Robust Universal Adversarial Perturbations

Changming Xu
Department of Computer Science
University of Illinois Urbana-Champaign
Champaign, IL 61820
cx23@illinois.edu

Gagandeep Singh
Department of Computer Science
University of Illinois Urbana-Champaign
Champaign, IL 61820
ggnds@illinois.edu

Abstract

Universal Adversarial Perturbations (UAPs) are imperceptible, image-agnostic vectors that cause deep neural networks (DNNs) to misclassify inputs from a data distribution with high probability. Existing methods do not create UAPs robust to transformations, thereby limiting their applicability as a real-world attacks. In this work, we introduce a new concept and formulation of robust universal adversarial perturbations. Based on our formulation, we build a novel, iterative algorithm that leverages probabilistic robustness bounds for generating UAPs robust against transformations generated by composing arbitrary sub-differentiable transformation functions. We perform an extensive evaluation on the popular CIFAR-10 and ILSVRC 2012 datasets measuring robustness under human-interpretable semantic transformations, such as rotation, contrast changes, etc, that are common in the real-world. Our results show that our generated UAPs are significantly more robust than those from baselines.

1 Introduction

Deep neural networks (DNNs) have achieved impressive results in many application domains such as natural language processing [1, 9], medicine [16, 17], and computer vision [36, 38]. Despite their performance, they can be fragile in the face of *adversarial* perturbations [37, 19, 30, 29]: small imperceptible changes added to a correctly classified input that make the DNN misclassify. While there is a large amount of work on generating adversarial examples [19, 30, 29, 10, 41, 15, 12, 40, 45, 4, 13, 39], the vulnerabilities exposed by most existing works are not exploitable by a practical adversary. This is because they depend upon unrealistic assumptions about the power of the attacker: the attacker knows the DNN input in advance, generates input-specific perturbation in real-time and *exactly* combines the perturbation with the input before it is processed by the DNN. To expose exploitable vulnerabilities, considering a practically feasible threat model is necessary.

Practically feasible threat model In this work, we limit the powers of the attacker. We assume that the attacker (i) does not know the DNN inputs in advance and (ii) the generated perturbations can be modified before combination with inputs occurs due to real-world effects. Examples of attacks in our threat model include adding stickers to the cameras for fooling image classifiers [27] or transmitting perturbations over the air for deceiving audio classifiers [26].

The first requirement in our threat model can be fulfilled by generating Universal Adversarial Perturbations (UAPs) [31]. Here the attacker can train a single adversarial perturbation that has a high probability of generating adversarial examples on all inputs by training the perturbation on the publicly available dataset. However, as our experimental results show, the generated universal perturbations need to be combined exactly otherwise they do not remain adversarial. Changes to UAPs are likely in practice due to real-world effects. For example, the stickers applied to a camera can undergo changes in contrast due to weather conditions or the transmitted perturbation in the

Preprint. Under review.

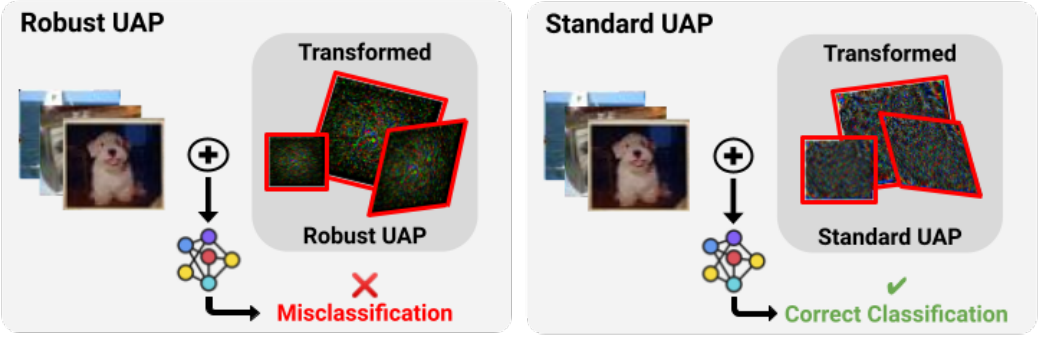

Figure 1: Robust UAPs cause a classier to misclassify on *most* of the data distribution even after transformations are applied on them. Standard UAPs are not robust to transformations and have a low probability of remaining UAPs after transformation.

audio case can change due to noise in the transmission channel. This makes existing UAP generation methods [31, 27, 26] unsuitable for generating exploitable perturbations.

This work: Robust UAPs. In this paper, we propose the novel concept of robust UAPs: universal perturbations that have a high probability of remaining universally adversarial even after applying a set of transformations on them. This enables the attacker to not rely on exactly combining their UAPs with the input. The main challenge in generating robust UAPs is that the optimization problem is harder than generating standard UAPs [31] as we are looking for perturbations that are adversarial for a set of inputs as well as to transformations applied to the perturbations. To address this challenge and generate perturbations effective in our threat model, we make the following main contributions:

- We introduce the novel concept of *Robust UAP* and pose its generation as an optimization problem.
- We design a new iterative algorithm for training robust UAPs, which we call `RobustUAP`, based upon combining Expectation over Transformation (EoT) [5] and standard UAPs [31]. At each iteration during training, our algorithm uses bounds based on Chernoff inequality [11] to ensure the robustness of perturbation under transformations and for the current batch of inputs.
- We perform an extensive evaluation of the effectiveness of `RobustUAP` for generating robust UAPs on state-of-the-art models for the popular CIFAR-10 [25] and ILSVRC 2012 [14] datasets. We compare the robustness of the generated UAPs under composition of challenging semantic transformations, such as rotation, contrast change, etc., showing that on both datasets the UAPs generated by `RobustUAP` are significantly more robust, achieving up to 12% more robustness, than the UAPs generated from the baselines.

2 Background

In this section, we provide necessary background definitions and notation for our work.

