Towards Universal Adversarial Examples and Defenses

Rakin, Adnan S; Wang, Ye; Aeron, Shuchin; Koike-Akino, Toshiaki; Moulin, Pierre; Parsons, Kieran

TR2021-125 October 21, 2021

Abstract

Adversarial example attacks have recently exposed the severe vulnerability of neural network models. However, most of the existing attacks require some form of target model information (i.e., weights/model inquiry/architecture) to improve the efficacy of the attack. We leverage the information-theoretic connections between robust learning and generalized rate-distortion theory to formulate a universal adversarial example (UAE) generation algorithm. Our algorithm trains an offline adversarial generator to minimize the mutual information of a given data distribution. At the inference phase, our UAE can efficiently generate effective adversary examples without high computation cost.These adversarial examples in turn allow for developing universal defense responses through adversarial training. Our experiments demonstrate promising gains in improving the training efficiency of conventional adversarial training

IEEE Information Theory Workshop 2021

Towards Universal Adversarial Examples and Defenses

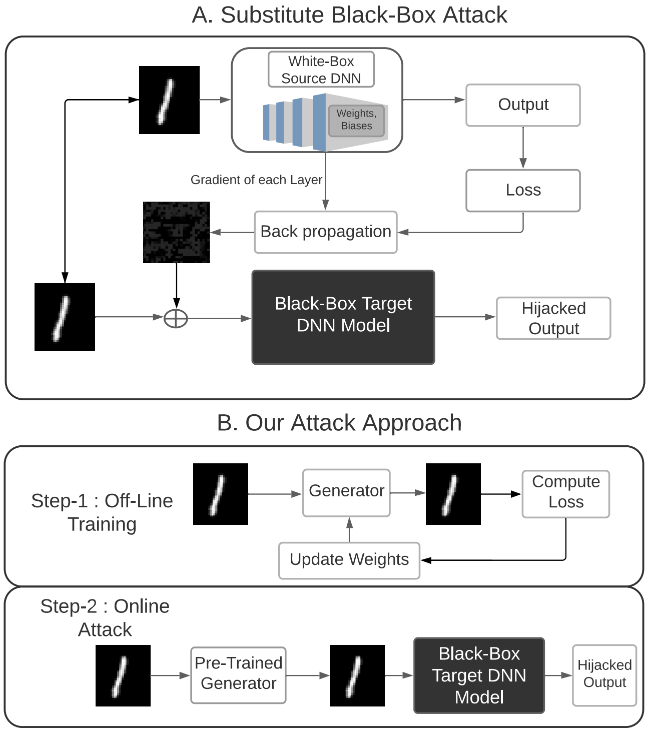
Adnan Siraj Rakin*, Ye Wang†, Shuchin Aeron‡, Toshiaki Koike-Akino†, Pierre Moulin§, Kieran Parsons†
*Arizona State University, †Mitsubishi Electric Research Laboratories (MERL),
‡Tufts University, §University of Illinois at Urbana-Champaign
*asrakin@asu.edu, †{yewang, koike, parsons}@merl.com, ‡shuchin@ece.tufts.edu, §pmoulin@illinois.edu

***Abstract*—Adversarial examples have recently exposed the severe vulnerability of neural network models. However, most of the existing attacks require some form of target model information (i.e., weights/model inquiry/architecture) to improve the efficacy of the attack. We leverage the information-theoretic connections between robust learning and generalized rate-distortion theory to formulate a universal adversarial example (UAE) generation algorithm. Our algorithm trains an offline adversarial generator to minimize the mutual information between the label and perturbed data. At the inference phase, our UAE method can efficiently generate effective adversarial examples without high computation cost. These adversarial examples in turn allow for developing universal defenses through adversarial training. Our experiments demonstrate promising gains in improving the training efficiency of conventional adversarial training.**

I. Introduction

The security of Deep Neural Networks (DNNs) has been under severe scrutiny after being exposed to the threat of adversarial example attacks. The term *adversarial example attack* refers to an attack where an adversary achieves a malicious objective (e.g., accuracy degradation) by adding imperceptible noise to the input of the DNN. Recently, a series of adversarial example methodologies have been proposed [1]–[7] demonstrating the widespread vulnerability of deep learning models.

