Constructing Unrestricted Adversarial Examples with Generative Models

Yang Song
Stanford University
yangsong@cs.stanford.edu

Rui Shu
Stanford University
ruishu@cs.stanford.edu

Nate Kushman
Microsoft Research
nkushman@microsoft.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Abstract

Adversarial examples are typically constructed by perturbing an existing data point within a small matrix norm, and current defense methods are focused on guarding against this type of attack. In this paper, we propose unrestricted adversarial examples, a new threat model where the attackers are not restricted to small norm-bounded perturbations. Different from perturbation-based attacks, we propose to synthesize unrestricted adversarial examples entirely from scratch using conditional generative models. Specifically, we first train an Auxiliary Classifier Generative Adversarial Network (AC-GAN) to model the class-conditional distribution over data samples. Then, conditioned on a desired class, we search over the AC-GAN latent space to find images that are likely under the generative model and are misclassified by a target classifier. We demonstrate through human evaluation that unrestricted adversarial examples generated this way are legitimate and belong to the desired class. Our empirical results on the MNIST, SVHN, and CelebA datasets show that unrestricted adversarial examples can bypass strong adversarial training and certified defense methods designed for traditional adversarial attacks.

1 Introduction

Machine learning algorithms are known to be susceptible to adversarial examples: imperceptible perturbations to samples from the dataset can mislead cutting edge classifiers [1, 2]. This has raised concerns for safety-critical AI applications because, for example, attackers could use them to mislead autonomous driving vehicles [3, 4, 5] or hijack voice controlled intelligent agents [6, 7, 8].

To mitigate the threat of adversarial examples, a large number of methods have been developed. These include augmenting training data with adversarial examples [2, 9, 10, 11], removing adversarial perturbations [12, 13, 14], and encouraging smoothness for the classifier [15]. Recently, [16, 17] proposed theoretically-certified defenses based on minimizing upper bounds of the training loss under worst-case perturbations. Although inspired by different perspectives, a shared design principle of current defense methods is to make classifiers more robust to small perturbations of their inputs.

In this paper, we introduce a more general attack mechanism where adversarial examples are constructed entirely from scratch instead of perturbing an existing data point by a small amount. In practice, an attacker might want to change an input significantly while not changing the semantics. Taking traffic signs as an example, an adversary performing perturbation-based attacks can draw graffiti [4] or place stickers [18] on an existing stop sign in order to exploit a classifier. However, the attacker might want to go beyond this and replace the original stop sign with a new one that was specifically manufactured to be adversarial. In the latter case, the new stop sign does not have

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

to be a close replica of the original one—the font could be different, the size could be smaller—as long as it is still identified as a stop sign by humans. We argue that *all inputs that fool the classifier without confusing humans can pose potential security threats*. In particular, we show that previous defense methods, including the certified ones [16, 17], are not effective against this more general attack. Ultimately, we hope that identifying and building defenses against such new vulnerabilities can shed light on the weaknesses of existing classifiers and enable progress towards more robust methods.

Generating this new kind of adversarial example, however, is challenging. It is clear that adding small noise is a valid mechanism for generating new images from a desired class—the label should not change if the perturbation is small enough. How can we generate completely new images from a given class? In this paper, we leverage recent advances in generative modeling [19, 20, 21]. Specifically, we train an Auxiliary Classifier Generative Adversarial Network (AC-GAN [20]) to model the set of legitimate images for each class. Conditioned on a desired class, we can search over the latent code of the generative model to find examples that are mis-classified by the model under attack, even when protected by the most robust defense methods available. The images that successfully fool the classifier without confusing humans (verified via Amazon Mechanical Turk [22]) are referred to as *Unrestricted Adversarial Examples*[1]. The efficacy of our attacking method is demonstrated on the MNIST [25], SVHN [26], and CelebA [27] datasets, where our attacks uniformly achieve over 84% success rates. In addition, our unrestricted adversarial examples show moderate transferability to other architectures, reducing by 35.2% the accuracy of a black-box certified classifier (i.e. a certified classifier with an architecture unknown to our method) [17].

2 Background

In this section, we review recent work on adversarial examples, defense methods, and conditional generative models. Although adversarial examples can be crafted for many domains, we focus on image classification tasks in the rest of this paper, and will use the words "examples" and "images" interchangeably.

Adversarial examples Let $x \in \mathbb{R}^m$ denote an input image to a classifier $f : \mathbb{R}^m \to \{1, 2, \cdots, k\}$, and assume the attacker has full knowledge of f (a.k.a., white-box setting). [1] discovered that it is possible to find a slightly different image $x' \in \mathbb{R}^m$ such that $\|x' - x\| \leq \epsilon$ but $f(x') \neq f(x)$, by solving a surrogate optimization problem with L-BFGS [28]. Different matrix norms $\|\cdot\|$ have been used, such as l_∞ ([9]), l_2 ([29]) or l_0 ([30]). Similar optimization-based methods are also proposed in [31, 32]. In [2], the authors observe that $f(x)$ is approximately linear and propose the Fast Gradient Sign Method (FGSM), which applies a first-order approximation of $f(x)$ to speed up the generation of adversarial examples. This procedure can be repeated several times to give a stronger attack named Projected Gradient Descent (PGD [10]).

Defense methods Existing defense methods typically try to make classifiers more robust to small perturbations of the image. There has been an "arms race" between increasingly sophisticated attack and defense methods. As indicated in [33], the strongest defenses to date are adversarial training [10] and certified defenses [16, 17]. In this paper, we focus our investigation of unrestricted adversarial attacks on these defenses.

Generative adversarial networks (GANs) A GAN [19, 20, 34, 35, 36] is composed of a generator $g_\theta(z)$ and a discriminator $d_\phi(x)$. The generator maps a source of noise $z \sim P_z$ to a synthetic image $x = g_\theta(z)$. The discriminator receives an image x and produces a value $d_\phi(x)$ to distinguish whether it is sampled from the true image distribution P_x or generated by $g_\theta(z)$. The goal of GAN training is to learn a discriminator to reliably distinguish between fake and real images, and to use this discriminator to train a good generator by trying to fool the discriminator. To stabilize training, we use the Wasserstein GAN [34] formulation with gradient penalty [21], which solves the following optimization problem

$$\min_\theta \max_\phi \mathbb{E}_{z\sim P_z}[d_\phi(g_\theta(z))] - \mathbb{E}_{x\sim P_x}[d_\phi(x)] + \lambda\, \mathbb{E}_{\tilde{x}\sim P_{\tilde{x}}}[(\|\nabla_{\tilde{x}} d_\phi(\tilde{x})\|_2 - 1)^2],$$

[1] In previous drafts we called it Generative Adversarial Examples. We switched the name to emphasize the difference from [23]. Concurrently, the same name was also used in [24] to refer to adversarial examples beyond small perturbations.

where $P_{\tilde{x}}$ is the distribution obtained by sampling uniformly along straight lines between pairs of samples from P_x and generated images from $g_\theta(z)$.

To generate adversarial examples with the intended semantic information, we also need to control the labels of the generated images. One popular method of incorporating label information into the generator is Auxiliary Classifier GAN (AC-GAN [20]), where the conditional generator $g_\theta(z, y)$ takes label y as input and an auxiliary classifier $c_\psi(x)$ is introduced to predict the labels of both training and generated images. Let $c_\psi(y \mid x)$ be the confidence of predicting label y for an input image x. The optimization objectives for the generator and the discriminator, respectively, are:

$$\min_\theta -\mathbb{E}_{z\sim P_z, y\sim P_y}[d_\phi(g_\theta(z,y)) - \log c_\psi(y \mid g_\theta(z,y)))]$$

$$\min_{\phi,\psi} \mathbb{E}_{z\sim P_z, y\sim P_y}[d_\phi(g_\theta(z,y))] - \mathbb{E}_{x\sim P_x}[d_\phi(x)] - \mathbb{E}_{x\sim P_x, y\sim P_{y|x}}[\log c_\psi(y \mid x)] + \lambda\, \mathbb{E}_{\tilde{x}\sim P_{\tilde{x}}}[(\|\nabla_{\tilde{x}} d_\phi(\tilde{x})\|_2 - 1)^2],$$

where $d_\phi(\cdot)$ is the discriminator, P_y represents a uniform distribution over all labels $\{1, 2, \cdots, k\}$, $P_{y|x}$ denotes the ground-truth distribution of y given x, and P_z is chosen to be $\mathcal{N}(0, 1)$ in our experiments.

3 Methods

3.1 Unrestricted adversarial examples

We start this section by formally characterizing perturbation-based and unrestricted adversarial examples. Let $\mathcal{I}$ be the set of all digital images under consideration. Suppose $o : \mathcal{O} \subseteq \mathcal{I} \to \{1, 2, \cdots, K\}$ is an oracle that takes an image in its domain $\mathcal{O}$ and outputs one of K labels. In addition, we consider a classifier $f : \mathcal{I} \to \{1, 2, \cdots, K\}$ that can give a prediction for any image in $\mathcal{I}$, and assume $f \neq o$. Equipped with those notations, we can provide definitions used in this paper:

Definition 1 (Perturbation-Based Adversarial Examples)**.** *Given a subset of (test) images $\mathcal{T} \subset \mathcal{O}$, small constant $\epsilon > 0$, and matrix norm $\|\cdot\|$, a perturbation-based adversarial example is defined to be any image in $\mathcal{A}_p \triangleq \{x \in \mathcal{O} \mid \exists x' \in \mathcal{T}, \|x - x'\| \leq \epsilon \wedge f(x') = o(x') = o(x) \neq f(x)\}$.*

In other words, traditional adversarial examples are based on perturbing a correctly classified image in $\mathcal{T}$ so that f gives an incorrect prediction, according to the oracle o.

Definition 2 (Unrestricted Adversarial Examples)**.** *An unrestricted adversarial example is any image that is an element of $\mathcal{A}_u \triangleq \{x \in \mathcal{O} \mid o(x) \neq f(x)\}$.*

In most previous work on perturbation-based adversarial examples, the oracle o is implicitly defined as a black box that gives ground-truth predictions (consistent with human judgments), $\mathcal{T}$ is chosen to be the test dataset, and $\|\cdot\|$ is usually one of l_∞ ([9]), l_2 ([29]) or l_0 ([30]) matrix norms. Since o corresponds to human evaluation, $\mathcal{O}$ should represent all images that look realistic to humans, including those with small perturbations. The past work assumed $o(x)$ was known by restricting x to be close to another image x' which came from a labeled dataset. Our work removes this restriction, by using a high quality generative model which can generate samples which, with high probability, humans will label with a given class. From the definition it is clear that $\mathcal{A}_p \subset \mathcal{A}_u$, which means our proposed unrestricted adversarial examples are a strict generalization of traditional perturbation-based adversarial examples, where we remove the small-norm constraints.