Adversarial Examples and Perturbations. An adversarial example is a misclassified data point that is *close* (in some norm) to a correctly classified data point [19, 29, 10]. Let $\mu \subset \mathbb{R}^d$ be the input data distribution, $x \in \mu$ be an input point with the corresponding true label $y \in \mathbb{R}$, and $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ be our target classifier. For ease of notation, we define $f_k(x)$ to be the k^{th} element of $f(x)$ and allow $\hat{f}(x) = \arg\max_k f_k(x)$ to directly refer to the classification label. We use v to reference per image perturbations and u to reference universal adversarial perturbations, v_r and u_r refer to the robust variants and will be defined in Sec. 4. We can now formally define an adversarial example.

Definition 2.1. Given a correctly classified point x, a distance function $d(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and bound $\epsilon \in \mathbb{R}$, x' is an *adversarial example* iff $d(x', x) < \epsilon$ and $\hat{f}(x') \neq y$.

In this paper, we consider examples x' generated as $x' = x + v$ where v is an *adversarial perturbation*.

Universal Adversarial Perturbations. UAPs are single vector, input-agnostic perturbations [31]. They differ from traditional adversarial attacks, which create perturbations dependent on each input sample. To measure UAP performance, we introduce the notion of universal adversarial success rate

for a perturbation over a data distribution, which measures the probability that a perturbation u when added to x, sampled from μ, causes a change in classification under f.

Definition 2.2. Given a data distribution μ, and perturbation u, *universal adversarial success rate* ASR_U for u, is defined as

$$\text{ASR}_U(f, \mu, u) = \underset{x \sim \mu}{P}(\hat{f}(x+u) \neq \hat{f}(x)) \tag{1}$$

Using Definition 2.2, we formally define a UAP.

Definition 2.3. A *universal adversarial perturbation* is a vector $u \in \mathbb{R}^d$ which, when added to almost all datapoints in μ causes the classifier f to misclassify. Formally, given γ, a bound on universal adversarial success rate, and l_p-norm with corresponding bound ϵ, u is a UAP iff $\text{ASR}_U(f, \mu, u) > \gamma$ and $||u||_p < \epsilon$.

In general, if the additive perturbations have small l_p-norm , then they look like noise and do not affect the semantic content of the image. For ease of notation in later parts of the paper, we can also pose the construction of UAPs as an expectation minimization problem:

$$\arg\min_u \mathbb{E}_{x \sim \mu}[\delta(\hat{f}(x+u), \hat{f}(x))] \text{ s.t. } ||u||_p < \epsilon \tag{2}$$

where δ is the Kronecker Delta function [2].

Semantic Transformations. While our concept and formulation of Robust UAPs is general and can be used to generate UAPs robust under arbitrary bijective sub-differentiable transformation functions and for classifiers in a variety of domains including text, speech, and wireless, we focus on measuring robustness under a set of semantic transformations to perturbations for image classifiers. The semantic perturbations have human-interpretable meaning and naturally occur in the real-world. In this paper, we will consider five popular semantic transformations well represented in existing literature [5, 7]: brightness/contrast, rotation, scaling, shearing, and translation. When attacking image classifiers, additive perturbations may undergo such transformations, e.g. a sticker can be rotated. Further details about the semantic transformations used in this paper are in Appendix A.

3 Related Work

In this section, we survey works closely related ours.

Set of UAPs. Most works focusing on UAPs [31, 32, 21, 43] generate singular vectors. [6] introduce a perturbation generator model (PGM) for the wireless domain which creates UAPs with random trigger patterns. They show that both adversarial training and noise subtracting defenses used in the wireless domain are highly effective in mitigating the effects of a single vector UAP attack; they further show that their method of generating a set of UAPs is an effective way for an attacker to circumvent these defenses. Although PGM provides a method for efficiently sampling unique UAPs, our methods generate more powerful UAPs that are robust to transformations as well as allowing for efficient sampling.

Geometric/Semantic Adversarial Attacks. Semantic adversarial attacks have gained popularity as researchers have realized they are more interpretable measures of closeness that do not affect the semantic content of an input [44, 42, 22, 8, 33]. Robust UAPs are l_p-norm bounded perturbations and are not created by applying a semantic transformation to the target input, instead we are defining the perturbation itself as robust to semantic transformations.

Robust Adversarial Examples. The following papers all introduce notions of robustness under different viewpoints and environmental conditions for constructing realizable adversarial examples in comparison to the additive perturbations discussed in this paper. Luo et al. [28] constructs adversarial examples which minimize human detectability, they further introduce the idea of robustness for adversarial examples. They show that their attacks are robust against jpeg compression. Sharif et al. [35] attack facial recognition systems by putting adversarial perturbations on glass frames. Their work demonstrates a successful physical attack under stable conditions and poses. Eykholt et al. [18] proposes Robust Physical Perturbations (RP_2) in order to show that adding graffiti on a stop sign can cause it to be misclassified in both simulations and in the real world. Athalye et al. [5] introduce

Expectation over Transformation (EOT) and use it to print real-world objects which are adversarial given a range of physical and environmental conditions.

Robust Adversarial Perturbations. Li et al. [26] generates music which affects a voice assistant based system from picking up its wake word. Li et al. [27] presents a method for generating a targeted adversarial sticker which changes an image classifier's classification from one pre-specified class to another. Both of these methods rely on specific use cases and are tailored towards generating adversaries coming from strict distributions, e.g. [26] generates guitar music while [27] generates a small grid of dots. Robust UAPs generalize to a much wider range of applications and are both robust under a set of transformations but also on an entire data distribution.

4 Robust Universal Adversarial Perturbations

In this section, we will define the optimization problem for generating robust UAPs. We first define transformation sets and neighborhoods.

Definition 4.1. A *transformation*, τ, is a composition of bijective sub-differentiable transformation functions. A *transformation set*, T, is a set of distinct transformations.

Using Definition 4.1 we can define a neighborhood.