Adversarial example attacks are categorized under two classes of threat models. First, in a *white-box* adversarial attack [1], [2], the attacker has complete knowledge (e.g., architecture, weights, and gradients) of the target DNN model. As a consequence, these attacks require strong adversarial access to conduct a successful attack, which makes them less practical. Additionally, strong white-box attacks [1], [3], [4] suffer from higher computational overhead (i.e., time and attack iterations). In contrast, the threat model for *black-box attacks* [5], [8] limits information about the target DNN from the attacker. For example, the attacker may only have access to example input and output pairs for the target DNN, which is otherwise treated like a black-box. Alternatively, as shown in Fig. 1(A), a black-box attacker might not utilize any direct knowledge about the target DNN, except by training a substitute/source model with the same training data. This source model could be used to generate an adversarial perturbation that the attacker can add to the input to fool the target black-box DNN. Recent works have focused on improving the efficacy of these black-box attacks by obtaining additional information from the target black-box model through model query [6], [9]–[11].


A. Substitute Black-Box Attack

B. Our Attack Approach

Fig. 1. Overview of traditional black-box attack and our attack framework. A) In a substitute black-box attack [5], [8], the attacker trains a substitute white-box model. Then the attacker feeds the image to be attacked “1” into the substitute model and generates the noise. Finally, the attack adds the noise to the original image to fool a target black-box DNN. B) In contrast, our approach trains an off-line generator to generate a universal adversary. Then at the online stage, the attacker can generate the adversary w/o any back-propagation or model-specific gradient (i.e., strictly black-box). Thus, our UAE attack can be labeled as a universal adversary.

However, in this work, we approach the problem of generating adversarial examples from a more strict black-box scenario perspective. We assume that the attacker has no knowledge of the target DNN and is denied access to any model query. The only available resource to the attacker is the publicly available portion of the training dataset for the given task. Thus, the goal is to generate universal adversarial examples that are independent of any target model information.

Inspired by the formulation in [12], we propose a novel attack method that we call a *Universal Adversarial Example (UAE)* attack. First, we train an off-line generator as shown in Fig. 1(B). The goal of the generator training step is to learn to minimize the mutual information between the label and perturbed data. We train the generator parameters by adopting a recently proposed mutual information gradient estimation (MIGE) [13] method that, instead of estimating the mutual information (a challenging task in high dimensions), directly estimates the gradient of mutual information with respect to parameters given the data samples. Once the generator is trained, it can produce model-independent, universal adversarial examples with low computational cost (i.e., without gradient computations through backpropagation). To summarize our contributions:

1) We formulate a framework for generating model-independent, universal adversarial examples that target the underlying data distribution, by leveraging the connections between robust learning and generalized rate-distortion theory.
2) We propose a novel attack methodology called *universal adversarial example (UAE)*, which trains an off-line adversarial generator via mutual information gradient estimation. The proposed UAE can generate fast, effective, model-independent adversarial examples.
3) We demonstrate the utility of these adversarial examples towards training universal defenses with considerable speed-up over conventional methods.

II. Related Work

Recent works have investigated different ways of enhancing black-box attacks. An ensemble network approach [14], [15] was proposed to increase the efficacy of substitute model black-box attacks. Additionally, [6], [9]–[11] considered exploiting queries with the target model to extract valuable information (i.e., about the decision rule) to generate an improved transferable adversary. However, these approaches come at the expense of additional memory, computational overhead, or online access required to send multiple query requests.

In contrast, our goal is to generate *universal adversarial examples* independent of any model-specific information. Prior works [16]–[19] have also referred to the term *universal adversarial perturbation* indicating a set of adversarial examples generated without any information about the target data set. We acknowledge that a data-independent universal adversary is useful in several applications such as black-box attacks. However, one drawback of these techniques is they are not model-agnostic. In fact, existing black-box attacks [20], [21] that train a specific generator crafts adversarial samples to fool a specific target model. In our approach, we shift the paradigm of universal adversary from being data-independent to a model-independent scenario.