3.2 Practical unrestricted adversarial attacks

The key to practically producing unrestricted adversarial examples is to model the set of legitimate images $\mathcal{O}$. We do so by training a generative model $g(z, y)$ to map a random variable $z \in \mathbb{R}^m$ and a label $y \in \{1, 2, \cdots, K\}$ to a legitimate image $x = g(z, y) \in \mathcal{O}$ satisfying $o(x) = y$. If the generative model is ideal, we will have $\mathcal{O} \equiv \{g(z, y) \mid y \in \{1, 2, \cdots, K\}, z \in \mathbb{R}^m\}$. Given such a model we can in principle enumerate all unrestricted adversarial examples for a given classifier f, by finding all z and y such that $f(g(z, y)) \neq y$.

In practice, we can exploit different approximations of the ideal generative model to produce different kinds of unrestricted adversarial examples. Because of its reliable conditioning and high fidelity image generation, we choose AC-GAN [20] as our basic class-conditional generative model. In what follows, we explore two attacks derived from variants of AC-GAN (see pseudocode in Appendix B).

Basic attack Let $g_\theta(z, y), c_\phi(x)$ be the generator and auxiliary classifier of AC-GAN, and let $f(x)$ denote the classifier that we wish to attack. We focus on *targeted* unrestricted adversarial attacks, where the attacker tries to generate an image x so that $o(x) = y_{\text{source}}$ but $f(x) = y_{\text{target}}$. In order to produce unrestricted adversarial examples, we propose finding the appropriate z by minimizing a loss function $\mathcal{L}$ that is carefully designed to produce high fidelity unrestricted adversarial examples.

We decompose the loss as $\mathcal{L} = \mathcal{L}_0 + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2$, where λ_1, λ_2 are positive hyperparameters for weighting different terms. The first component

$$\mathcal{L}_0 \triangleq -\log f(y_{\text{target}} \mid g_\theta(z, y_{\text{source}})) \tag{1}$$

encourages f to predict y_{target}, where $f(y \mid x)$ denotes the confidence of predicting label y for input x. The second component

$$\mathcal{L}_1 \triangleq \frac{1}{m} \sum_{i=1}^{m} \max\{|z_i - z_i^0| - \epsilon, 0\} \tag{2}$$

soft-constrains the search region of z so that it is close to a randomly sampled noise vector z^0. Here $z = (z_1, z_2, \cdots, z_m)$, $\{z_1^0, z_2^0, \cdots, z_m^0\} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$, and ϵ is a small positive constant. For a good generative model, we expect that $x^0 = g_\theta(z^0, y_{\text{source}})$ is diverse for randomly sampled z^0 and $o(x^0) = y_{\text{source}}$ holds with high probability. Therefore, by reducing the distance between z and z^0, $\mathcal{L}_1$ has the effect of generating more diverse adversarial examples from class y_{source}. Without this constraint, the optimization may always converge to the same example for each class. Finally

$$\mathcal{L}_2 \triangleq -\log c_\phi(y_{\text{source}} \mid g_\theta(z, y_{\text{source}})) \tag{3}$$

encourages the auxiliary classifier c_ϕ to give correct predictions, and $c_\phi(y \mid x)$ is the confidence of predicting y for x. We hypothesize that c_ϕ is relatively uncorrelated with f, which can possibly promote the generated images to reside in class y_{source}.

Note that $\mathcal{L}$ can be easily modified to perform *untargeted* attacks, for example replacing $\mathcal{L}_0$ with $-\max_{y \neq y_{\text{source}}} \log f(y \mid g_\theta(z, y_{\text{source}}))$. Additionally, when performing our evaluations, we need to use humans to ensure that our generative model is actually generating images which are in one of the desired classes with high probability. In contrast, when simply perturbing an existing image, past work has been able to assume that the true label does not change if the perturbation is small. Thus, during evaluation, to test whether the images are legitimate and belong to class y_{source}, we use crowd-sourcing on Amazon Mechanical Turk (MTurk).

Noise-augmented attack The representation power of the AC-GAN generator can be improved if we add small trainable noise to the generated image. Let ϵ_{attack} be the maximum magnitude of noise that we want to apply. The noise-augmented generator is defined as $g_\theta(z, \tau, y; \epsilon_{\text{attack}}) \triangleq g_\theta(z, y) + \epsilon_{\text{attack}} \tanh(\tau)$, where τ is an auxiliary trainable variable with the same shape as $g_\theta(z, y)$ and $\tanh$ is applied element-wise. As long as ϵ_{attack} is small, $g_\theta(z, \tau, y; \epsilon_{\text{attack}})$ should be indistinguishable from $g_\theta(z, y)$, and $o(g_\theta(z, \tau, y; \epsilon_{\text{attack}})) = o(g_\theta(z, y))$, i.e., adding small noise should preserve image quality and ground-truth labels. Similar to the basic attack, noise-augmented unrestricted adversarial examples can be obtained by solving $\min_{z,\tau} \mathcal{L}$, with $g_\theta(z, y_{\text{source}})$ in (1) and (3) replaced by $g_\theta(z, \tau, y_{\text{source}}; \epsilon_{\text{attack}})$.

One interesting observation is that traditional perturbation-based adversarial examples can also be obtained as a special case of our noise-augmented attack, by choosing a suitable $g_\theta(z, y)$ instead of the AC-GAN generator. Specifically, let $\mathcal{T}$ be the test dataset, and $\mathcal{T}_y = \{x \in \mathcal{T} \mid o(x) = y\}$. We can use a discrete latent code $z \in \{1, 2, \cdots, |\mathcal{T}_{y_{\text{source}}}|\}$ and specify $g_\theta(z, y)$ to be the z-th image in $\mathcal{T}_y$. Then, when z^0 is uniformly drawn from $\{1, 2, \cdots, |\mathcal{T}_{y_{\text{source}}}|\}$, $\lambda_1 \to \infty$ and $\lambda_2 = 0$, we will recover an objective similar to FGSM [2] or PGD [10].

4 Experiments

4.1 Experimental details

Amazon Mechanical Turk settings In order to demonstrate the success of our unrestricted adversarial examples, we need to verify that their ground-truth labels disagree with the classifier's

Table 1: Attacking certified defenses on MNIST. The unrestricted adversarial examples here are untargeted and without noise-augmentation. Numbers represent success rates (%) of our attack, based on human evaluations on MTurk. While no perturbation-based attack with $\epsilon = 0.1$ can have a success rate larger than the *certified rate* (when evaluated on the training set) we are able to achieve that by considering a more general attack mechanism.

Classifier \ Source	0	1	2	3	4	5	6	7	8	9	Overall	Certified Rate ($\epsilon = 0.1$)
Raghunathan et al. [16]	90.8	48.3	86.7	93.7	94.7	85.7	93.4	80.8	96.8	95.0	86.6	≤ 35.0
Kolter & Wang [17]	94.2	57.3	92.2	94.0	93.7	89.6	95.7	81.4	96.3	93.5	88.8	≤ 5.8

predictions. To this end, we use Amazon Mechanical Turk (MTurk) to manually label each unrestricted adversarial example.

To improve signal-to-noise ratio, we assign the same image to 5 different workers and use the result of a majority vote as ground-truth. For each worker, the MTurk interface contains 10 images and for each image, we use a button group to show all possible labels. The worker is asked to select the correct label for each image by clicking on the corresponding button. In addition, each button group contains an "N/A" button that the workers are instructed to click on if they think the image is not legitimate or does not belong to any class. To confirm our MTurk setup results in accurate labels, we ran a test to label MNIST images. The results show that **99.6%** of majority votes obtained from workers match the ground-truth labels.

For some of our experiments, we want to investigate whether unrestricted adversarial examples are more similar to existing images in the dataset, compared to perturbation-based attacks. We use the classical A/B test for this, *i.e.*, each synthesized adversarial example is randomly paired with an existing image from the dataset, and the annotators are asked to identify the synthesized images. Screen shots of all our MTurk interfaces can be found in the Appendix E.

Datasets The datasets used in our experiments are MNIST [25], SVHN [26], and CelebA [27]. Both MNIST and SVHN are images of digits. For CelebA, we group the face images according to female/male, and focus on gender classification. We test our attack on these datasets because the tasks (digit categorization and gender classification) are easier and less ambiguous for MTurk workers, compared to those having more complicated labels, such as Fashion-MNIST [37] or CIFAR-10 [38].

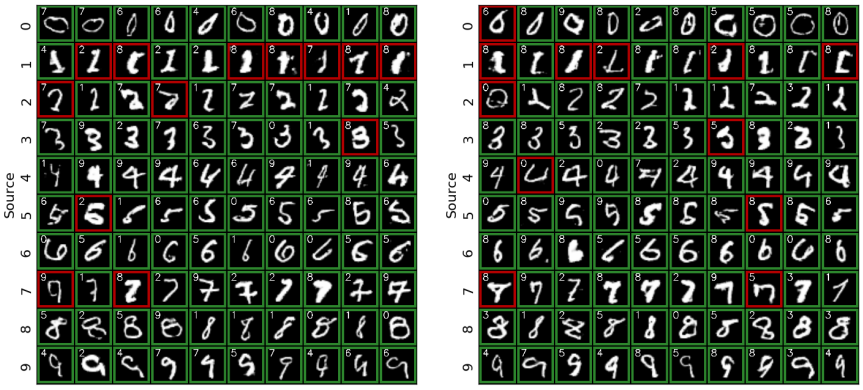
Model Settings We train our AC-GAN [20] with gradient penalty [21] on all available data partitions of each dataset, including training, test, and extra images (SVHN only). This is based on the assumption that attackers can access a large number of images. We use ResNet [39] blocks in our generative models, mainly following the architecture design of [21]. For training classifiers, we only use the training partition of each dataset. We copy the network architecture from [10] for the MNIST task, and use a similar ResNet architecture to [13] for other datasets. For more details about architectures, hyperparameters and adversarial training methods, please refer to Appendix C.

4.2 Untargeted attacks against certified defenses

We first show that our new adversarial attack can bypass the recently proposed *certified defenses* [16, 17]. These defenses can provide a theoretically verified certificate that a training example cannot be classified incorrectly by any perterbation-based attack with a perturbation size less than a given ϵ.