Definition 4.2. A point v' is in the *neighborhood* $N_T(v)$, of v, if there is a transform in T that maps v to v'. Formally,

$$v' \in N_T(v) \iff \exists \tau \in T \text{ s.t. } \tau(v) = v' \tag{3}$$

Example 4.3. Let T be the set of all transformations represented by a rotation of $\pm 30°$, scaling of up to a factor of 2, and a translation of up to ± 2 pixels, in this case one $\tau \in T$ could be {rotation of $8°$, scaling a factor of 1.2, and translation of -1.3} in that order and $N_T(v)$ would include any point that can be obtained by applying one of the transformations from T on v.

In order to define robust UAPs we introduce robust universal adversarial success rate.

Definition 4.4. Given a data distribution μ, transformation set T, universal adversarial success rate level γ, bound ϵ on l_p-norm, and perturbation u_r, *robust universal adversarial success rate*, ASR_R, is defined as,

$$\text{ASR}_R(f, \mu, T, \gamma, u_r) = \underset{u'_r \sim N_T(u_r)}{P} (\text{ASR}_U(f, \mu, u'_r) > \gamma \wedge ||u'_r||_p < \epsilon) \tag{4}$$

The *robust universal adversarial success rate* measures the probability that a neighbor of u_r is also an UAP on μ, i.e. after transformation it maintains high universal ASR. We note that even though $||u_r||_p \leq \epsilon$, it can happen that a $u'_r \in N_T(u_r)$ has $||u'_r||_p > \epsilon$, this is particularly true for the semantic transformations considered in this work. Therefore, we also require that the norm of u'_r is small.

Using Definition 4.4 we can now formally define a robust UAP.

Definition 4.5. A *robust universal adversarial perturbation*, u_r, is one which *most* points within a neighborhood of u_r when added to *most* points in μ fool the classifier, f. u_r satisfies $||u_r||_p < \epsilon$ and $\text{ASR}_R(f, \mu, T, \gamma, u_r) > \zeta$.

In order to construct robust UAPs, we can pose the following expectation minimization problem:

$$\arg\min_{u_r} \underset{u'_r \in N_T(u_r)}{\mathbb{E}} [I(||u'_r|| < \epsilon) \times \underset{x \sim \mu}{\mathbb{E}} [\delta(\hat{f}(x + u'_r), \hat{f}(x))]] \text{ s.t. } ||u_r||_p < \epsilon \tag{5}$$

Here I denotes an indicator function. The inside expectation represents the UAP condition for the transformed perturbation u'_r while the outside expectation represents the neighborhood robustness condition. Solving Equation 5 requires computing a perturbation u_r which minimizes the expectation over the transformation set and data distribution. This composition makes it computationally harder than minimizing over only the transformation set, as in EOT [5], or than minimizing over only the data distribution, as done for standard UAP [31].

5 Generating Robust Universal Adversarial Perturbations

In this section, we will discuss our approach for optimizing Equation 5 to generate UAPs robust to transformations generated by a composition of arbitrary sub-differentiable transformation functions. From a high level, the objective can be seen as gluing the outer expectation, a EOT objective over the transformations applied on the perturbation, with the inner expectation, a UAP objective over the input data distribution. We first discuss straightforward baselines for optimizing Equation 5 and then present our new algorithm.

5.1 Stochastic Gradient Descent

The first baseline approach is to directly solve Equation 5 using gradient descent. Since we are solving a constrained optimization problem we can not directly use gradient descent. Instead, we can solve the Lagrangian-relaxed form of the problem similar to [10, 5].

$$\arg\min_{u_r} \mathop{\mathbb{E}}_{u'_r \in N_T(u_r)} [\mathop{\mathbb{E}}_{x \sim \mu} [\delta(\hat{f}(x + u'_r), \hat{f}(x))]] - \lambda ||u_r||_p \tag{6}$$

We use a momentum based Stochastic Gradient Descent (SGD) method for solving Equation 6. Shafahi et al. [34] suggests that this is an effective method for generating standard UAPs. In order to implement this, we replace the Kronecker Delta function with a loss function, L. We iteratively converge towards the inner expectation by computing it in batches, and towards the outer expectation by sampling a large number of transformations. Given that we would like to estimate on a batch, $\hat{x} \subset \mu$, and a random set of transformations sampled from T, $\hat{\tau} \subset T$, we can approximate Equation 6:

$$\mathop{\mathbb{E}}_{u'_r \in N_T(u_r)} [\mathop{\mathbb{E}}_{x \sim \mu} [L[f(x + u'_r), f(x)]]] \approx \frac{1}{|\hat{x}| \times |\hat{\tau}|} \sum_{i=1}^{|\hat{x}|} \sum_{j=1}^{|\hat{\tau}|} L[f(\hat{x}_i + \hat{\tau}_j(u_r)), f(\hat{x}_i)] \tag{7}$$

Our final algorithm is in Appendix C.

5.2 Standard UAP Algorithm with Robust Adversarial Perturbations

For our second baseline, we leverage the standard UAP algorithm from Moosavi-Dezfooli et al. [31] (see Appendix D for the algorithm). The standard UAP algorithm iterates over the entire training dataset and at each input, x_i, it computes the smallest additive change, Δu, to the current perturbation, u, that would make $u + \Delta u$ an adversarial perturbation for x_i. Intuitively, over time the algorithm will approach a perturbation that works on most inputs in the training dataset. Our second baseline approach works by computing robust adversarial perturbations rather than standard adversarial perturbations. At each point x_i, we compute the smallest additive change, Δu_r, to the current robust perturbation, u_r, that would make $u_r + \Delta u_r$ a robust adversarial perturbation for x_i.