While our work focuses on the problem of adversarial examples and robust machine learning, we note that our fundamental approach is also immediately applicable to data privacy problems [22]–[27]. The optimal privacy-utility tradeoffs can also be formulated as a constrained maximum conditional entropy problem [22], as later given by Eq. (6). Our approach, as a novel data-driven method to solve the maximum entropy problem, provides an alternative to other methods [28], [29].

III. Methodology

A. Problem Formulation

Popular white-box attack algorithms [30] utilize model-specific gradient information to maximize the conditional entropy of a given target model f_α parameterized by α. The goal of such a worst-case perturbation mechanism can be formalized as follows:

$$\max_{\substack{Z\in\mathcal{X}:\\ d(X,Z)\le\epsilon}} \ell(f_\alpha(Z),Y), \tag{1}$$

where ℓ denotes the loss function for the model f_α, X is the original input data with the domain $\mathcal{X}$, Y is the corresponding label in $\mathcal{Y} := \{1,\dots,m\}$, and Z is the attack within the allowable perturbation $\epsilon \ge 0$, with respect to some suitably chosen distortion metric $d : \mathcal{X}\times\mathcal{X}\to[0,\infty]$ (e.g., often ℓ_0, ℓ_p, or ℓ_∞ norm distance). A popular approach [1] to defend against a worst-case adversary attempts to solve the following minimax optimization problem, via adversarial training:

$$\min_\alpha \mathbb{E}\big[\max_{\substack{Z\in\mathcal{X}:\\ d(X,Z)\le\epsilon}} \ell(f_\alpha(Z),Y)\big], \tag{2}$$

where the expectation is over the distribution of (X,Y).

Extending the above formulation, [12] considers a stronger stochastic adversary, allowing for mixed strategies, as captured by a noisy channel $P_{Z|X,Y}\in\mathcal{D} := \big\{P_{Z|X,Y} : \Pr[d(X,Z)\le\epsilon] = 1\big\}$. For the ideal optimization over all decision rules $q\in\mathcal{P}(\mathcal{Y}|\mathcal{X})$, and with cross-entropy as the loss function, i.e., $\ell(q(Y|Z)) := -\log q(Y|Z)$, [12] establishes the following minimax equality and equivalence with the maximum conditional entropy problem:

$$\begin{aligned}\min_q \max_{P_{Z|X,Y}\in\mathcal{D}} &\mathbb{E}[-\log q(Y|Z)] \quad (3)\\ = \max_{P_{Z|X,Y}\in\mathcal{D}} \min_q &\mathbb{E}[-\log q(Y|Z)] \quad (4)\\ = \max_{P_{Z|X,Y}\in\mathcal{D}} &H(Y|Z). \quad (5)\end{aligned}$$

Since the entropy $H(Y)$ is constant over $P_{Z|X,Y}\in\mathcal{D}$, the optimization can be equivalently formulated as minimization of mutual information $I(Z;Y)$ as follows:

$$\max_{P_{Z|X,Y}\in\mathcal{D}} H(Y|Z) = H(Y) - \min_{P_{Z|X,Y}\in\mathcal{D}} I(Z;Y). \tag{6}$$

B. UAE Attack Method

Our proposed UAE generation method is trained to minimize the mutual information of Eq. (6). The generator model produces $Z = g_\theta(X,Y)$, where g_θ is a parameterized, random mapping, which is specifically realized as a neural network with noise samples as an auxiliary input to provide randomization. The neural network is constructed to ensure that the distortion limit is not exceeded, i.e., $\Pr[d(X,Z)\le\epsilon] = 1$, by projecting the output Z back within the distortion limit, if it is exceeded. The model g_θ implicitly defines the conditional distribution $P_\theta(Z|X,Y)$, and combined with the underlying data distribution $P_{X,Y}$, also implicitly defines the marginal $P_\theta(Z)$ and conditional $P_\theta(Z|Y)$ distributions.