Setup For each source class, we use our method to produce 1000 untargeted unrestricted adversarial examples without noise-augmentation. By design these are all incorrectly classified by the target classifer. Since they are synthesized from scratch, we report in Tab. 1 the fraction labeled by human annotators as belonging to the intended source class. We conduct these experiments on the MNIST dataset to ensure a fair comparison, since certified classifiers with pre-trained weights for this dataset can be obtained directly from the authors of [16, 17]. We produce untargeted unrestricted adversarial examples in this task, as the certificates are for untargeted attacks.

Results Tab. 1 shows the results. We can see that the stronger of the two certified defenses, [17], provides a certificate for 94.2% of the samples in the MNIST test set with $\epsilon = 0.1$ (out of 1). Since our technique is not perturbation-based, 88.8% of our samples are able to fool this defense. Note this does not mean the original defense is broken, since we are considering a different threat model.



(a) Untargeted attacks against [16]. (b) Untargeted attacks against [17].

Figure 1: Random samples of untargeted unrestricted adversarial examples (w/o noise) against certified defenses on MNIST. Green and red borders indicate success/failure respectively, according to MTurk results. The annotation in upper-left corner of each image represents the classifier's prediction. For example, the entry in the top left corner (left panel) is classified as 0 by MTurk, and as a 7 by the classifier, and is therefore considered a success. The entry in the top left corner (right panel) is not classified as 0 by MTurk, and is therefore counted as a failure.

Figure 2: Comparing PGD attacks with high $\epsilon = 0.31$ (top) and ours (bottom) for [16] on MNIST. The two methods have comparable success rates of 86.0% (with $\epsilon = 0.31$) and 86.6% respectively. Our images, however, look significantly more realistic.

4.2.1 Evading human detection

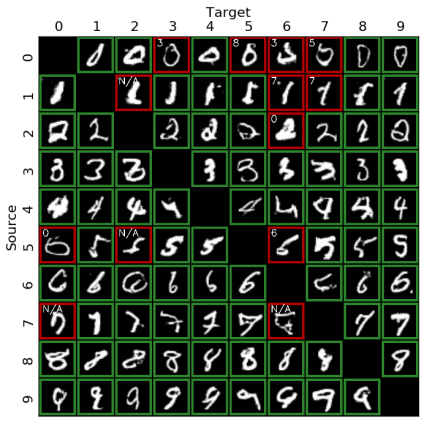
One natural question is, *why not increase ϵ to achieve higher success rates?* With a large enough ϵ, existing perturbation-based attacks will be able to fool certified defenses *using larger perturbations*. However, we can see the downside of this approach in Fig. 2: because perturbations are large, the resulting samples appear obviously altered.

Setup To show this quantitatively, we increase the ϵ value for a traditional perturbation-based attack (a 100-step PGD) until the attack success rates on the certified defenses are similar to those for our technique. Specifically, we used an ϵ of 0.31 and 0.2 to attack [16] and [17] respectively, resulting in success rates of 86.0% and 83.5%. We then asked human annotators to distinguish between the adversarial examples and unmodified images from the dataset, in an A/B test setting. If the two are indistinguishable we expect a 50% success rate.

Results We found that with perturbation-based examples [16], annotators can correctly identify adversarial images with a 92.9% success rate. In contrast, they can only correctly identify adversarial examples from our attack with a 76.8% success rate. Against [17], the success rates are 87.6% and 78.2% respectively. We expect this gap to increase even more as better generative models and defense mechanisms are developed.

4.3 Targeted attacks against adversarial training

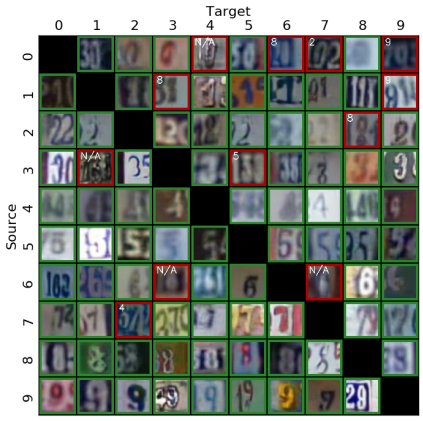
The theoretical guarantees provided by certified defenses are satisfying, however, these defenses require computationally expensive optimization techniques and do not yet scale to larger datasets. Furthermore, more traditional defense techniques actually perform better in practice in certain cases. [33] has shown that the strongest non-certified defense currently available is the adversarial training technque presented in [10], so we also evaluated our attack against this defense technique. We perform this evaluation in the targeted setting in order to better understand how the success rate varies between various source-target pairs.



(a) sampled examples on MNIST

Source \ Target	0	1	2	3	4	5	6	7	8	9
0		92	94	83	94	91	95	89	91	100
1	74		66	79	78	66	57	61	70	77
2	77	81		83	83	59	82	68	86	76
3	88	96	92		89	96	85	97	92	96
4	93	96	82	80		86	93	95	92	94
5	80	79	78	93	89		79	81	90	88
6	90	96	82	86	90	96		72	91	79
7	55	85	83	93	83	69	31		91	87
8	96	96	93	96	97	94	96	93		96
9	79	90	84	96	99	89	69	99	94	

(b) success rates on MNIST



(c) sampled examples on SVHN

Source \ Target	0	1	2	3	4	5	6	7	8	9
0		79	87	81	79	88	85	81	70	87
1	61		75	70	65	57	70	83	57	60
2	90	91		88	92	92	84	93	82	88
3	89	91	93		88	90	97	86	85	88
4	83	87	83	73		82	72	80	68	88
5	90	97	98	94	92		92	98	96	94
6	76	73	70	77	82	85		65	75	66
7	87	92	83	88	83	90	90		84	80
8	91	83	94	86	91	89	89	89		94
9	90	91	96	93	87	92	86	91	89	

(d) success rates on SVHN

Figure 3: (a)(c) Random samples of targeted unrestricted adversarial examples (w/o noise). Row i and column j is an image that is supposed to be class i and classifier predicts it to be class j. Green border indicates that the image is voted legitimate by MTurk workers, and red border means the label given by workers (as shown in the upper left corner) disagrees with the image's source class. (b)(d) The success rates (%) of our targeted unrestricted adversarial attack (w/o noise). Also see Tab. 2.

Table 2: Overall success rates of our targeted unrestricted adversarial attacks. Success rates of PGD are provided to show that the network is adversarially-trained to be robust. †Best public result.

Robust Classifier	Accuracy (orig. images)	Success Rate of PDG	Our Success Rate (w/o Noise)	Our Success Rate (w/ Noise)	ϵ_{attack}
Madry network [10] on MNIST	98.4	10.4†	85.2	85.0	0.3
ResNet (adv-trained) on SVHN	96.3	59.9	84.2	91.6	0.03
ResNet (adv-trained) on CelebA	97.3	20.5	91.1	86.7	0.03

Setup We produce 100 unrestricted adversarial examples for each pair of source and target classes and ask human annotators to label them. We also compare to traditional PGD attacks as a reference. For the perturbation-based attacks against the ResNet networks, we use a 20-step PGD with values of ϵ given in the table. For the perturbation-based attack against the Madry network [10], we report the best published result [40]. It's important to note that the reference perturbation-based results are not directly comparable to ours because they are i) untargeted attacks and ii) limited to small perturbations. Nonetheless, they can provide a good sense of the robustness of adversarially-trained networks.

Results A summary of the results can be seen in Tab. 2. We can see that the defense from [10] is quite effective against the basic perturbation-based attack, limiting the success rate to 10.4% on

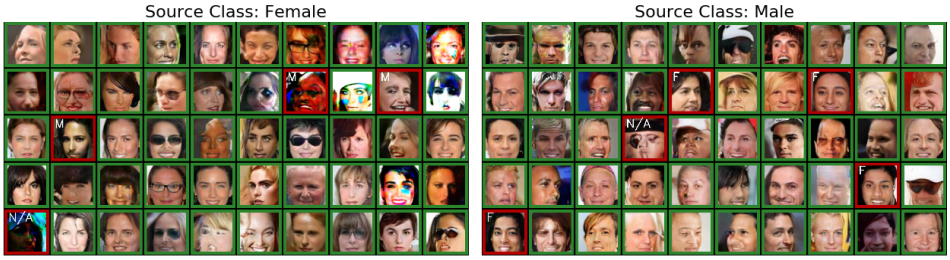


Figure 4: Sampled unrestricted adversarial examples (w/o noise) for fooling the classifier to misclassify a female as male (left) and the other way around (right). Green, red borders and annotations have the same meanings as in Fig. 3, except "F" is short for "Female" and "M" is short for "Male".

Table 3: Transferability of unrestricted adversarial examples on MNIST. We attack Madry Net [10] (adv) with our method and feed legitimate unrestricted adversarial examples, as verified by AMT workers, to other classifiers. Here "adv" is short for "adversarially-trained" (with PGD) and "no adv" means no adversarial training is used. Numbers represent accuracies of classifiers.

Attack Type \ Classifier	Madry Net [10] (no adv)	Madry Net [10] (adv)	ResNet (no adv)	ResNet (adv)	[16]	[17]
No attack	99.5	98.4	99.3	99.4	95.8	98.2
Our attack (w/o noise)	95.1	0	92.7	93.7	77.1	84.3
Our attack (w/ noise, $\epsilon_{\text{attack}} = 0.3$)	78.3	0	73.8	84.9	78.1	63.0

MNIST, and 20.5% on CelebA. In contrast, our unrestricted adversarial examples (with or without noise-augmentation) can successfully fool this defense with more than an 84% success rate on all datasets. We find that adding noise-augmentation to our attack does not significantly change the results, boosting the SVNH success rate by 7.4% while reducing the CelebA success rate by 4.4%. In Fig. 3 and Fig. 4, we show samples and detailed success rates of unrestricted adversarial attacks without noise-augmentation. More samples and success rate details are provided in Appendix D.

4.4 Transferability

An important feature of traditional perturbation-based attacks is their transferability across different classifiers.

Setup To test how our unrestricted adversarial examples transfer to other architectures, we use all of the unrestricted adversarial examples we created to target Madry Network [41] on MNIST for the results in Section 4.3, and filter out invalid ones using the majority vote of a set of human annotators. We then feed these unrestricted adversarial examples to other architectures. Besides the adversarially-trained Madry Network, the architectures we consider include a ResNet [39] similar to those used on SVHN and CelebA datasets in Section 4.3. We test both normally-trained and adversarially-trained ResNets. We also take the architecture of Madry Network in [10] and train it without adversarial training.