We can search for robust adversarial perturbations by optimizing the expectation that a point in the neighborhood of v_r is adversarial while restricting the perturbation to an l_p norm of ϵ. We can formulate this as the following minimization problem:

$$\arg\min_{v_r} \mathop{\mathbb{E}}_{v'_r \in N_T(v_r)} [\delta(\hat{f}(x + v'_r), \hat{f}(x))] \text{ s.t. } ||v_r||_p < \epsilon \tag{8}$$

We can generate robust adversarial perturbations using Projected Gradient Descent by relaxing the optimization problem posed in Equation 8:

$$\arg\min_{v_r} \mathop{\mathbb{E}}_{v'_r \in N_T(v_r)} [\delta(\hat{f}(x + v'_r), \hat{f}(x))] - \lambda ||v_r||_p \tag{9}$$

5.3 Robust UAP Algorithm

The baseline algorithms have two fundamental limitations: (i) they rely on random sampling over the symbolic transformation region but the sampling strategy does not explicitly try to maximize the

Algorithm 1 EstimateRobustness

Draw $n = \lceil \frac{1}{2\psi^2} \ln \frac{2}{\phi} \rceil$ samples $\tau_i \sim T$
Compute $\hat{p}_n(\gamma) = \frac{1}{n} \sum_i^n I(ASR_U(f, X, \tau_i(u_r)) > \gamma)$
Return $\hat{p}_n(\gamma)$

robustness of the generated UAP over the entire symbolic region, and (ii) they do not have a method for estimating robustness on unseen transformations. As a result, the baselines yield suboptimal UAPs (as confirmed by our experiments). To overcome these fundamental limitations, we need a way to compute probabilistic bounds for expected robustness on the entire symbolic region. We next present a method to obtain these probabilistic bounds. We do not want to limit ourselves to handling specific transformation sets. Computing expected robustness for general transformation sets requires reasoning about arbitrary distributions. Therefore, we make a simplifying assumption that $N_T(u_r)$ has a well defined probability distribution function (PDF) to approximate the bounds on the expected robustness. We will leverage our method for approximating expected robustness in our new algorithm for generating robust UAPs. Our experiments show that even though our assumptions may not hold for the transformation sets considered in this work they significantly improve the robustness of our generated UAPs. Our approximation of the expected robustness relies on the following theorem:

Theorem 5.1. *Given a perturbation u_r, a neural network f, a finite set of inputs X, a set of transformations T, and minimum universal adversarial success rate $\gamma \in \mathbb{R}$. Let $p(\gamma) = P_{u'_r \sim N_T(u_r)}(ASR_U(f, X, u'_r) > \gamma)$. For $i \in 1 \dots n$, let $u_r^i \sim N_T(u_r)$ be random variables with a well defined PDF and I be the indicator function, let*

$$\hat{p}_n(\gamma) = \frac{1}{n} \sum_{i=1}^{n} I(ASR_U(f, X, u_r^i) > \gamma)$$

For accuracy level, $\psi \in (0, 1)$, and confidence, $\phi \in (0, 1)$, where $(0, 1)$ is the open interval between 0 and 1. If $n \geq \frac{1}{2\psi^2} \ln \frac{2}{\phi}$ then

$$P(|\hat{p}_n(\gamma) - p(\gamma)| < \psi) \geq 1 - \phi \tag{10}$$

Proof. We first construct a function u which estimates the probability of u_r being adversarial on the current batch X:

$$u(u'_r) = \frac{1}{|X|} \sum_{i=1}^{|X|} \delta(f(X_i + u'_r), f(X_i)) \tag{11}$$

The result from Alippi [3] states that, given $n \geq \frac{1}{2\psi^2} \ln \frac{2}{\phi}$, where the bound on n is derived via Chernoff inequality [11], Equation 10 holds if u is Lebesgue measurable over a distribution Ψ with a well defined PDF. In our case, $\Psi = N_T(u_r)$ has a well-defined PDF. □

Algorithm 2 Robust UAP Algorithm

Initialize $u_r \leftarrow 0, n \leftarrow \lceil \frac{1}{2\psi^2} \ln \frac{2}{\phi} \rceil$
repeat
  for $B \subset X$ **do**
    if `EstimateRobustness`$(f, B, T, \gamma, u_r, \psi, \phi) < \zeta$ **then**
      $\Delta u_r \leftarrow 0$
      repeat
        For $i = 1 \dots n$ sample $\tau_i \sim T$
        Compute $L_{B,\tau} = \frac{1}{|B| \times n} \sum_{i=1}^{|B|} \sum_{j=1}^{|n|} L[f(B_i + \tau_j(u_r + \Delta u_r)), f(B_i)]$
        $\Delta u_r = \mathcal{P}_{p,\epsilon}(\Delta u_r + \alpha \text{sign}(\nabla L_{B,\tau}))$
      until `EstimateRobustness`$(f, B, T, \gamma, u_r + \Delta u_r, \psi, \phi) < \zeta$
      Update the perturbation with projection:
      $u \leftarrow \mathcal{P}_{p,\epsilon}(u_r + \Delta u_r)$
    end if
  end for
until `EstimateRobustness`$(f, X, T, \gamma, u_r, \psi, \phi) < \zeta$

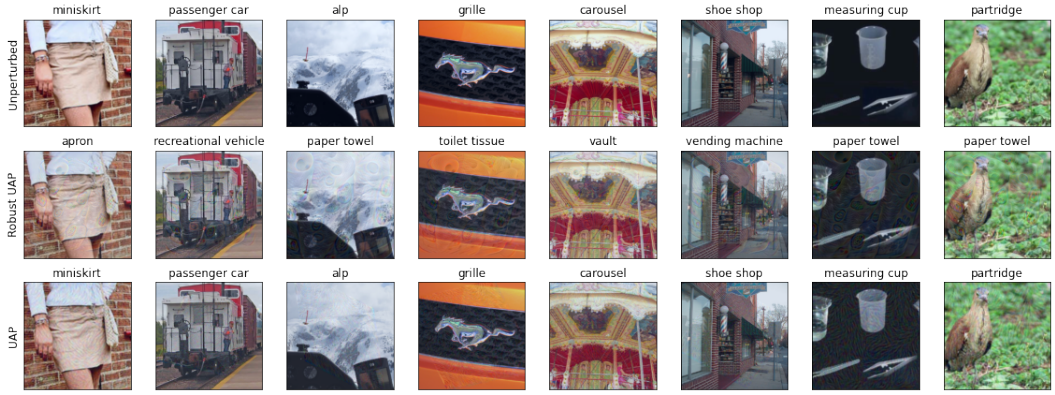

Figure 2: Examples of perturbed images with labels. The top row are unperturbed images from the ILSVRC 2012 test set, the second row has a randomly transformed Robust UAP added to it, and the bottom row has a randomly transformed standard UAP added to it. All labels are calculated using Inception-v3 [38].