Our approach comprises two steps: (1) *Off-Line Generator Training*, and (2) *Online Adversarial Example Generation*. After the adversarial generator is trained in the first step, universal adversarial examples can be generated with low computational cost in the second step.

1) Off-Line Generator Training: We train the UAE generator g_θ to minimize the mutual information as formulated in Eq. (6). To perform this optimization, given access to only the samples, we avoid direct calculation/estimation of the true mutual information, and instead approximate its gradient using the MIGE method [13] applied to the expansion:

$$\begin{aligned}\nabla_\theta I(Z;Y) &= \nabla_\theta H(Z) - \frac{1}{m}\sum_{y=1}^{m} \nabla_\theta H(Z|Y=y) \qquad (7)\\ &= \nabla_\theta \mathbb{E}[-\log P_\theta(Z)]\\ &\quad + \frac{1}{m}\sum_{y=1}^{m} \nabla_\theta \mathbb{E}[\log P_\theta(Z|Y)|Y=y]. \qquad (8)\end{aligned}$$

The MIGE method efficiently estimates the gradient of mutual information by estimating the score functions for the implicit distributions in the first and second terms given in Eq. (8). Specifically, it employs a change of variables [13], [31] to rewrite these terms, with $Z = g_\theta(X,Y)$, as follows:

$$\begin{aligned}\nabla_\theta I(Z;Y) &= \mathbb{E}[-\nabla_Z \log P_\theta(Z) \nabla_\theta g_\theta(X,Y)]\\ &+ \frac{1}{m}\sum_{y=1}^{m} \mathbb{E}[\nabla_Z \log P_\theta(Z|Y) \nabla_\theta g_\theta(X,Y)|Y=y]. \qquad (9)\end{aligned}$$

Then, we use the Spectral Stein Gradient Estimator (SSGE) [31] to estimate the terms $\nabla_Z \log P_\theta(Z)$ and $\nabla_Z \log P_\theta(Z|Y=y)$ from samples drawn from $P_\theta(Z)$ and $P_\theta(Z|Y=y)$, respectively. The overall off-line training stage is summarized in Algorithm 1.

2) Online Attack Generation: After completing the off-line generator training step, the attacker can use the pre-trained UAE generator g_θ to generate universal adversarial examples as $Z = g_\theta(X,Y)$. As a result, our UAE attack samples do not rely on any target model information. Such a universal adversary can be used to attack any black-box target model with negligible computational overhead. In contrast, most of the existing black-box and white-box methods require adjusting the adversarial information depending on the target classifier model. Such an adjustment is often computationally expensive. Finally, our UAE generated samples can be used to perform adversarial training with minor cost as well. We refer to this defense as a *universal defense response* since the training samples again do not depend on any specific neural network model.

Algorithm 1 Off-line UAE generator training

```
procedure TRAIN GENERATOR
    Initialize the generator model parameters $\theta$.
    for each training iteration do
        Prepare a data batch $\{(X_i, Y_i)\}_{i=1}^{n} \overset{\text{iid}}{\sim} P_{X,Y}$.
        Apply the generator to produce $Z_i = g_\theta(X_i, Y_i)$.
        Estimate $\nabla_Z \log P_\theta(Z)$ using SSGE over the entire batch of samples $\{Z_i\}_{i=1}^{n}$.
        for each class $y = 1, \ldots, m$ do
            Select data samples corresponding to class $y$.
            Estimate $\nabla_Z \log P_\theta(Z|Y = y)$ using SSGE over only the samples $Z_i$, where the corresponding $Y_i = y$.
        end for
        Compute the mutual information gradient according to Eq. (9), with expectations approximated by the empirical mean over the data batch.
        Perform a gradient descent step to update $\theta$.
    end for
    Return: Trained generator $g_\theta$.
end procedure
```

IV. Experimental Setup

A. Dataset

As a proof of concept, we evaluate our UAE method on the MNIST dataset [32], which contains 28 × 28 pixel, gray-scale images of hand-written digits. We use the standard train-test data split with 60K images for the training set and the remaining 10K for testing.