Results We show in Tab. 3 that unrestricted adversarial examples exhibit moderate transferability to different classifiers, which means they can be threatening in a black-box scenario as well. For attacks without noise-augmentation, the most successful transfer happens against [16], where the success rate is 22.9%. For the noise-augmented attack, the most successful transfer is against [17], with a success rate of 37.0%. The results indicate that the transferability of unrestricted adversarial examples can be generally enhanced with noise-augmentation.

5 Analysis

In this section, we analyze why our method can attack a classifier using a generative model, under some idealized assumptions. For simplicity, we assume the target $f(x)$ is a binary classifier, where $x \in \mathbb{R}^n$ is the input vector. Previous explanations for perturbation-based attacks [2] assume that the score function $s(x) \in \mathbb{R}$ used by f is almost linear. Suppose $s(x) \approx w^\intercal x + b$ and $w, x \in \mathbb{R}^n$ both have high dimensions (large n). We can choose the perturbation to be $\delta = \epsilon\,\text{sign}(w)$, so that

$s(x+\delta) - s(x) \approx \epsilon \cdot n$. Though ϵ is small, n is typically a large number, therefore $\epsilon \cdot n$ can be large enough to change the prediction of $f(x)$. Similarly, we can explain the existence of unrestricted adversarial examples. Suppose $g(z, y) \in \mathbb{R}^n$ is an ideal generative model that can always produce legitimate images of class $y \in \{0, 1\}$ for any $z \in \mathbb{R}^m$, and assume for all z^0, $f(g(z^0, y)) = y$. The end-to-end score function $s(g(z, y))$ can be similarly approximated by $s(g(z, y)) \approx w_g^\intercal z + b_g$, and we can again take $\delta_z = \epsilon \operatorname{sign}(w_g)$, so that $s(g(z^0 + \delta_z, y)) - s(g(z^0, y)) \approx \epsilon \cdot m$. Because $m \gg 1$, $\epsilon \cdot m$ can be large enough to change the prediction of f, justifying why we can find many unrestricted adversarial examples by minimizing $\mathcal{L}$.

It becomes harder to analyze the case of an imperfect generative model. We provide a theoretical analysis in Appendix A under relatively strong assumptions to argue that most unrestricted adversarial examples produced by our method should be legitimate.

6 Related work

Some recent attacks also use more structured perturbations beyond simple norm bounds. For example, [42] shows that wearing eyeglass frames can cause face-recognition models to misclassify. [43] tests the robustness of classifiers to "nuisance variables", such as geometric distortions, occlusions, and illumination changes. [44] proposes converting the color space from RGB to HSV and shifting H, S components. [45] proposes mapping the input image to a latent space using GANs, and search for adversarial examples in the vicinity of the latent code. In contrast to our unrestricted adversarial examples where images are synthesized from scratch, these attacking methods craft malicious inputs based on a given test dataset using a limited set of image manipulations. Similar to what we have shown for traditional adversarial examples, we can view these attacking methods as special instances of our unrestricted adversarial attack framework by choosing a suitable generative model.

There is also a related class of maliciously crafted inputs named fooling images [46]. Different from adversarial examples, fooling images consist of noise or patterns that do not necessarily look realistic but are nonetheless predicted to be in one of the known classes with high confidence. As with our unrestricted adversarial examples, fooling images are not restricted to small norm-bounded perturbations. However, fooling images do not typically look legitimate to humans, whereas our focus is on generating adversarial examples which look realistic and meaningful.

Generative adversarial networks have also been used in some previous attack and defense mechanisms. Examples include AdvGAN [23], DeepDGA [47], ATN [48], GAT [49] and Defense-GAN [14]. The closest to our work are AdvGAN and DeepDGA. AdvGAN also proposes to use GANs for creating adversarial examples. However, their adversarial examples are still based on *small norm-bounded perturbations*. This enables them to assume adversarial examples have the same ground-truth labels as unperturbed images, while we use *human evaluation* to ensure the labels for our evaluation. DeepDGA uses GANs to generate adversarial domain names. However, domain names are arguably easier to generate than images since they need to satisfy fewer constraints.

7 Conclusion

In this paper, we explore a new threat model and propose a more general form of adversarial attacks. Instead of perturbing existing data points, our unrestricted adversarial examples are synthesized entirely from scratch, using conditional generative models. As shown in experiments, this new kind of adversarial examples undermines current defenses, which are designed for perturbation-based attacks. Moreover, unrestricted adversarial examples are able to transfer to other classifiers trained using the same dataset. After releasing the first draft of this paper, there has been a surge of interest in more general adversarial examples. For example, a contest [24] has recently been launched on unrestricted adversarial examples.

Both traditional perturbation-based attacks and the new method proposed in this paper exploit current classifiers' vulnerability to covariate shift [50]. The prevalent training framework in machine learning, Empirical Risk Minimization [51], does not guarantee performance when tested on a different data distribution. Therefore, it is important to develop new training methods that can generalize to different input distributions, or new methods that can reliably detect covariate shift [52]. Such new methods should be able to alleviate threats of both perturbation-based and unrestricted adversarial examples.

Acknowledgements

The authors would like to thank Shengjia Zhao for reviewing an early draft of this paper. We also thank Ian Goodfellow, Ben Poole, Anish Athalye and Sumanth Dathathri for helpful online discussions. This research was supported by Intel Corporation, TRI, NSF (#1651565, #1522054, #1733686) and FLI (#2017-158687).

References

[1] 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.

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

[3] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[4] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *International Conference on Computer Vision. IEEE*, 2017.

[6] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530, 2016.

[7] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117. ACM, 2017.

[8] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[10] 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.

[11] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

[12] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[13] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

[14] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. 2018.

[15] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863, 2017.

[16] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

[17] J Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

[18] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[20] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pages 2642–2651, 2017.

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[22] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[23] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[24] Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

[25] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[28] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[29] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

[30] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[31] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[32] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.

[33] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[34] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[35] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *AAAI Conference on Artificial Intelligence*, 2018.

[36] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. 2018.

[37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[38] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Mnist adversarial examples challenge, 2017.

[41] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[42] 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. ACM, 2016.

[43] Alhussein Fawzi and Pascal Frossard. Measuring the effect of nuisance variables on classifiers. In *British Machine Vision Conference (BMVC)*, number EPFL-CONF-220613, 2016.

[44] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. *arXiv preprint arXiv:1804.00499*, 2018.

[45] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018.

[46] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[47] Hyrum S Anderson, Jonathan Woodbridge, and Bobby Filar. Deepdga: Adversarially-tuned domain generation and detection. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, pages 13–21. ACM, 2016.

[48] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

[49] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

[50] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[51] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[52] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

[53] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

[54] Phillippe Rigollet. High-dimensional statistics. *Lecture notes for course 18S997*, 2015.

A Analysis of imperfect generators

In this section, we give one possible explanation for why in practice generative models can create adversarial examples that fool classifiers. We will now assume that generators not always generate legitimate images from the desired class. We will argue that, under some strong assumptions, when the classifier's prediction contradicts the generator's label conditioning, it is more likely for the classifier to make a mistake, rather than the generator generates an incorrect image.

In order to make our argument, we first need Proposition 1:

Proposition 1. *Let $\mathbf{W} \in \mathbb{R}^{n\times m}$ be a random matrix. Assume entries of $\mathbf{W}$ are mutually independent and bounded,* i.e.*, $|w_{ij}| \leq K$ for all i and j. Then, with probability $1-\delta$, the following bound holds*

$$\frac{1}{n}\max_{\|\Delta\mathbf{x}\|_\infty \leq \epsilon} \|\mathbf{W}\cdot\Delta\mathbf{x}\|_1 \leq 4\epsilon K\sqrt{\frac{m\big(m\log 2+\log\frac{1}{\delta}\big)}{n}} + \epsilon K m. \tag{4}$$

Intuitively, Proposition 1 characterizes the robustness of a "typical" linear function as a function of its input and output dimensions. When the input dimension m is fixed, the average maximum perturbation of output $\frac{1}{n}\max_{\|\Delta\mathbf{x}\|_\infty\leq\epsilon}\|\mathbf{W}\cdot\Delta\mathbf{x}\|$ is upper bounded by $4\epsilon K\sqrt{\frac{m\left(m\log 2+\log\frac{1}{\delta}\right)}{n}} + \epsilon K m$, which decreases as n gets greater. Similarly, when the output dimension n is fixed, the average maximum perturbation is upper bounded by $O(m)$, which decreases as m gets smaller. As long as the entries of the weight matrix are mutually independent and bounded, this relationship between robustness and dimensions persists.

Although not rigorously proven, we believe that a similar relationship holds for non-linear functions as well. Consider a non-linear function $\mathbf{f}(\mathbf{x})$, where $\mathbf{x}\in\mathbb{R}^m$ and $\mathbf{f}(\mathbf{x})\in\mathbb{R}^n$. For each data sample $\mathbf{x}_i$, we can linearize $\mathbf{f}(\mathbf{x})$ at $\mathbf{x}_i$ to get $\mathbf{f}(\mathbf{x}) \approx \mathbf{J}_{\mathbf{x}_i}(\mathbf{x}-\mathbf{x}_i) = \mathbf{J}_{\mathbf{x}_i}\Delta\mathbf{x}$. We shall assume that the induced random matrix $\mathbf{J}_{\mathbf{x}}\in\mathbb{R}^{n\times m}, \mathbf{x}\sim P_x$ has bounded and mutually independent entries, and can therefore apply Proposition 1 to get the same robustness-dimension relationship.

In unrestricted adversarial attacks, we have a conditional generative model $\mathbf{g}(\mathbf{z}, y)$ that takes as input a random noise vector $\mathbf{z}\in\mathbb{R}^l$ and a label y and generates an image from $\mathbb{R}^m$. The target classifier $\mathbf{f}(\mathbf{x})$ takes an image $\mathbf{x}\in\mathbb{R}^m$ and output scores from $\mathbb{R}^n$ that are subsequently used for classification. In practice, we usually have $l \ll m$ and $m \gg n$. From our discussion above, the generative model should be asymptotically more robust than the classifier. Therefore, when perturbing $\mathbf{z}$ such that $\mathbf{f}(\mathbf{g}(\mathbf{z}', y)), \|\mathbf{z}'-\mathbf{z}\|_\infty \leq \epsilon$ predicts an incorrect label (not y), it is more likely that $\mathbf{f}$ makes the mistake, rather than $\mathbf{g}$ not generating an image with label y. That could explain why we can construct legitimate unrestricted adversarial examples.

