and θ are its weights/parameters. The function $g(x)$ creates the purified image X_p using eq 2.12 with $\varepsilon = 0.01$ for K langevin updates, T is the number of attack iterations of BPDA which we have set to 100 in order to be consistent with results from [ACW18]. The step size α denotes the strength of each adversarial perturbation (length of path in ϵ-ball), and ϵ specifies the ϵ-ball around the original image X which is the maximum allowed per-pixel deviation that approaches the limit of human perceptibility constrained by the l_∞ norm. $\sigma()$ represents the log-softmax function.

label	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
color	red	orange	blue	purple	white	cyan	pink	brown	magenta	turqoise

Table 3.1: Label maps for purification images

3.1.1 Saturation

The use of the non-convergent energy-based model risks not only removing the adversarial signal but also important image features. When the non-convergent (short-run) model runs too many Langevin updates (eq 2.12) sampling moves towards the mode of the dataset instead of the steady state (as convergent models do). Examples of this saturating phenomena are shown in figure 3.6 which display purified images in the saturation region. The columns span 20 attack iterations of BPDA and each row is the result of a purified adversary over each iteration. The more Langevin steps performed, the more both the adversarial signal and important image features are reduced (saturation). Therefore, more Langevin updates will not lead to better adversarial defense on its own. This is present if we consider the increased number of colored boxes from 3.4, 3.5, 3.6. In these figures each colored image represents an incorrectly classified image where each color maps to a label shown in table 3.1. A concise description of this phenomena can be visualized in figure 3.2. Readers can refer to fig 3.3 to visualize how adversarial and purified images relate to the classifier's loss landscape.

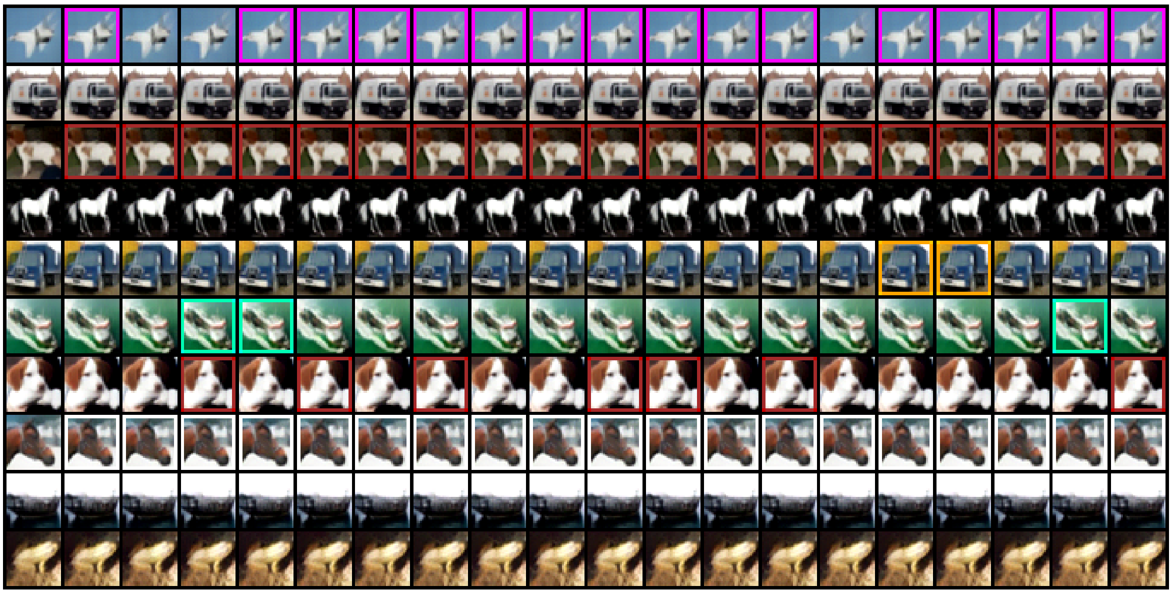

Figure 3.4: Adversarial purification using K=15 langevin steps (eq 2.12)