B. Baseline Attacks

For comparison to a baseline white-box attack, we use the Projected Gradient Descent (PGD) [1] attack. PGD is a popular method used in adversarial training, where the model is jointly trained with clean and attack examples to enhance robustness [1]. In our evaluation, we refer to this model as the *PGD trained model* for a specific ϵ and metric (e.g., ℓ_2 or ℓ_∞ distance in our experiments). The baseline model is labeled as the *clean model* indicating that no adversarial examples were used in the training phase.

For the black-box *substitute model* attack [5], [8], we used a substitute model attack as depicted in Fig. 1. For our attack, we used the MNIST training set to train the generator. Thus to keep the comparison fair with our attack, we trained the substitute model for the black-box attack using the same MNIST training dataset. Then, during evaluation, we used the substitute/source model to generate PGD samples and to perform the attack on the target model using these samples.

C. Attack Settings

For the PGD attack and PGD training, our attack uses ℓ_∞ distance with $\epsilon = 0.3$, an attack step-size of $2/255$, and attack iterations of 40, unless specified otherwise. For our attack, we train the generator model for 20 iterations with the standard stochastic gradient descent (SGD) optimizer of PyTorch with

TABLE I
BLACK-BOX ATTACK PERFORMANCE ON A CLEAN MODEL (99.28% ACCURACY). REPORTING THE TEST ACCURACY OF THE MODEL AFTER ATTACKING WITH DIFFERENT (ℓ_2 DISTANCE ϵ) ATTACK.

Attack	4	4.5	5	5.5	6	6.5	7	7.5
Substitute	81.86	73.78	63.52	52.3	40.59	30.14	21.17	13.85
UAE	91.46	82.15	65.02	49.25	36.65	27.52	21.18	16.92

a learning rate of 0.01 and momentum of 0.9. We run each attack experiment three times and report the mean with error margin (i.e., $\pm$ 1.68$\times$ standard deviation) in Fig. 3 & Table II. To compute the gradient of the implicit data distribution, we use SSGE [33][1]. We record the attack generation time of the attack methods on NVIDIA GeForce GTX 1080 Ti GPU. We measure the attack time in seconds (s) after attacking the 10K MNIST testing samples and evaluate the test accuracy in our platform.

D. Model Architecture

In our experiment, we used a four-layer DNN as the classifier model for the MNIST dataset. This classifier DNN model contains two convolution layers and two fully connected layers[2]. Our generator is a multi-layer perceptron with three fully connected layers. Finally, as a source model for substitute black-box attack, we used a two-layer fully-connected neural network. Later in Section VI, we demonstrate that even a smaller (two-layer) black-box substitute model requires more time to generate adversarial examples than our generator (three-layer).

V. RESULTS

A. Universal Attack

The black-box attack evaluation under a given ℓ_2 distance constraint is presented in Table I. We vary the ℓ_2 distance from 4 to 7.5 and report the test accuracy for a clean model. The test accuracy of this model without any attack (i.e., $\ell_2 = 0$) is 99.28%. After attack, both the substitute and UAE attacks degrade the test accuracy with increasing attack strength (i.e., higher ϵ). At low ϵ, the substitute model shows better performance; but with increasing ϵ, in particular at $\epsilon = 5.5$–6.5, UAE outperforms the substitute model attack. As the UAE attack does not rely on any model-specific information, at a large $\epsilon = 7.5$ the attack can at best degrade the accuracy close to a random guess (10% for a 10-class problem).

Our UAE attack generated samples can also be useful for stronger white-box attacks (e.g., PGD). For example, before a PGD attack we can initialize the samples with our universal adversary (labeled as *PGD with universal initialization*) instead of random initialization. As shown in Fig. 2, our UAE initialization can increase the strength of the PGD attack in comparison to random initialization. For a fixed attack step-size and ϵ, our universal initialization consistently achieves better attack performance than the baseline PGD method.