Proof of Proposition 1

Proof. Let $\mathbf{W} = (\mathbf{w}_1^\intercal, \mathbf{w}_2^\intercal, \cdots, \mathbf{w}_n^\intercal)^\intercal$, where $\mathbf{w}_i$ represents the i-th row vector of $\mathbf{W}$. In order to bound $\max_{\|\Delta\mathbf{x}\|_\infty\leq\epsilon}\|\mathbf{W}\Delta\mathbf{x}\|_1$, we first bound $\|\mathbf{W}\Delta\mathbf{x}\|_1$ for any fixed vector $\Delta\mathbf{x}$ from the ball $\mathcal{B}_\epsilon := \{\mathbf{x} \mid \|\mathbf{x}\|_\infty \leq \epsilon\}$ and then apply union bound. Note that $\|\mathbf{W}\Delta\mathbf{x}\|_1$ can be written as the sum of n terms, *i.e.*, $\|\mathbf{W}\Delta\mathbf{x}\|_1 = \sum_{i=1}^n |\mathbf{w}_i^\intercal\Delta\mathbf{x}|$, where each term $\mathbf{w}_i^\intercal\Delta\mathbf{x}$ can be bounded using McDiarmid's inequality [53]

$$\mathbb{P}\left(\sum_{j=1}^m w_{ij}(\Delta x)_j - M_i \geq \lambda\right) \leq \exp\left(\frac{-\lambda^2}{2mK^2\epsilon^2}\right), \tag{5}$$

where $M_i = \sum_{j=1}^m \mathbb{E}[w_{ij}(\Delta x)_j] \leq \epsilon K m$, according to the assumption.

From (5) we conclude the random variable $\mathbf{w}_i^\intercal\Delta\mathbf{x} - M_i$ is sub-Gaussian [54], hence

$$\begin{aligned}
&\mathbb{E}\left[\exp(s|\mathbf{w}_i^\intercal\Delta\mathbf{x} - M_i|)\right] \leq \exp\left(4s^2mK^2\epsilon^2\right)\\
\Rightarrow &\mathbb{E}\left[\exp\left(s\sum_{i=1}^n|\mathbf{w}_i^\intercal\Delta\mathbf{x} - M_i|\right)\right] \leq \exp(4s^2nmK^2\epsilon^2).
\end{aligned}$$

With Markov inequality [53], we obtain

$$\mathbb{P}\left(\sum_{i=1}^{n}|\mathbf{w}_i^\intercal\Delta\mathbf{x}-M_i|\geq\lambda\right)\leq\exp(4s^2nmK^2\epsilon^2-s\lambda). \tag{6}$$

Because (6) holds for every $s\in\mathbb{R}$, we can optimize s to get the tightest bound

$$\begin{aligned}
&\mathbb{P}\left(\sum_{i=1}^{n}|\mathbf{w}_i^\intercal\Delta\mathbf{x}-M_i|\geq\lambda\right)\leq\exp\left(-\frac{\lambda^2}{16nmK^2\epsilon^2}\right)\\
\Rightarrow&\mathbb{P}\left(\|\mathbf{W}\Delta\mathbf{x}\|_1\geq\lambda+\sum_{i=1}^{n}|M_i|\right)\leq\exp\left(-\frac{\lambda^2}{16nmK^2\epsilon^2}\right)\\
\Rightarrow&\mathbb{P}\left(\|\mathbf{W}\Delta\mathbf{x}\|_1\geq\lambda+K\epsilon nm\right)\leq\exp\left(-\frac{\lambda^2}{16nmK^2\epsilon^2}\right).
\end{aligned}$$

Now we are ready to apply union bound to control $\max_{\Delta\mathbf{x}\in\mathcal{B}_\epsilon}\|\mathbf{W}\Delta\mathbf{x}\|_1$. Although $\mathcal{B}_\epsilon$ is an infinite set, we only need to consider a finite set of vertices $\mathcal{V}_\epsilon:=\{\mathbf{x}\mid x_i\in\{-\epsilon,\epsilon\},\ \forall i=1,2,\cdots,m\}$. To see this, assume $\mathbf{x}^*=\arg\max_{\mathbf{x}\in\mathcal{S}_\epsilon}\|\mathbf{W}\mathbf{x}\|_1$ but $\mathbf{x}^*\notin\mathcal{V}_\epsilon$. Because $\mathcal{B}_\epsilon$ is a convex polytope with vertices $\mathcal{V}_\epsilon$, we have $\mathbf{x}^*=\sum_{i=1}^{2^m}\lambda_i\mathbf{v}_i$, where $\lambda_i\in[0,1]$, $\sum_{i=1}^{2^m}\lambda_i=1$ and $\mathbf{v}_i$ denotes the i-th vertex in $\mathcal{V}_\epsilon$. By triangle inequality we have

$$\begin{aligned}
\|\mathbf{W}\mathbf{x}^*\|_1&=\left\|\sum_{i=1}^{2^m}\lambda_i\mathbf{W}\mathbf{v}_i\right\|_1\\
&\leq\sum_{i=1}^{2^m}\lambda_i\|\mathbf{W}\mathbf{v}_i\|_1\\
&\leq\left(\sum_{i=1}^{2^m}\lambda_i\right)\max_{i\in[0,2^m]}\|\mathbf{W}\mathbf{v}_i\|_1\\
&=\max_{i\in[0,2^m]}\|\mathbf{W}\mathbf{v}_i\|_1\leq\|\mathbf{W}\mathbf{x}^*\|_1.
\end{aligned}$$

Let $\mathbf{v}^*:=\arg\max_{\mathbf{v}_i,i\in[0,2^m]}\|\mathbf{W}\mathbf{v}_i\|_1$. From the above derivation we conclude $\|\mathbf{W}\mathbf{v}^*\|_1=\|\mathbf{W}\mathbf{x}^*\|_1$ and therefore it is sufficient to only consider $\mathcal{V}_\epsilon$ for union bound:

$$\begin{aligned}
\mathbb{P}\left(\max_{\Delta\mathbf{x}\in\mathcal{S}_\epsilon}\|\mathbf{W}\Delta\mathbf{x}\|_1\geq\lambda+\epsilon Knm\right)&=\mathbb{P}\left(\max_{\Delta\mathbf{x}\in\mathcal{V}_\epsilon}\|\mathbf{W}\Delta\mathbf{x}\|_1\geq\lambda+\epsilon Knm\right)\\
&\leq|\mathcal{V}_\epsilon|\mathbb{P}\left(\|\mathbf{W}\Delta\mathbf{x}\|_1\geq\lambda+\epsilon Knm\right)\qquad\text{(Union Bound)}\\
&\leq 2^m\exp\left(-\frac{\lambda^2}{16nmK^2\epsilon^2}\right).
\end{aligned}$$

In other words, with probability $1-\delta$,

$$\max_{\Delta\mathbf{x}\in\mathcal{S}_\epsilon}\|\mathbf{W}\Delta\mathbf{x}\|_1\leq 4K\epsilon\sqrt{mn\left(m\log 2+\log\frac{1}{\delta}\right)}+\epsilon Knm,$$

and the statement of our theorem gets proved. □

B Pseudocode

C Detailed experimental settings

Datasets The datasets used in our experiments are MNIST [25], SVHN [26], and CelebA [27]. Both MNIST and SVHN are images of digits. MNIST contains 60000 28-by-28 gray-scale digits in the training set, and 10000 digits in the test set. In SVHN, there are 73257 32-by-32 images of house numbers (captured from Google Street View) for training, 26032 images for testing, and 531131 additional images as extra training data. For CelebA, there are 202599 celebrity faces, each of which has 40 binary attribute annotations. We group the face images according to female/male, and focus on gender classification. The first 150000 images are split for training and the rest are used for testing.

Algorithm 1 Unrestricted Adversarial Example Targeted Attack

procedure ATTACK($y_{\text{target}}, y_{\text{source}}, f, \theta, \phi, \epsilon, \epsilon_{\text{attack}}, \lambda_1, \lambda_2, \alpha, T$)
  Define $\mathcal{L}(z, \tau\,; z^0, y_{\text{target}}, y_{\text{source}}, f, \theta, \phi, \epsilon, \epsilon_{\text{attack}}, \lambda_1, \lambda_2)$

$$\mathcal{L} = -\log f(y_{\text{target}} \mid g_\theta(z, y_{\text{source}}) + \epsilon_{\text{attack}} \tanh(\tau)) + \lambda_1 \cdot \frac{1}{m} \sum_{i=1}^{m} \max\{|z_i - z_i| - \epsilon, 0\}$$
$$- \lambda_2 \cdot \log c_\phi(y_{\text{source}} \mid g_\theta(z, y_{\text{source}}) + \epsilon_{\text{attack}} \tanh(\tau))$$

  Initialize $x_{\text{attack}} \leftarrow \varnothing$
  while $x_{\text{attack}} = \varnothing$ **do**
    Sample $\tau \sim \mathcal{N}(0, 1)$
    Sample $z^0 \sim \mathcal{N}(0, 1)$
    Initialize $z \leftarrow z^0$
    for $i = 1 \ldots T$ **do**
      Update $z \leftarrow z - \alpha \cdot \frac{\partial \mathcal{L}(z,\tau)}{\partial z}$ ▷ α is the learning rate.
      $\Delta \leftarrow \frac{\partial \mathcal{L}(z,\tau)}{\partial \tau}$
      Update $\tau \leftarrow \tau - \alpha \cdot \frac{\Delta}{\|\Delta\|}$ ▷ We found gradient normalization to be effective
    end for
    $x \leftarrow g_\theta(z, y_{\text{source}}) + \epsilon_{\text{attack}} \tanh(\tau)$
    if $y_{\text{target}} = \arg\max_y p(f(x) = y)$ **then**
      $x \leftarrow x_{\text{attack}}$
    end if
  end while
  return x_{attack}
end procedure

Adversarial training Regarding adversarial training, we directly use the weights provided by [10] for MNIST. For other tasks, we combine the techniques from [10] and [9]. More specifically, suppose pixel space is $[0, 255]$. We first sample ϵ from $\mathcal{N}(0, 8)$, take the absolute value and truncate it to $[0, 16]$, after which we use PGD with ϵ and iteration number $\lfloor \min(\epsilon + 4, 1.25\epsilon) \rfloor$ to generate adversarial examples for adversarial training. As suggested in [9], this has the benefit of making models robust to attacks with different ϵ.

Model architectures For the AC-GAN architecture, we mostly follow the best designs tested in [21]. Specifically, we adapted their AC-GAN architecture on CIFAR-10 for our experiments of MNIST and SVHN, and used their AC-GAN architecture on 64×64 LSUN for our CelebA experiments. Since MNIST digits have lower resolution than CIFAR-10 images, we reduced one residual block in the generator so that the output shape is smaller, and reduced the channels of output from 3 to 1. The other components of AC-GAN, including architecture of the discriminator and auxiliary classifier, are all the same as described in [21].