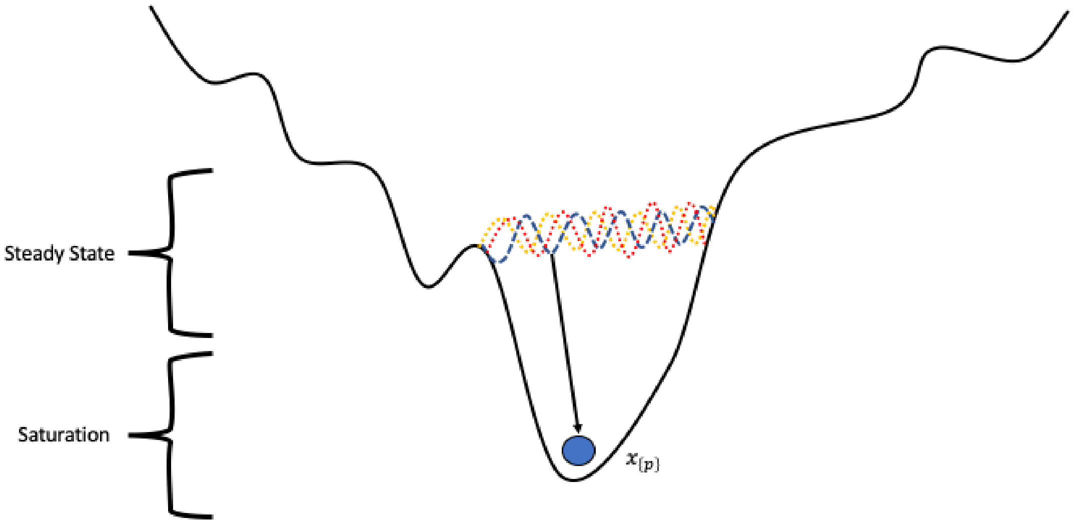

Figure 3.2: Visualization of the energy landscape $U(x;\theta)$. The blue line represents the data distribution q. The yellow dashed line represents steady state samples $x \sim p_\theta$ from a convergent DeepFrame model and the red dashed line represents the adversarial samples $\hat{x} \leftarrow x + \delta$ from PGD or BPDA. Note that all three distributions have relatively the same energy because there is no perceptible difference between them (due to the l_∞ norm perturbation restriction). The blue sample represents the occurrence of saturation when an adversarial sample is purified using too many Langevin steps, as seen in fig 3.6. In this case, the sample goes towards the mode of the distribution which has low energy.

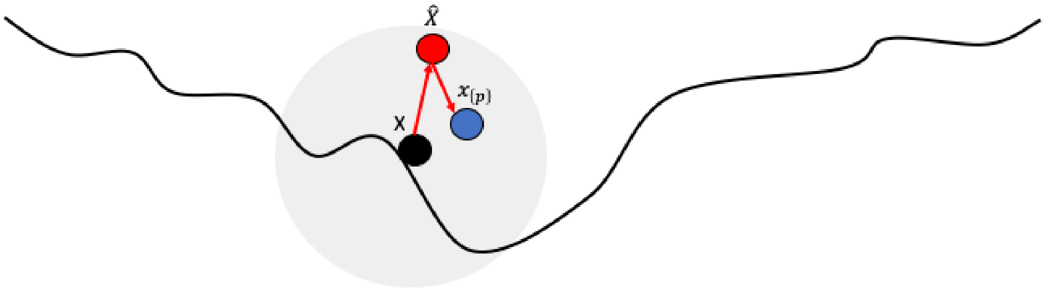

Figure 3.3: Visualization of the loss landscape $L(x, y, \theta)$ of the classifier f. The gray region around the data X represents the ϵ-ball of allowed perturbation. Adversarial samples and purified samples have higher loss than the data. In this case the purified samples are within the ϵ-ball but that need not be true. The steady state distribution (from convergent sampling) may also have much higher loss than the data distribution.