Theorem 5.1 states that with enough samples from the neighborhood of a perturbation, u_r, we know that with probability at least $1 - \phi$ that the adversarial success rate of u_r on the entire neighborhood is arbitrarily close to the adversarial success rate of u_r on sampled transformations. Transformations such as brightness or contrast changes yield neighborhood regions with well-defined PDFs we can sample from and thus allow us to obtain provable bounds on the robustness with the above theorem. For the combinations of semantic transformations, such as rotation, translation, etc. used in the experiment section the neighborhood does not have a well-defined PDF, thus we uniformly sample the parameter space of each transformation to produce a point in the neighborhood. We believe uniform transformations of parameters is feasible in the real world. Although this means that for the transformation sets we are considering, the assumption in Theorem 5.1 does not hold, we still use its result to inform our sampling.

Leveraging Theorem 5.1, we create a method which given accuracy, ψ, and confidence, ϕ, returns the robust adversarial success rate on a finite set of inputs with probabilistic guarantees under the assumptions of the Theorem. The pseudocode for `EstimateRobustness` is in Algorithm 1

Now we leverage Theorem 5.1 and Algorithm 1 to develop our `RobustUAP` algorithm. Similar to the SGD Baseline we approximate the expectation in Equation 5 in batches. However, in our case the $|\hat{t}| = n$ which is based on Theorem 5.1 to satisfy the desired confidence level and accuracy. Furthermore, we do not take only a single gradient step, but instead take multiple steps, using PGD, until the robustness property is satisfied on the given batch. We use `EstimateRobustness` to compute the probabilistic bound for robustness over the transformation space. At the end of each epoch, we use `EstimateRobustness` to check the robustness across the entire training set and transformation space. The psuedocode for `RobustUAP` is in Algorithm 2.

6 Evaluation

Our Robust UAP framework works for any transformation set in a variety of domains. We empirically evaluate the methods on popular models from the vision domain by generating semantically robust perturbations as we believe these perturbations are more likely to occur in a real-world transmission of the perturbation. We compute robust UAPs using the three proposed robust methods (`SGD`, `StandardUAP_RP`, `RobustUAP`) and compare them against the standard UAP algorithm [31] (`StandardUAP`).

Experimental evaluation. We consider two popular image recognition datasets: CIFAR-10[25] and ILSVRC 2012[14]. For CIFAR-10, we evaluate on the entire test set (1,000 images) and use a state-of-the-art pretrained VGG16 [36] network as the target classification model. For ILSVRC 2012, we evaluate on a random subset of the test set (1,000 images), and use a state-of-the-art

Inception-v3 [38] network. We leverage a composition of transformations from brightness/contrast, rotation, scaling, shearing, and translation. All experiments were performed on a desktop PC with a GeForce RTX(TM) 3090 GPU and a 16-core Intel(R) Core(TM) i9-9900KS CPU @ 4.00GHz.

We report the results for l_2-norm with $\epsilon = 100$ for ILSVRC 2012 and $\epsilon = 10$ for CIFAR-10. These values were chosen based on the values presented by the original UAP paper [31]. We use an image normalization function given by our pretrained models and thus scaled our ϵ values accordingly. We note that the ϵ-values are significantly smaller than the image norms. Therefore the generated perturbation is imperceptible and does not affect the semantic content of the image. Due to the hardness of the optimization problem, for the same norm value, the effectiveness of a UAP is less than input-specific perturbations. We note that crafting input-specific perturbations requires making unrealistic assumptions about the power of the attacker as mentioned in the introduction and therefore we do not consider them part of our threat model which aims to generate practically feasible perturbations.

We use a variety of different transformation sets in order to show that our algorithm works under different conditions and base our parameters for the transformations on [7]. For our experiments, $R(\theta)$ corresponds to rotations with angles between $\pm\theta$; $T(x, y)$, to translations of $\pm x$ horizontally and $\pm y$ vertically; $Sc(p)$ to scaling the image between $\pm p\%$; $Sh(m)$ to shearing by shearing factor between $\pm m\%$; and $B(\alpha, \beta)$ to changes in contrast between $\pm\alpha\%$ and brightness between $\pm\beta$. We consider compositions of different subsets and ranges of these transformations shown in Table 2 including composing all transformations together. The hardness of generating robust UAP increases with the number of transformations in the composition as well as the range of parameters for each individual transformation. For example, generating robust UAP is harder for the composition shown in the first and last row for ILSVRC 2012 in Table 2 compared to the second and third row. We use $\psi = 0.1$ and $\phi = 0.05$ resulting in $n = 185$ for generating samples for our `RobustUAP` algorithm as well as reporting robust ASR in our evaluation. The UAPs are trained on 2,000 images, other parameters for evaluation are given in Appendix F.

6.1 Semantically Robust UAPs

Next, we analyze the effectiveness of our robust UAPs.

Visualization. We visualize our UAPs added to images in ILSVRC 2012 in Figure 2, with the robust UAPs generated with our `RobustUAP` algorithm. The UAP is trained on a difficult transformation set containing all transformations $R(10), T(2, 2), Sh(2), Sc(2), B(2, 0.001)$. We randomly apply a transformation from the set before adding to the input in the Figure. We observe that our robust UAPs have a similar level of imperceptibility as standard UAPs and do not affect the semantic content of the images. While standard UAPs do not have a high probability of affecting the model classification after transformation, our Robust UAPs do.