For classifier architectures, we obtained networks and weights from authors of [10, 16, 17] so that we can be consistent with their papers. The ResNet architectures used for SVHN, CelebA and transferability experiments are shown in Tab. 5.

Hyperparameters of attacks Tab. 4 shows our hyperparameters of all unrestricted adversarial attacks used in this paper, following the same notations in arguments of Algorithm 1.

Table 4: Hyperparameters of unrestricted adversarial attacks. Notations are the same as in Algorithm 1. *Target class is "Male". †Target class is "Female".

Datasets	Classifier	Targeted	Noise	λ_1	λ_2	ϵ	ϵ_{attack}	α	T
MNIST	Madry Net [10]	Yes	No	50	0	0.1	0	1	500
MNIST	Madry Net [10]	Yes	Yes	50	0	0.1	0.3	1	500
MNIST	[16]	No	No	100	0	0.1	0	10	100
MNIST	[17]	No	No	100	0	0.1	0	1	100
SVHN	ResNet	Yes	No	100	100	0.01	0	0.1	200
SVHN	ResNet	Yes	Yes	100	100	0.01	0.03	0.5	300
CelebA*	ResNet	Yes	No	100	100	0.001	0	1	200
CelebA*	ResNet	Yes	Yes	100	100	0.001	0.03	1	200
CelebA†	ResNet	Yes	No	100	100	0.1	0	0.1	200
CelebA†	ResNet	Yes	Yes	100	100	0.1	0.03	0.1	200

Table 5: ResNet Classifier Architecture. We set the base number of feature maps to $m = 4$ for MNIST and $m = 16$ for SVHN and CelebA.

Name	**Configuration**	**Replicate Block**
Initial Layer	3×3 conv. m maps. 1×1 stride.	—
Residual Block 1	batch normalization, leaky relu 3×3 conv. m maps. 1×1 stride batch normalization, leaky relu 3×3 conv. m maps. 1×1 stride residual addition	$\times 10$
Resize Block 1	batch normalization, leaky relu 3×3 conv. $2m$ maps. 2×2 stride batch normalization, leaky relu 3×3 conv. $2m$ maps. 1×1 stride average pooling, padding	—
Residual Block 2	batch normalization, leaky relu 3×3 conv. $2m$ maps. 1×1 stride batch normalization, leaky relu 3×3 conv. $2m$ maps. 1×1 stride residual addition	$\times 9$
Resize Block 2	batch normalization, leaky relu 3×3 conv. $4m$ maps. 2×2 stride batch normalization, leaky relu 3×3 conv. $4m$ maps. 1×1 stride average pooling, padding	—
Residual Block 3	batch normalization, leaky relu 3×3 conv. $4m$ maps. 1×1 stride batch normalization, leaky relu 3×3 conv. $4m$ maps. 1×1 stride residual addition	$\times 9$
Pooling Layer	batch normalization, leaky relu, average pooling	—
Output Layer	10 dense, softmax	—

D Additional samples

(a) Extended plot for untargeted attacks against [16].

(b) Extended plot for untargeted attacks against [17].

Figure 5: (Extended plot of Fig. 1) Random samples of untargeted unrestricted adversarial examples (w/o noise) against certified defenses on MNIST. Green and red borders indicate success/failure respectively, according to MTurk results. The annotation in upper-left corner of each image represents the classifier's prediction.

Source \ Target	0	1	2	3	4	5	6	7	8	9
0		89	93	93	92	86	85	97	92	96
1	70		72	82	82	73	72	82	85	80
2	81	79		92	89	78	78	80	98	86
3	85	90	93		82	85	74	87	80	93
4	84	94	77	85		91	93	92	84	81
5	74	78	70	89	88		79	85	89	89
6	82	95	85	89	91	96		79	90	79
7	59	78	77	83	84	75	47		86	78
8	95	95	92	97	91	95	95	92		97
9	84	80	86	89	94	82	72	97	93	

(a) success rates on MNIST

Source \ Target	0	1	2	3	4	5	6	7	8	9
0		90	95	93	90	94	95	90	87	96
1	82		84	87	86	66	80	83	72	80
2	93	93		94	92	96	96	97	89	95
3	91	96	94		91	97	94	100	90	98
4	89	90	93	91		90	91	90	90	95
5	94	96	97	97	99		97	97	93	97
6	93	79	89	91	93	92		93	90	89
7	89	94	95	93	88	91	88		85	94
8	94	91	89	91	93	95	97	90		99
9	96	93	96	96	93	95	95	92	94	

(b) success rates on SVHN

Figure 6: The success rates (%) of our targeted unrestricted adversarial attack with noise-augmentation.

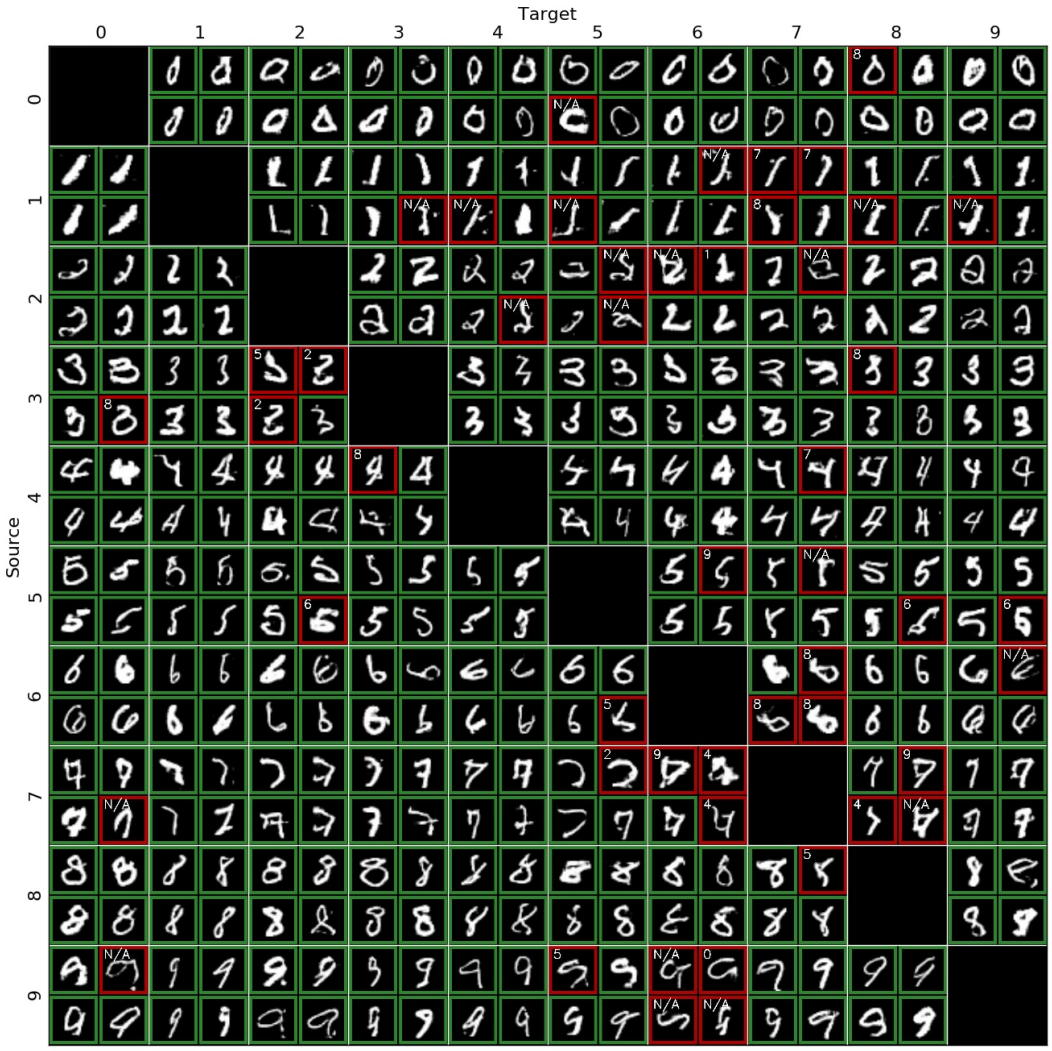


Figure 7: (Extended plot of Fig. 3(a)) Random samples of targeted unrestricted adversarial examples (w/o noise) on MNIST. Green border indicates that the image is voted legitimate by MTurk workers, and red border means the label given by workers (as shown in the upper left corner) disagrees with the image's source class.