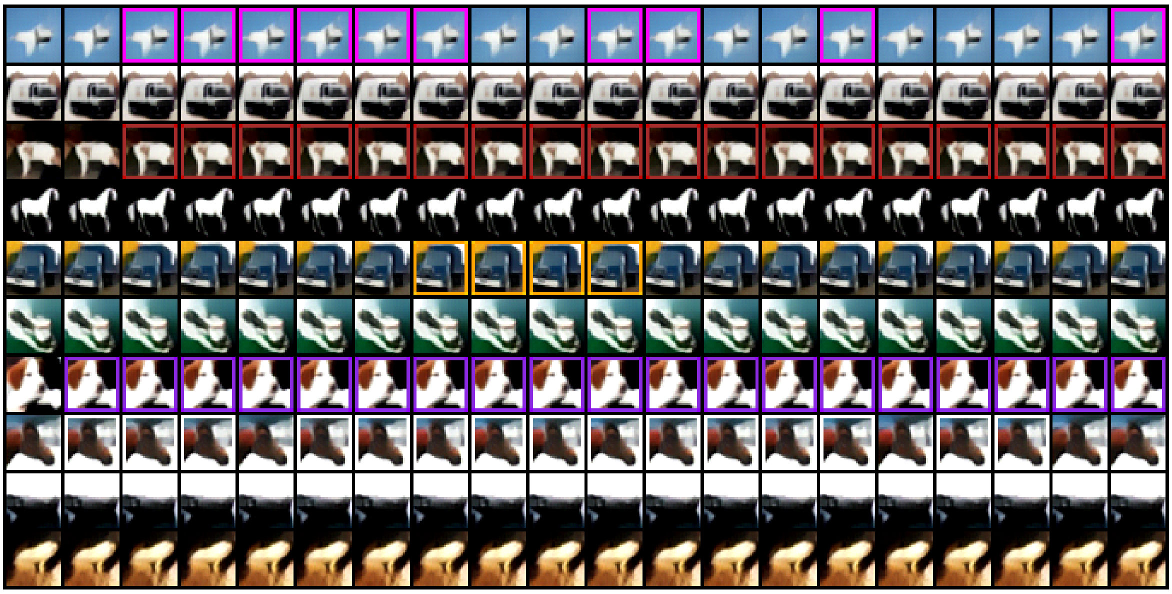

Figure 3.5: Adversarial purification using K=30 langevin steps (eq 2.12)

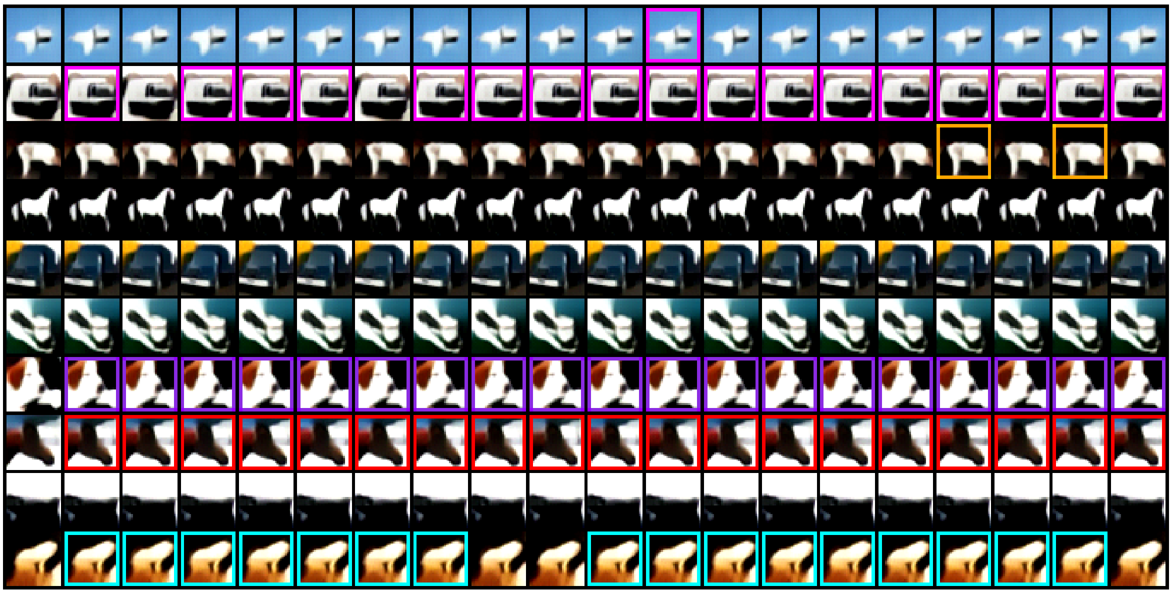

Figure 3.6: Adversarial purification using K=45 langevin steps (eq 2.12)