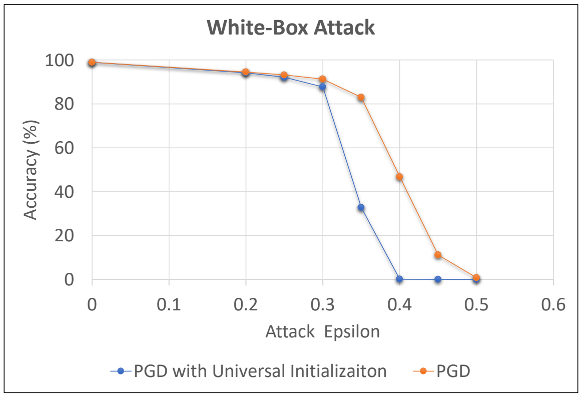


Fig. 2. We attack a PGD ($\epsilon = 0.3$) trained model using PGD attack in orange and using PGD examples generated from universal initialization in blue.

TABLE II
MODEL TRAINED WITH ADVERSARIAL EXAMPLES FOR TWO CASES: I) WITH PGD SAMPLES; II) WITH PGD SAMPLES GENERATED USING OUR UNIVERSAL INITIALIZATION. WE ATTACK BOTH CASES WITH PGD ATTACK BY VARYING ϵ FROM 0.2 TO 0.35 TO REPORT ACCURACY.

Method	0	0.2	0.25	0.3	0.35
PGD	98.95	94.5 $\pm$ 0.06	93.1 $\pm$ 0.03	91.3 $\pm$ 0.05	83.08 $\pm$ 0.1
PGD with UAE	98.81	94.8 $\pm$ 0.3	93.7 $\pm$ 0.5	92.5 $\pm$ 0.3	86.97 $\pm$ 2.3

B. Universal Defense Response

As prior observation shows (Fig. 2), the UAE adversary can enhance the performance of PGD attack, we utilize those adversaries to produce a defense response. In Table II, we train two models using adversarial examples of ℓ_∞ distance $\epsilon = 0.3$. The first one uses adversarial examples generated by the PGD algorithm with random initialization and the second one uses adversarial examples generated from the PGD algorithm with universal initialization. Then, we attack both models using the white-box PGD attack for $\epsilon = 0.2, \ldots, 0.35$. Our initialization during adversarial training helps to enhance the robustness against PGD attacks. In particular, at $\epsilon = 0.35$ (in Table II), our UAE PGD training can improve the test accuracy by approximately 4% in comparison to PGD training. Our UAE attack can provide an enhanced initialization method for a strong white-box attack, but this incurs the heavy computation burden of the subsequent PGD iterations during training.

As a low-complexity alternative, we directly train a classifier model using our universal adversarial examples in Fig. 3. To compare this universal defense response with a similar method, we use adversarial training with the substitute black-box adversarial examples, which also has a low computation cost similar to UAE. Both our attack and black-box PGD have comparable computation costs, because in both cases the adversarial examples can be generated even before the training starts. Finally, a clean model is also evaluated as a baseline. We train these three models on three different ℓ_∞ distance attacks (e.g., at $\epsilon = 0.5/0.4/0.2$). For each case, we attack these three models with a white-box PGD attack of different strengths ($\epsilon = 0.2, \ldots, 0.6$). In Fig. 3, our UAE attack based adversarial training (blue) shows better resistance to the white-box PGD attack for a range of attack strengths. In contrast, the

[1] https://github.com/zhouyiji/MIGE/tree/master/toy
[2] https://github.com/pytorch/examples/tree/master/mnist