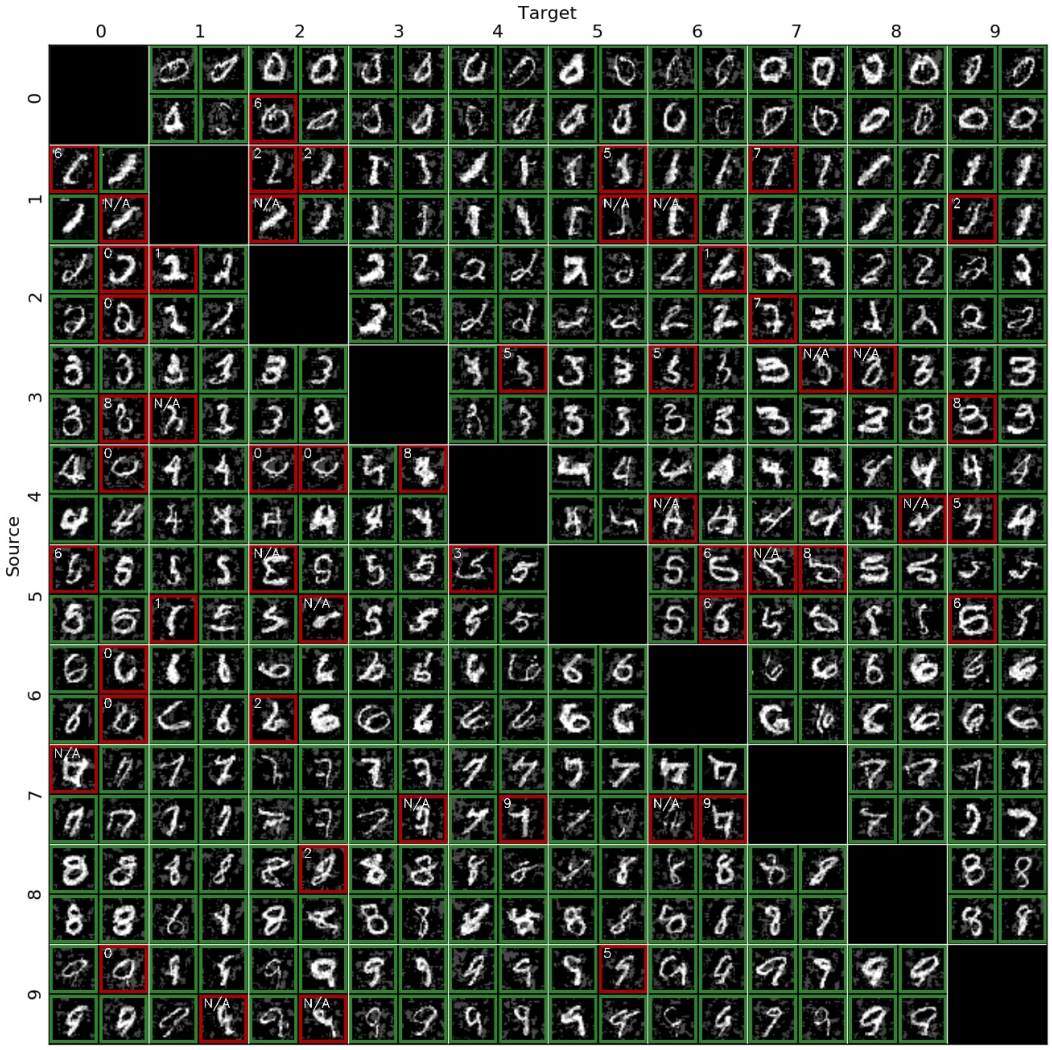


Figure 8: Random samples of targeted unrestricted adversarial examples (w/ noise) on MNIST. Green border indicates that the image is voted legitimate by MTurk workers, and red border means the label given by workers (as shown in the upper left corner) disagrees with the image's source class.

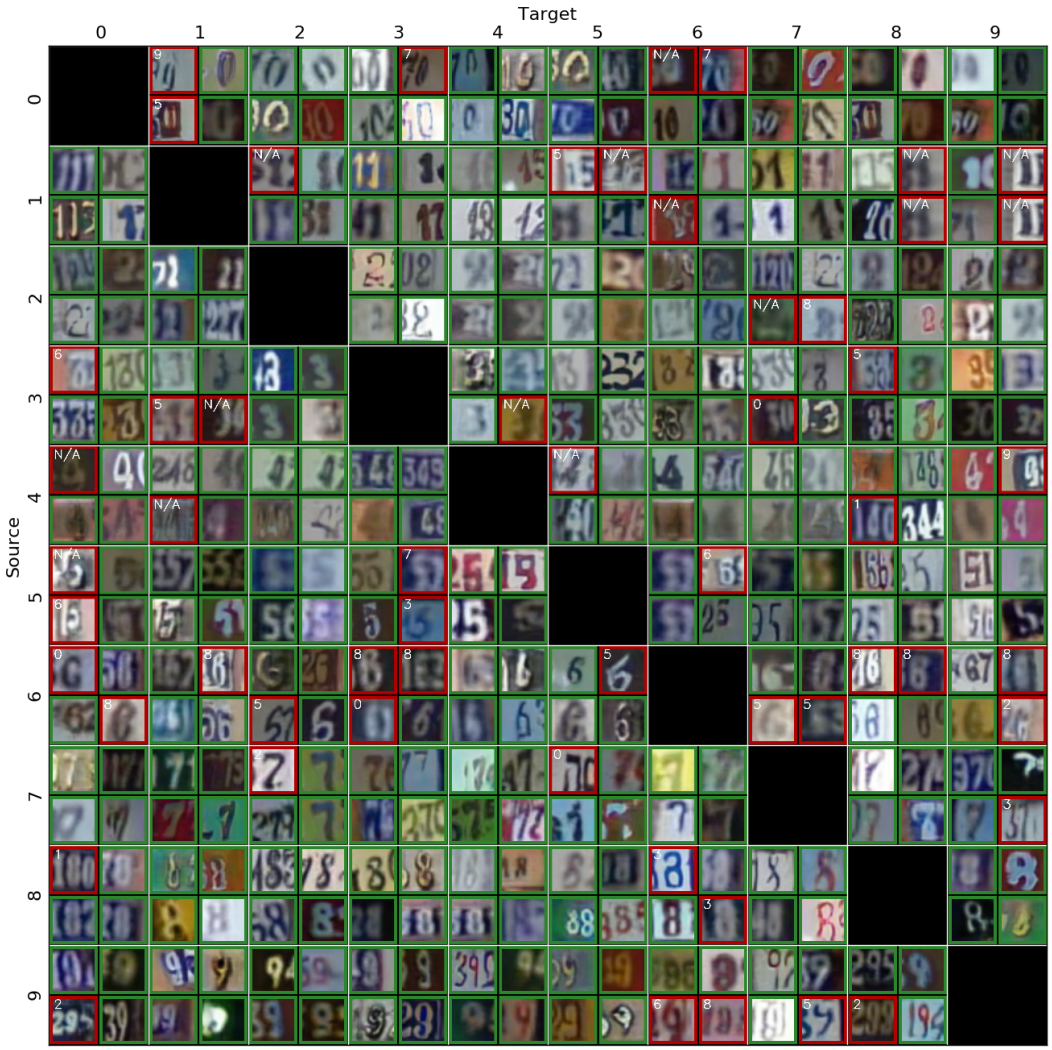


Figure 9: (Extended plot of Fig. 3(c)) Random samples of targeted unrestricted adversarial examples (w/o noise) on SVHN. Green border indicates that the image is voted legitimate by MTurk workers, and red border means the label given by workers (as shown in the upper left corner) disagrees with the image's source class.

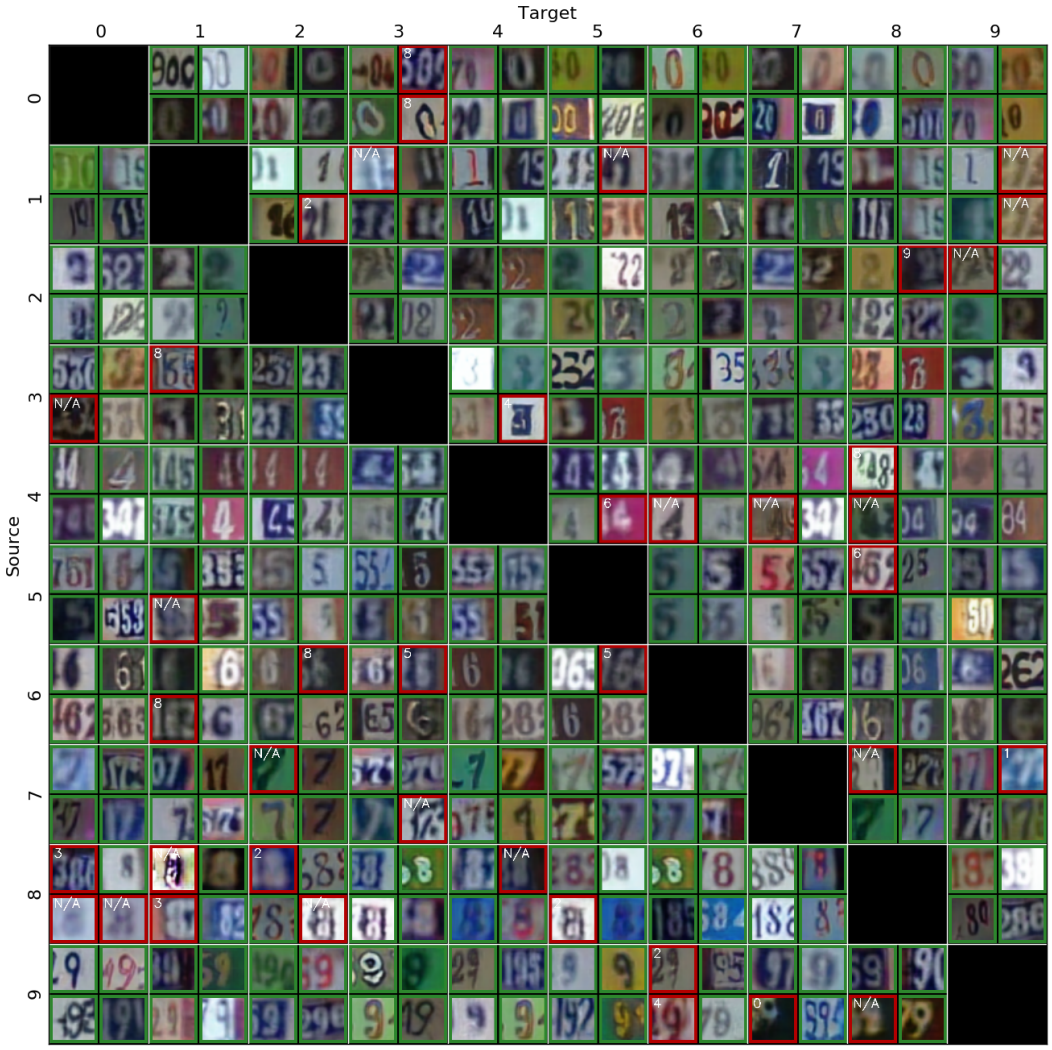


Figure 10: Random samples of targeted unrestricted adversarial examples (w/ noise) on SVHN. Green border indicates that the image is voted legitimate by MTurk workers, and red border means the label given by workers (as shown in the upper left corner) disagrees with the image's source class.