3.2 Purification Triangle

Non-convergent noise initialized synthesis model's are able to start from noise and create realistic images. Therefore, they create direct pathways between noise and realistic images. Our assumption is that these models will treat the adversarial signals as ”noise” and simply purify them through synthesis.

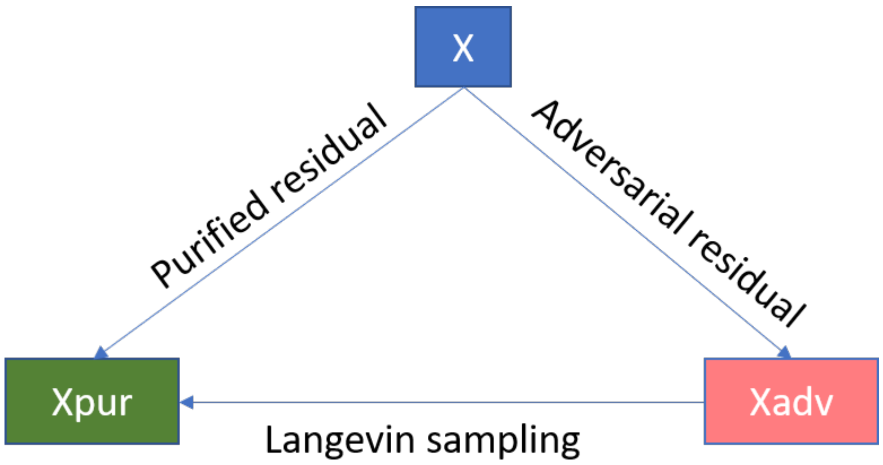

Figure 3.7: Purification Triangle where Langevin sampling represents stochastic Langevin dynamics in eq 2.12 where X_{pur} are purified samples and X_{adv} are the adversaries created from X_{pur} using eq 2.4

We consider the Pearson correlation between the adversarial residual $R_A = \hat{x} - x$ and the purified residual $R_p = x_p - x$ as a meaningful metric to determine the relative distance traveled in R^D by the purification process. The purification process can be represented by the stochastic Langevin sampling vector $X_{pur} \leftarrow g(x_{adv})$ in figure 3.7. Consider figure 3.8 which displays the correlation between R_A, R_p over different step sizes α in 1. If the step size of the BPDA attack α is smaller, then the adversary traverses a finer path in the l_∞ ball of the image space, leading to stronger adversarial noise, causing each Langevin step to be less orthogonal to the adversarially perturbed image. When using a larger step size in the BPDA attack, we obtain a rougher approximation of the "true" adversarial path, leading to weaker noise which allows the Langevin sampling process to recognize and remove this noise. For this reason, residual correlation is a quantifiable metric to assess purification on the BPDA attack. Figure 3.8 displays this correlation over different attack and Langevin steps. Therefore, the step size α quantifies how much the adversarial signal appears as a feature instead of noise. If the signal appears as noise, the model will be able to directly reduce the noise and traverse the synthesis path, but if the signal appears as a feature then the model will keep it and classify incorrectly.