We further visualize UAPs generated with our three robust algorithms on the same transformation set against a standard UAP [31] generated on ILSVRC 2012 in Figure 3. We observe that the UAP generated by the `StandardUAP` algorithm resembles the one generated by the `StandardUAP_RP` algorithm. We believe that this is due to the similarity in the workings of both algorithms. However, the two UAPs are not identical as the `StandardUAP_RP` algorithm concentrates its perturbation towards the center of the image as under our transformation set, this area is least perturbed. Both the `RobustUAP` and the `SGD` algorithm generate larger patterns distributed over the entire image.

Table 1: Universal ASR of our Robust UAP algorithms and the standard UAP [31] method.

DATASET	STANDARDUAP	SGD	STANDARDUAP_RP	ROBUSTUAP
ILSVRC 2012	**95.5**%	85.6%	82.3%	91.3%
CIFAR-10	**87.6**%	77.1%	76.0%	82.9%

Universal ASR ($\mathbf{ASR_U}$). We first compare our robust UAPs to standard UAPs on the non-robust universal ASR metric, see Table 1. We observe that at the same l_2-norm all robust UAPs achieve a lower universal ASR than the standard UAP algorithm. This result is not too surprising as solving the optimization problem for robust UAP is significantly more difficult. We further observe that our `RobustUAP` algorithm is the most effective in comparison to the other robust baseline approaches.

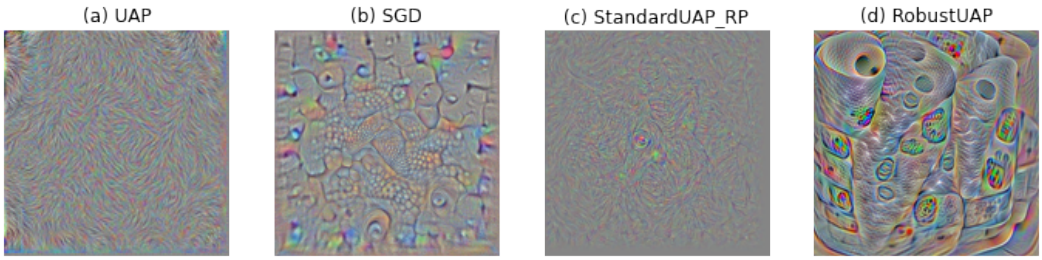

Figure 3: Comparison of UAPs generated with (a) `StandardUAP` [31], (b) `RobustUAP`, (c) `StandardUAP_RP`, and (d) `RobustUAP` on ILSVRC 2012.

Robust ASR ($\mathbf{ASR_R}$). In Table 2 we compare robust universal adversarial success rate ASR_R with $\gamma = 0.6$, in other words, we are finding the percentage of sampled neighbors of the perturbation that are still UAPs with 60% effectiveness on the testing set. We provide average ASR_U scores as well as ASR_R for different γ levels in Appendix E. In contrast to universal ASR, we observe that our `RobustUAP` algorithm achieves much higher robust ASR when compared to the standard UAP algorithm on both datasets and the challenging transformation sets shown in Table 2. Furthermore, our `RobustUAP` algorithm significantly outperforms both robust baseline approaches. We observe that the baseline `SGD` achieves high robust ASR on relatively easier transformation sets which have fewer transformations and have smaller range for the parameters of the individual transformations. Its robust ASR is significantly lower than `RobustUAP` on harder to optimize transformation sets.

Table 2: Robust ASR of our Robust UAP algorithms to the standard UAP [31] method.

Dataset	Transformation Set	Standard UAP	SGD	Standard UAP_RP	Robust UAP
ILSVRC 2012	$R(20)$	0.0%	70.1%	4.2%	$\mathbf{82.2}\%$
	$T(2,2)$	38.6%	98.8%	45.3%	$\mathbf{99.2}\%$
	$Sc(5), R(5), B(5, 0.01)$	12.1%	94.7%	40.2%	$\mathbf{95.0}\%$
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	0.0%	72.3%	5.9%	$\mathbf{81.8}\%$
CIFAR-10	$R(30), B(2, 0.001)$	0.0%	65.5%	3.1%	$\mathbf{76.3}\%$
	$R(2), Sh(2)$	45.1%	98.0%	52.3%	$\mathbf{98.1}\%$
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	0.0%	69.6%	8.0%	$\mathbf{78.9}\%$

6.2 Additional Experiments

We additionally evaluate our methods on ResNet18 [20] and MobileNet [23] for CIFAR-10 and ILSVRC 2012 respectively in Appendix G. The results follow the same trends as those reported in Table 2. In Appendix E we provide the average ASR_U achieved by all the algorithms and also provide ASR_R computed with different values of γ for the same transformation sets in Table 2. Finally, we provide runtimes for all algorithms in Appendix H.

7 Conclusion

In this paper, we define a novel attack scenario that requires creating UAPs robust to real-world transformations. We demonstrate that standard UAPs are highly susceptible to transformations, i.e. they fail to be universally adversarial under transformation. We propose a new method, `RobustUAP` to generate robust UAPs based upon obtaining probabilistic bounds on UAP robustness across an entire transformation space. Our experiments provide empirical evidence that this principled approach generates UAPs that are practically more robust under a set of semantic transformation sets than those from the baseline methods.

References

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[2] DC Agarwal. *Tensor Calculus and Riemannian Geometry*. Krishna Prakashan Media, 2013.

[3] Cesare Alippi. *Intelligence for embedded systems*. Springer, 2014.

[4] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search, 2019.

[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[6] Alireza Bahramali, Milad Nasr, Amir Houmansadr, Dennis Goeckel, and Don Towsley. Robust adversarial attacks against dnn-based wireless communication systems. *arXiv preprint arXiv:2102.00918*, 2021.

[7] Mislav Balunović, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. *Advances in Neural Information Processing Systems 32*, 2019.

[8] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 2019.

[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[10] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[11] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[12] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack, 2019.