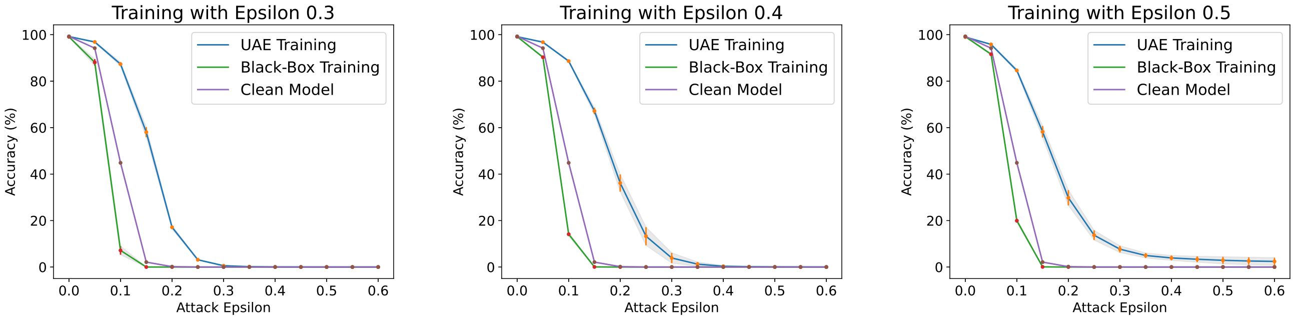


Fig. 3. Effect of training a model with adversarial examples of ℓ_∞ distance $\epsilon = 0.5, 0.4, 0.3$. In each case, we attack three models with a white-box PGD attack: i) Clean (i.e., no adversarial training) (violet), ii) Black-box adversarial trained (green), and iii) UAE trained model (blue).

TABLE III
COMPARISON OF ATTACK TIME IN SECONDS (S): RELATIVE TIME COST OF SUBSTITUTE AND PGD ATTACKS IN COMPARISON TO OUR ATTACK.

UAE	Substitute	PGD
0.913s (1×)	1.12s (1.2×)	11.37s (12.5×)

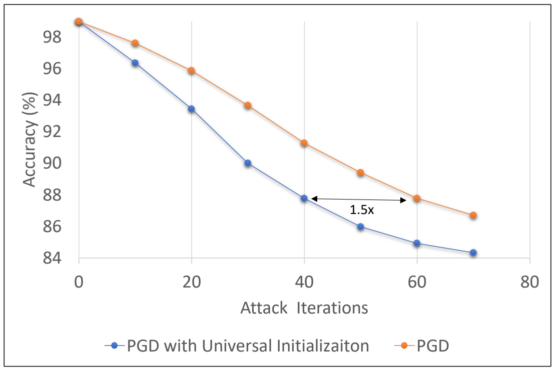


Fig. 4. Test accuracy *vs* number of attack iterations on a PGD ($\epsilon = 0.3$) trained model.

black-box PGD (green) makes the model even more vulnerable to PGD attack than a clean model (violet).

VI. Analysis

1) Time Analysis: The amount of time required to conduct the PGD attack, black-box substitute model attack, and UAE attack on a clean model is summarized in Table III. Our UAE attack is about 12× faster than the PGD white-box attack and is slightly faster than the substitute model attack. Such a pre-trained generator provides two major benefits: i) UAE can generate adversarial examples faster and ii) the pre-trained generator can generate the adversarial examples before the training starts for adversarial training.

In summary, the UAE attack is a fast and efficient way of generating adversarial examples, which are independent of any target model (i.e., universal). The pre-trained generator opens door for augmenting the training data off-line (e.g., even before training starts). Hence, it will be a useful tool enabling large-scale and computationally efficient adversarial training.

2) Computation Reduction: In this section, we evaluate the effect of attack iterations on the efficacy of PGD examples with our universal initialization. Fig. 4 shows that UAE initialization reduces the number of attack iterations of the white-box PGD attack. It is possible to achieve the same attack efficacy as a 60-iteration white-box PGD attack with only 40 attack iterations using our UAE initialization. Thus, universal initialization reduces the attack computation by 1.5× while achieving the same attack strength (e.g., about 87% test accuracy). In fact, even single step adversarial attacks (i.e., FGSM [2]) requires generating adversarial samples during each iteration of the training process for each trained model. In contrast, UAE can generate the training samples once universally for any model at any given training iteration. Practically, such reduction in computation may save hundreds of GPU training hours during large-scale adversarial training. This shows that the UAE reduces the attack complexity of a white-box attack, and hence also the complexity of an adversarial training defense.