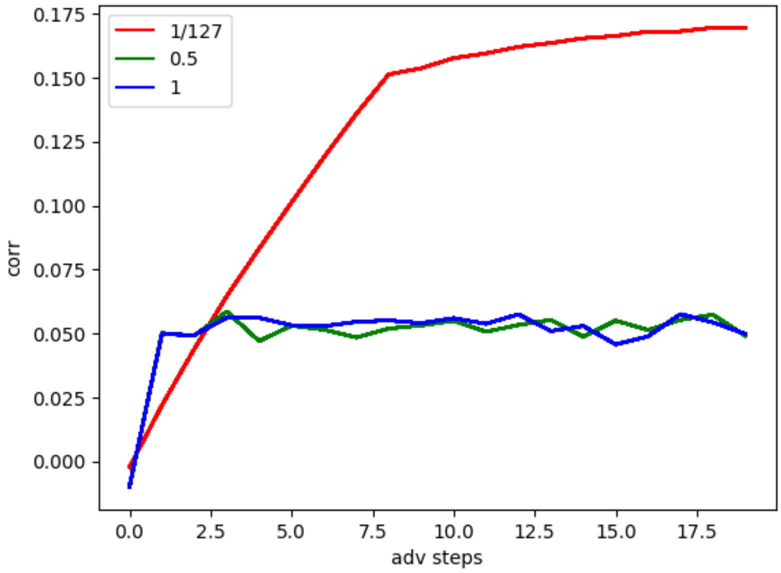

Figure 3.8: Residual Correlation of the purification triangle. Only 20 attack iterations (x-axis) are shown because the loss saturates after that. Results are displayed for step size $\alpha = \frac{1}{127.5}$,0.5,1.

3.3 Reconstruction-based Purification

The non-convergent MCMC model is trained using noise initialization $z \sim U(-1, 1)$ and is optimized based on equation 2.11 so that the positive and negative samples are as similar as possible. If we are able to initialize this model from noise z and obtain realistic images then we can consider this model to be a generator of the form $x = M_\theta(z)$. This was discovered in [NHZ19], which also claims that we can reconstruct an image x by running gradient descent over the reconstructive loss. We utilize this result and take an approach similar to Defense-Gan [SKC18] to reconstruct our adversarial image $\hat{x}$ from noise. In Defense-Gan, the authors utilize a generator network and sample from the latent vector $G(z)$ to reconstruct an adversarial image. Their claim is that generative sampling will lead to imperfect reconstruction which will remove a lot of the adversarial noise. However their results are only supported on the MNIST dataset and not on CIFAR-10. To adopt this algorithm for the task of reconstruction-based purification we consider the adversarial image $\hat{x}$ and the reconstruction loss

$$L(z) = ||\hat{x} - M_\theta(z)||_2^2 \tag{3.1}$$

where we initialize z_0 from uniform noise and perform

$$z_{t+1} \leftarrow z_t - \eta \nabla_{z_t} L(z_t) \tag{3.2}$$

In order to generate negative samples from the noise $M_\theta(z)$ we must perform equation 2.12 for K steps which involves computing the gradient of the energy-based network. Then we must also compute the gradient from equation 3.2 and store both in memory, thus taking the double derivative. Therefore, this defense method is extremely computationally heavy and we provide a separate category for it in our results in table 3.5.

In order to achieve adversarial an even more stochastic and robust model we utilize a recursive "rolling" procedure. we first reconstruct the adversarial image

$$x_t \leftarrow h(\hat{x})$$

where $\hat{x}$ is the adversarial image and $h(.)$ is equation 3.2, which is run for 240 iterations. Then we "roll" the algorithm and reconstruct x_{240} by setting x_{240} as the reconstruction target and running the algorithm for another 240 iterations.

$$x_{480} \leftarrow h(x_{240})$$

The advantage in "rolling" is that we have already removed a lot of the adversarial signal from $\hat{x}$ by producing x_{240}, so each time we "roll" we lose an even greater amount of the signal. The only tricky part is discovering which iteration to use when creating another roll. This was empirically found to be 240 iterations when using the squarred euclidean norm for 3.1. When performing this method we are able to achieve 22% accuracy on the BPDA attack without any adversarial training or task adjustment.