[13] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[15] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[17] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[20] Kaiming He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition." computer vision and pattern recognition (2015). *Google Scholar There is no corresponding record for this reference*, pages 770–778, 2015.

[21] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 2755–2764, 2017.

[22] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018.

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[26] Juncheng Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J. Zico Kolter, and Florian Metze. Adversarial music: Real world audio adversary against wake-word detection system. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 11908–11918, 2019.

[27] Juncheng Li, Frank R. Schmidt, and J. Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *Proc. International Conference on Machine Learning, ICML*, volume 97, pages 3896–3904, 2019.

[28] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Thirty-second aaai conference on artificial intelligence*, 2018.

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[32] Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*, 2017.

[33] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *European Conference on Computer Vision*, pages 19–37. Springer, 2020.

[34] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643, 2020.

[35] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[39] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *CoRR*, abs/2002.08347, 2020.

[40] Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Enhancing gradient-based attacks with symbolic intervals, 2019.

[41] Chaowei Xiao, Bo Li, Jun yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 3905–3911, 2018.

[42] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

[43] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Cd-uap: Class discriminative universal adversarial perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6754–6761, 2020.

[44] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

[45] Tianhang Zheng, Changyou Chen, and Kui Ren. Distributionally adversarial attack. *Proc. AAAI Conference on Artificial Intelligence*, 33:2253–2260, 07 2019.

Appendix

A Semantic Transformations

In this section, we discuss the semantic transformations used in the paper. Brightness and contrast can be represented via *bias* (β) and *gain* ($\alpha > 0$) parameters respectively. Formally, if x is the original image, then the transformed image, x', can be represented as

$$x' = \alpha x + \beta \tag{12}$$

Rotation, scaling, shearing, and translation are all affine transformations acting on the coordinate system, c, of the images instead of the pixel values, x. In order to recover the pixel values and differentiate over the transformation, we will need sub-differentiable interpolation, see Appendix B. For finite dimensions, affine transformations can be represented as a linear coordinate map where the original coordinates are multiplied by an invertible augmented matrix and then translated with additional bias vector. Below, we give the general form for an affine transformation given augmented matrix A, bias matrix b, and input coordinates c. We can compute the output coordinates, c', as

$$\begin{bmatrix} \mathbf{c'} \\ 1 \end{bmatrix} = \begin{bmatrix} [ccc|c] & A & & \mathbf{b} \\ 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ 1 \end{bmatrix} \tag{13}$$

Below, we give the augmented matrix A and additional bias matrix b for rotation, scaling, shearing, and translation.

Rotation, $R(\theta)$, by θ degrees:

$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{14}$$

Scaling, $Sc(p)$, by $p\%$:

$$A = \begin{pmatrix} 1 + \frac{p}{100} & 0 \\ 0 & 1 + \frac{p}{100} \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{15}$$

Shearing, $Sh(m)$, by shear factor $m\%$:

$$A = \begin{pmatrix} 1 & 1 + \frac{m}{100} \\ 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{16}$$

Translation, $T(x, y)$, by x pixels horizontally and y pixels vertically:

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, b = \begin{pmatrix} x \\ y \end{pmatrix} \tag{17}$$

B Interpolation

Affine transformations may change a pixel's integer coordinates into non-integer coordinates. Interpolation is typically used to ensure that the resulting image can be represented on a lattice (integer) pixel grid. For this paper, we will be using bilinear interpolation, a common interpolation method which achieves a good trade-off between accuracy and efficiency in practice and is commonly used in literature [42, 7]. Let $x_{i,j}$, $x'_{i,j}$ represent the pixel value at position i, j for the original and transformed image respectively. Let ${c'}^{x}_{i,j}$, ${c'}^{y}_{i,j}$ represent the x-coordinate and y-coordinate of the pixel at i, j after transformation. We define our transformed image by summing over all pixels $n, m \in [1 \dots H] \times [1 \dots W]$ where H and W represent the height and width of the image.

$$x'_{i,j} = \sum_{n}^{H} \sum_{m}^{W} x_{n,m} \max(0, 1 - |{c'}^{x}_{i,j} - m|) \max(0, 1 - |{c'}^{y}_{i,j} - n|) \tag{18}$$

Algorithm 3 Stochastic Gradient Descent UAP Algorithm

Initialize $u_r \leftarrow 0, \Delta u_r \leftarrow 0$
repeat
  for $B \in X$ **do**
    Sample $\hat{t} \subset T$
    $\Delta u_r \leftarrow \alpha \Delta u_r - \frac{\nu}{|\hat{x}| \times |\hat{t}|} \sum_{i=1}^{|\hat{x}|} \sum_{j=1}^{|\hat{t}|} \nabla L[f(\hat{x}_i + \hat{t}_j(u_r)), f(\hat{x}_i)]$
    Update the perturbation with projection:
    $u \leftarrow \mathcal{P}_{p,\epsilon}(u_r + \Delta u_r)$
  end for
until $ASR_R(f, X, T, \gamma, u_r) < \zeta$

Algorithm 4 Iterative Universal Perturbation Algorithm (Moosavi-Dezfooli et al. [31])

Initialize $u \leftarrow 0$
repeat
  for $x_i \in X$ **do**
    if $\hat{f}(x_i + u) = \hat{f}(x_i)$ **then**
      Compute minimal adversarial perturbation:
      $\Delta u \leftarrow \arg\min_r ||r||_2$ s.t. $\hat{f}(x_i + u + r) \neq \hat{f}(x_i)$
      Update the perturbation with projection:
      $u \leftarrow \mathcal{P}_{p,\epsilon}(u + \Delta u)$
    end if
  end for
until $ASR_U(f, X, u) < \gamma$

This interpolation can be computed for each channel in the image. While interpolation is typically not differentiable, in order to generate adversarial examples using standard techniques we need a differentiable version of interpolation. [24] introduces differentiable image sampling. Their method works for any interpolation method as long as the (sub-)gradients can be defined with respect to $x, {c'}_{i,j}$. For bilinear interpolation this becomes,

$$\frac{\partial x'_{i,j}}{\partial x_{n,m}} = \sum_{n}^{H} \sum_{m}^{W} \max(0, 1 - |{c'}_{i,j}^{x} - m|) \max(0, 1 - |{c'}_{i,j}^{y} - n|) \tag{19}$$

$$\frac{\partial x'_{i,j}}{\partial {c'}_{i,j}^{x}} = \sum_{n}^{H} \sum_{m}^{W} x_{n,m} \max(0, 1 - |{c'}_{i,j}^{y} - n|) \begin{cases} 1 & \text{if } m \geq |{c'}_{i,j}^{x} - m| \\ -1 & \text{if } m < |{c'}_{i,j}^{x} - m| \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

C SGD Algorithm

Our SGD UAP algorithm is based on standard momentum based SGD while optimizing over the objective proposed in 5, the algorithm details can be seen in Algorithm 3.