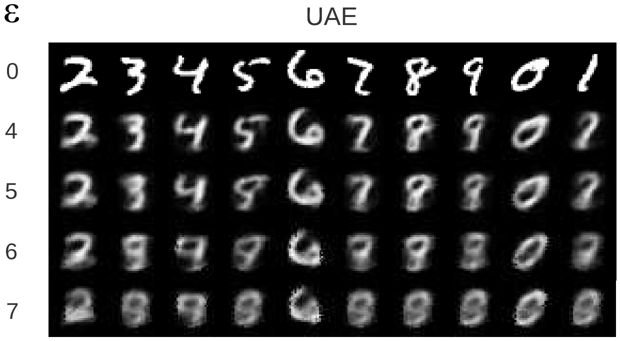


Fig. 5. UAE attack images for several values of ℓ_2 distance, ϵ. For large ϵ, the UAE attack is stronger and can potentially confuse a human eye.

3) Visualization: We visualize sample UAE attack images in Fig. 5. Increasing the attack strength ϵ reduces the dissimilarity between images in different classes.

VII. Conclusion

Our proposed UAE attack can generate universal adversarial examples in a model-independent manner. Such a universal attack can be as effective as a black-box substitute attack model, while reducing the attack time as well. Additionally, these samples can be applied to adversarial training to reduce the computational costs.

References

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[2] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[3] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 274–283.

[4] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 3–14.

[5] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[6] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[7] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[8] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[9] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.

[10] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," *arXiv preprint arXiv:1906.06919*, 2019.

[11] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 742–749.

[12] Y. Wang, S. Aeron, A. S. Rakin, T. Koike-Akino, and P. Moulin, "Robust machine learning via privacy/rate-distortion theory," *arXiv preprint arXiv:2007.11693*, 2020.

[13] L. Wen, Y. Zhou, L. He, M. Zhou, and Z. Xu, "Mutual information gradient estimation for representation learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ByxaUgrFvH

[14] Z. Huang and T. Zhang, "Black-box adversarial attack with transferable model-based embedding," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJxhNTNYwB

[15] J. Hang, K. Han, H. Chen, and Y. Li, "Ensemble adversarial black-box attacks against deep learning systems," *Pattern Recognition*, vol. 101, p. 107184, 2020.

[16] K. Reddy Mopuri, P. Krishna Uppala, and R. Venkatesh Babu, "Ask, acquire, and attack: Data-free UAP generation using class impressions," *arXiv e-prints*, pp. arXiv–1808, 2018.

[17] D. B. Sam, K. Sudharsan, and V. B. Radhakrishnan, "Crafting data-free universal adversaries with dilate loss," 2019.

[18] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2452–2465, 2018.

[19] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," *arXiv preprint arXiv:1707.05572*, 2017.

[20] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," *arXiv preprint arXiv:1805.07894*, 2018.

[21] S. Baluja and I. Fischer, "Learning to attack: Adversarial transformation networks," in *Thirty-second aaai conference on artificial intelligence*, 2018.

[22] F. d. P. Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1401–1408.

[23] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, 2013.

[24] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *2014 IEEE Information Theory Workshop (ITW 2014)*, 2014, pp. 501–505.

[25] S. Salamatian, A. Zhang, F. du Pin Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "Managing your private and public data: Bringing down inference attacks against your privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1240–1255, 2015.

[26] Y. O. Basciftci, Y. Wang, and P. Ishwar, "On privacy-utility tradeoffs for constrained data release mechanisms," in *2016 Information Theory and Applications Workshop (ITA)*, 2016, pp. 1–6.

[27] Y. Wang, Y. O. Basciftci, and P. Ishwar, "Privacy-utility tradeoffs under constrained data release mechanisms," *arXiv preprint arXiv:1710.09295*, 2017.

[28] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," *arXiv preprint arXiv:1712.07008*, 2017.

[29] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, 2017.

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1706.06083

[31] J. Shi, S. Sun, and J. Zhu, "A spectral approach to gradient estimation for implicit distributions," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4644–4653.

[32] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: a case study in handwritten digit recognition," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*, vol. 2. IEEE, 1994, pp. 77–82.

[33] Y. Li and R. E. Turner, "Gradient estimators for implicit models," *arXiv preprint arXiv:1705.07107*, 2017.