3.4 Adversarial training

Naturally trained classifier are extremely susceptible to adversarial attacks because of their limited input space. The CIFAR-10 training dataset contains only 50k $32x32$

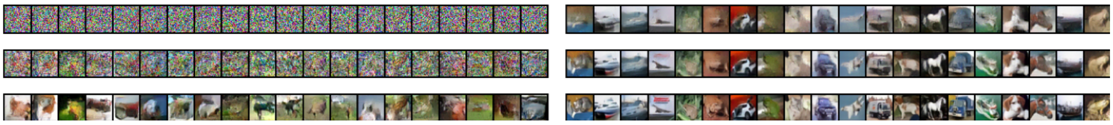

Figure 3.9: (Left) visualization of the first 100 langevin steps for the first reconstruction iteration for K=0 (top), 50 (mid), 100 (bot) using eq 2.12 from noise initialization. (Right) Visualization for reconstruction iteration 50, 150, 240. Note that the bottom right images appears identical to the first 20 images of the CIFAR-10 testing set.

colored images. In order to increase robustness we explicitly train on the adversarial samples provided by the BPDA attack. It is important to note that this method is no longer "add-on purification" because we add the task "defend" to an existing classifier whose primary goal used to be natural image classification.

3.4.1 Network capacity

Network capacity plays a large role in determining the capability of any adversarial trained network because the necessary data distribution to approximate is larger than a simple set of natural images. Not only must the classifier learn the task-dependent natural image samples but it must also learn the space of the attacker (which does not benefit the task!). For this purpose, training experiments were conducted using Res-Net50 [HZR15] and WideResNet-30 [ZK16]. ResNet-50 contains $25x10^6$ trainable parameters and the 30x10 variant of WideResNet-30 contains $45.6x10^6$ params. The adversarial robustness of ResNet-50 is less than WideResNet-30 solely due to training capacity limitations according to [MMS17]. Add-on experiments were performed using WideResNet-18 with $11x10^6$ trainable parameters whereas adversarial training uses WideResNet-30.

3.4.2 BPDA training

In order to provide a training algorithm that encompasses purification we re-formulate the optimization criterion in equation 2.1 to:

$$\min_{\theta} E_{(x,y)\in\bar{q}}[\max_{\delta\in S} L(g(x+\delta|\zeta), y, \theta)] \tag{3.3}$$

where $\bar{q}$ is the distribution of the purified samples and where $g(.)$ is equation 2.12 where ζ are the weights of the DeepFrame model and $\hat{x} = x + \delta$ is the adversarial input returned from equation 2.4.

Our goal is now to train a classifier robust to the BPDA attack and the PGD attack. In doing so, the classifier will be able to perform its original task (natural image classification) while being robust. We note the distribution $p(x|\zeta, \theta)$ as the purified image distribution with respect to $g(x)$ where dom $g = S$ because $x + \delta \in S$. Purified image samples need not necessarily be in S because we place no constraint on the output of $g(.)$. We describe the joint distribution as $\bar{q}(x) = q(x), p(x|\zeta)$.

If we train our model in the purified distribution $\bar{q}(x)$, and attempt to move any adversarial input into that distribution using Langevin dynamics, then the model should be robust in that distribution. We follow training algorithm 2.

Algorithm 2 Adversarial Training using BPDA attack and Purification

```
Require: $X_{i=1}^{m}$ = mini batch of natural images of size m, $K = 1$ langevin updates,
    $T = \#epochs$, $\alpha = 0.5$ adv step size, $\epsilon = \frac{8}{255}$ maximum allowed per-pixel deviation,
    $X_p$ bank of purified samples intialized to $X$, $\eta$ classifier learning rate, $r = 0.01$
    rejuvination rate.
    for t=1:T do
        $X_p \leftarrow g(\tilde{X}_{t-1})$ where $g(.)$ is eq 2.12 for 1 step
        $\theta \leftarrow \theta - \eta \nabla_{x_p} L(X_p, Y, \theta)$ Update weights of classifier $f$ on $X_p, Y$ using gradient descent
        $\hat{X}_t \leftarrow \hat{X}_{t-1} + \alpha \nabla_X L(X_p, Y, \Theta)$ using eq 2.4
        $\tilde{X}_t \leftarrow \max(\min(\hat{X}_t, x + \epsilon), x - \epsilon)$ enforces $l_\infty$ norm by projecting into $\epsilon$-ball of $X$
    end for
```

3.5 Results

In table 3.5 we display the results of reconstruction-based purification (3.3), add-on purification (3.1) and adversarial training 3.4 using non-convergent EBMs. The results are based on the BPDA atack eq (2.4) and the PGD attack eq (2.3). The reconstruction-based purification method is given its own column because it was only