D Iterative UAP Algorithm

Moosavi-Dezfooli et al. [31] introduces an iterative UAP algorithm, the algorithm can be seen in Algorithm 4.

E Average ASR_U and ASR_R with different γ's

We provide additional metrics computed on the same set of transformations, datasets, and models as in Table 2. In Table 3, we present the Average ASR_U rather than ASR_R. The average shows us that our `RobustUAP` algorithm creates UAPs which after transformation on average are better UAPs than

Table 3: Average Universal ASR of our Robust UAP algorithms to the standard UAP [31] method.

Dataset	Transformation Set	Standard UAP	SGD	Standard UAP_RP	Robust UAP
ILSVRC 2012	$R(20)$	16.3%	71.5%	24.7%	**85.3**%
	$T(2,2)$	52.6%	95.6%	55.4%	**99.4**%
	$Sc(5), R(5), B(5, 0.01)$	44.9%	92.3%	58.5%	**96.6**%
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	13.6%	72.8%	29.0%	**83.2**%
CIFAR-10	$R(30), B(2, 0.001)$	9.9%	68.8%	22.2%	**73.4**%
	$R(2), Sh(2)$	57.1%	96.3%	61.2%	**98.5**%
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	16.2%	70.8%	32.6%	**77.4**%

all other algorithms. We observe that the average shows us that even standard UAPs aren't completely ineffective after transformation they just have a very low chance of being highly effective.

In Table 4, we present ASR_R computed at $\gamma = [0.5, 0.7]$ rather than $\gamma = 0.6$. This table shows a similar story to above, and shows that our algorithm produces better results under a variety of success thresholds.

Table 4: Robust ASR of our Robust UAP algorithms to the standard UAP [31] method with $\gamma = [0.5, 0.7]$.

Dataset	Transformation Set	Standard UAP		SGD		Standard UAP_RP		Robust UAP	
		0.5	0.7	0.5	0.7	0.5	0.7	0.5	0.7
ILSVRC 2012	$R(20)$	1.8%	0.0%	88.1%	53.9%	10.3%	2.4%	**96.3**%	**76.6**%
	$T(2,2)$	55.2%	19.1%	99.7%	89.8%	59.0%	23.5%	**99.9**%	**91.4**%
	$Sc(5), R(5), B(5, 0.01)$	40.1%	9.8%	98.4%	86.4%	66.4%	27.5%	**98.5**%	**87.7**%
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	2.1%	0.0%	82.5%	56.4%	11.9%	1.3%	**94.2**%	**73.9**%
CIFAR-10	$R(30), B(2, 0.001)$	0.9%	0.0%	81.0%	49.6%	12.1%	0.4%	**86.2**%	**51.6**%
	$R(2), Sh(2)$	66.9%	23.4%	98.6%	83.1%	68.8%	21.9%	**99.4**%	**88.7**%
	$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	4.3%	0.0%	86.6%	56.6%	18.8%	6.1%	**93.0**%	**73.5**%

F Experiment Parameters

In our experiments, we have capped all algorithms at 5 epochs or if they have achieved an ASR_R of 0.95. The UAPs are trained with the same transformation set that they are evaluated on. For algorithms running PGD internally, we have capped the number of iterations to 40.

G Additional Models

We also provide additional data on our methods evaluated on the same transformations and datasets but on different models. In this case, we use ResNet-18 [20] for CIFAR-10 and MobileNet [23] for ILSVRC 2012. Results can be seen in Table 5. We observe similar performance across models suggesting that the performance of the attacks is more directly tied to transformation set and dataset.

H Algorithm Runtimes

We compare the runtimes of the different methods on one of our most challenging $R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$ transformation set on ILSVRC-2012. The results are in Table 6. We observe that `Robust UAP` is faster than both `Standard UAP` and `Standard UAP_RP` but slower than `SGD` as it performs more computations per iteration. The extra computation enables `Robust UAP` to obtain better robustness than `SGD`.

Table 5: Robust ASR on Resnet-18 for CIFAR-10 and MobileNet for ILSVRC 2012.

DATASET	MODEL	TRANSFORMATION SET	STANDARD UAP	SGD	STANDARD UAP_RP	ROBUST UAP
ILSVRC 2012	MOBILENET	$R(20)$	8.1%	71.2%	2.6%	**85.0**%
		$T(2,2)$	40.9%	98.7%	54.3%	**99.6**%
		$Sc(5), R(5), B(5, 0.01)$	16.3%	94.5%	44.3%	**96.3**%
		$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	4.1%	75.7%	8.6%	**86.2**%
CIFAR-10	RESNET-18	$R(30), B(2, 0.001)$	0.9%	67.8%	6.4%	**74.9**%
		$R(2), Sh(2)$	49.9%	99.5%	49.1%	**99.8**%
		$R(10), T(2,2), Sh(2), Sc(2), B(2, 0.001)$	8.0%	70.8%	12.2%	**83.8**%

Table 6: Average Runtime for Robust UAP algorithms

ALGORITHM	TIME(MIN)
STANDARD UAP	26
SGD	12
STANDARD UAP_RP	42
ROBUST UAP	24