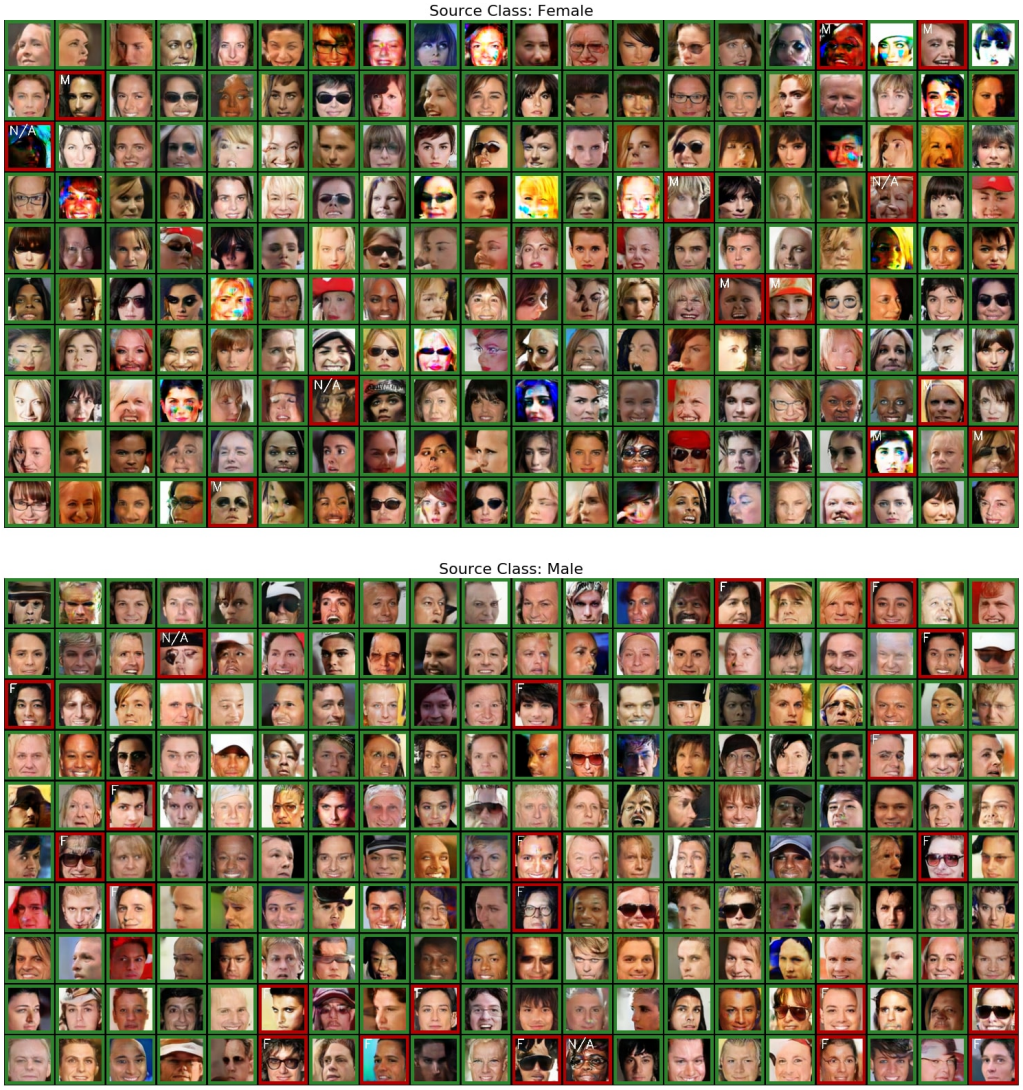
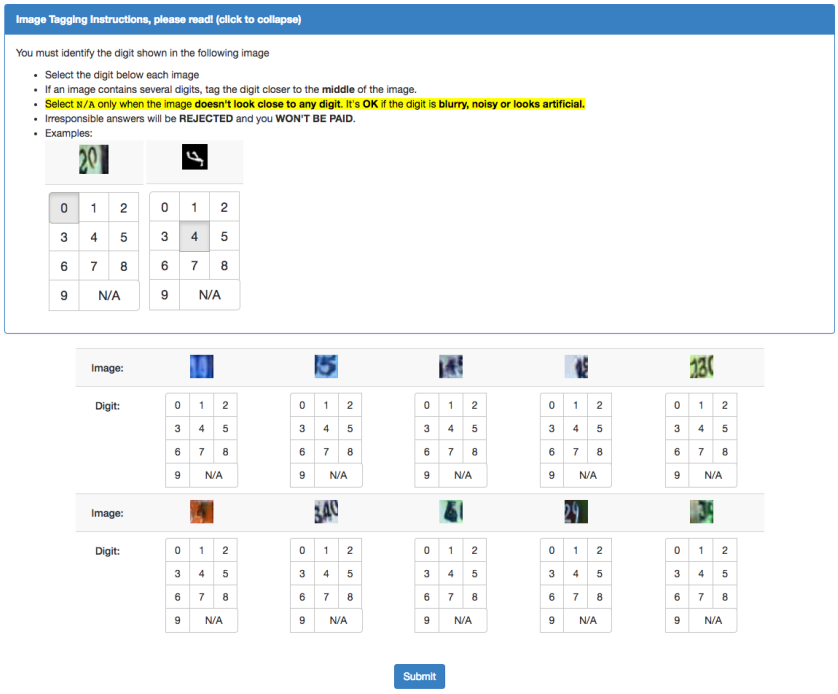


Figure 11: (Extended plot of Fig. 4) Sampled unrestricted adversarial examples (w/o noise) for fooling the classifier to misclassify a female as male (left) and the other way around (right). Green, red borders and annotations have the same meanings as in Fig. 3, except “F” is short for “Female” and “M” is short for “Male”.

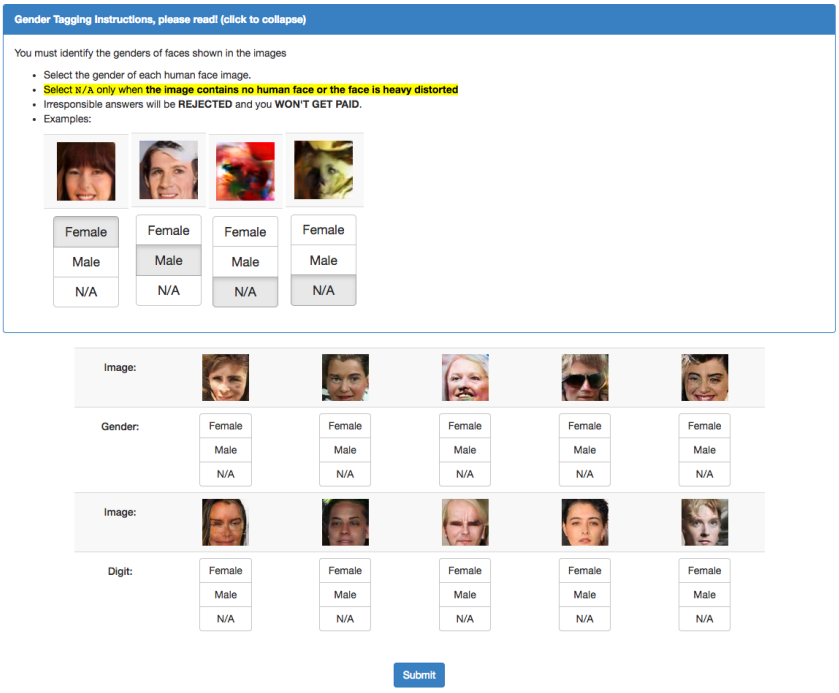
Figure 12: Sampled unrestricted adversarial examples (w/ noise) for fooling the classifier to misclassify a female as male (left) and the other way around (right). Green, red borders and annotations have the same meanings as in Fig. 3, except "F" is short for "Female" and "M" is short for "Male".

E MTurk web interfaces

The MTurk web interfaces used for labeling our unrestricted adversarial examples are depicted in Fig. 13. The A/B test interface used in Section 4.2.1 is shown in Fig. 14. In addition, Fig. 15 visualizes the uncertainty of MTurk annotators for labeling unrestricted adversarial examples, which indicates that more than 40%-50% unrestricted adversarial examples (depending on the dataset) get all of their 5 annotators agreed on one label.



(a) The MTurk web interface for MNIST and SVHN.



(b) The MTurk web interface for CelebA.

Figure 13: MTurk web interfaces for labeling unrestricted adversarial examples.

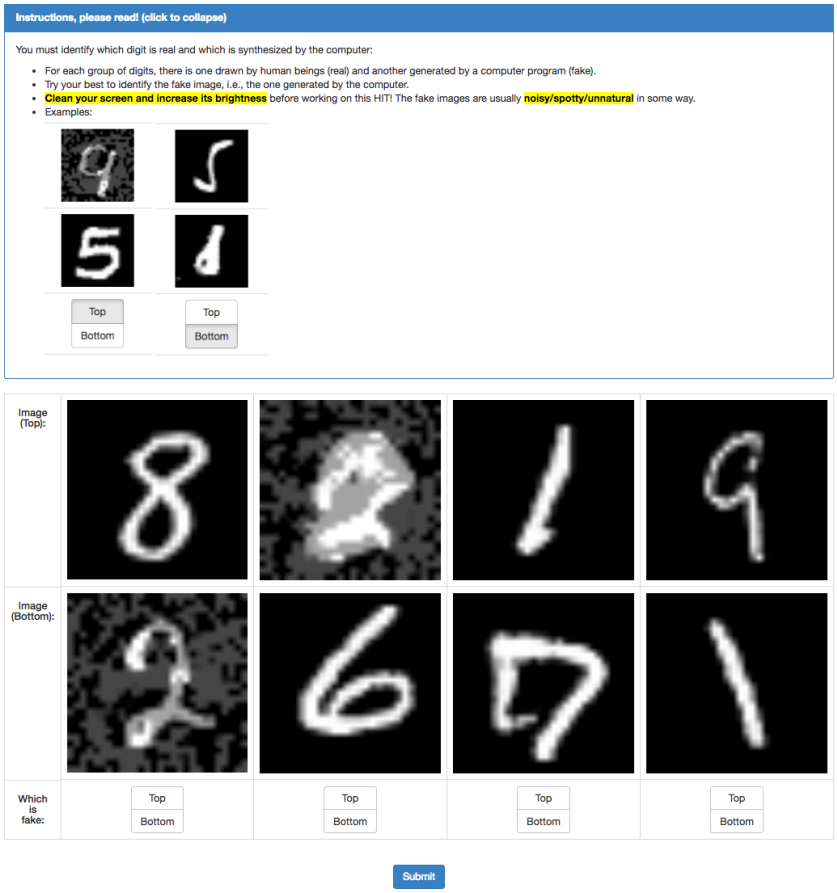


Figure 14: MTurk web interfaces for A/B test on MNIST.

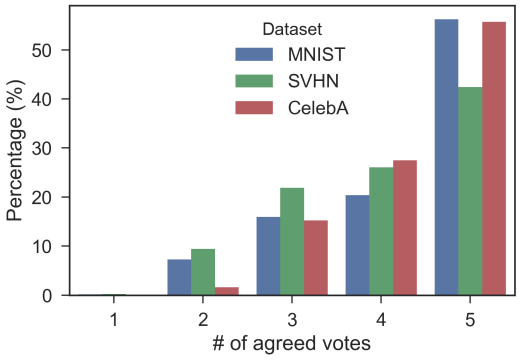


Figure 15: The distribution of number of agreed votes for each image. For example, the red bar on top of 5 means around 55% unrestricted adversarial examples on CelebA dataset are voted the same label by all 5 MTurk annotators.