BPDA attack	Reconstruction-based Add-on purification	Add-on purification	Adversarial Training	l_∞
Ours	**22**	9	33	8/255
PixelDefend[SKN17]	-	9	-	8/255
Ma et al[MLW18]	-	5	-	8/255
PGD Attack				
Ours	-	41	40	8/255
Madry et al[MMS17]	-	-	41	8/255
Du et al[DM19]	-	38	-	8/255

Table 3.2: Results of adversarial attacks

run on random subsamples of the CIFAR-10 testing dataset due to computational constraints. All BPDA attacks were run using 100 iterations with step size 1 and $\epsilon = \frac{8}{255}$ to be consistent with [ACW18]. PGD attacks were run for 20 iterations using a step size of $\frac{1}{255}$ to be consistent with [MMS17]. These results demonstrate the difficulty in combating the BPDA attack and the effectiveness of adversarial training. We also note that the accuracy with respect to the original dataset of natural images is 89% for the case of the PGD-trained classifier and 74% for the case of the BPDA trained classifier instead of the usual 99% on natural images. Therefore adversarial training reduces overall generalization accuracy with respect to the original task; image classification.

Add-on purification using energy-based models was attempted in [DM19] for the PGD attack. We assume that these results (38%) are merely due to gradient obfuscation and that they should be evaluated on the BPDA attack instead. We include our results for this task as a means of completeness (41%) with the warrant that it obfuscates the gradient.

3.6 Conclusion

Throughout this process we have demonstrated just how difficult white-box attacks are to combat. Currently there are no defense methods better than training the model on adversarial samples. Add-on purification has been explored since 2018 but the best results are in-line with the method presented here at 9%. While this may not be ideal, it demonstrates a weakness in DNNs that holds vast importance when integrating them into secure applications. Overall, reconstruction-based add-on purification (sec 3.3) presents the best results but it is infeasible for current computational constraints. A further investigation to explore convergent and conditional based EBMs is needed to continue this approach. In theory MCMC should be able to reduce all of the adversarial signal, but that may take an infeasible number of steps to run.

3.7 Future applications

Convergent models are troublesome to train and extremely unstable at inference. However, their use for a means of purification is a further topic to explore because they create steady state samples and can run for a long time without mixing. In the future, we hope to evaluate convergent models on the task of adversarial purification because they demonstrate a lower Pearson correlation between the residuals shown in fig 3.7.

If a stable convergent conditional energy-based model can be trained using the CIFAR-10 dataset, then we can sample from the joint distribution $p(x, y)$ by using the conditional model and summing over the marginal distributions $p(x|y) \forall y \in K$. This may keep sampling in each meta-stable region pure, however it also risks the possibility that a single label might dominate the meta-stable basin which would lead to "unpure" sampling.

REFERENCES

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.", 2018.

[BZ19] Adrian Barbu and Song-Chun Zhu. *Monte Carlo Methods.* Springer, 2019.

[CFG14] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. "Stochastic Gradient Hamiltonian Monte Carlo.", 2014.

[DM19] Yilun Du and Igor Mordatch. "Implicit Generation and Generalization in Energy-Based Models.", 2019.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition.", 2015.

[LZW15] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. "Learning FRAME Models Using CNN Filters.", 2015.

[MLW18] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. "Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality.", 2018.

[MMS17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks.", 2017.

[Nea12] Radford M. Neal. "MCMC using Hamiltonian dynamics.", 2012.

[NHH19] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. "On the Anatomy of MCMC-Based Maximum Likelihood Learning of Energy-Based Models.", 2019.

[NHZ19] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "On Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model.", 2019.

[RDS14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge.", 2014.

[SKC18] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models.", 2018.

[SKN17] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. “PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples.”, 2017.

[SLJ14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going Deeper with Convolutions.”, 2014.

[WXZ18] Yingnian Wu, Jianwen Xie, and Song-Chun Zhu. “Sparse and Deep Generalizations of the FRAME model.” *Annals of Mathematical Sciences and Applications*, 2018.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks.”, 2016.