Adversarial Examples Can Be Effective Data Augmentation for Unsupervised Machine Learning

Chia-Yi Hsu¹, Pin-Yu Chen², Songtao Lu², Sijia Liu³, Chia-Mu Yu¹
¹National Yang Ming Chiao Tung University
²IBM Research
³Michigan State University
{chiayihsu8315, chiamuyu}@gmail.com, {pin-yu.chen, songtao}@ibm.com, liusiji5@msu.edu

Abstract
Adversarial examples causing evasive predictions are widely used to evaluate and improve the robustness of machine learning models. However, current studies focus on supervised learning tasks, relying on the ground-truth data label, a targeted objective, or supervision from a trained classifier. In this paper, we propose a framework of generating adversarial examples for unsupervised models and demonstrate novel applications to data augmentation. Our framework exploits a mutual information neural estimator as an information-theoretic similarity measure to generate adversarial examples without supervision. We propose a new MinMax algorithm with provable convergence guarantees for efficient generation of unsupervised adversarial examples. Our framework can also be extended to supervised adversarial examples. When using unsupervised adversarial examples as a simple plug-in data augmentation tool for model retraining, significant improvements are consistently observed across different unsupervised tasks and datasets, including data reconstruction, representation learning, and contrastive learning. Our results show novel methods and considerable advantages in studying and improving unsupervised machine learning via adversarial examples.

1 Introduction
Adversarial examples are known as prediction-evasive attacks on state-of-the-art machine learning models (e.g., deep neural networks), which are often generated by manipulating native data samples while maintaining high similarity measured by task-specific metrics such as $L_p$-norm bounded perturbations (Goodfellow, Shlens, and Szegedy 2015; Biggio and Roli 2018). Due to the implications and consequences on mission-critical and security-centric machine learning tasks, adversarial examples are widely used for robustness evaluation of a trained model and for robustness enhancement during training (i.e., adversarial training).

Despite a plethora of adversarial attacking algorithms, the design principle of existing methods is primarily for supervised learning models — requiring either the true label or a targeted objective (e.g., a specific class label or a reference sample). Some recent works have extended to the semi-supervised setting, by leveraging supervision from a classifier (trained on labeled data) and using the predicted labels on unlabeled data for generating (semi-supervised) adversarial examples (Miyato et al. 2018; Zhang et al. 2019; Stanforth et al. 2019; Carmon et al. 2019). On the other hand, recent advances in unsupervised and few-shot machine learning techniques show that task-invariant representations can be learned and contribute to downstream tasks with limited or even without supervision (Ranzato et al. 2007; Zhu and Goldberg 2009; Zhai et al. 2019), which motivates this study regarding their robustness. Our goal is to provide efficient robustness evaluation and data augmentation techniques for unsupervised (and self-supervised) machine learning models through unsupervised adversarial examples (UAEs).

Table 1 summarizes the fundamental difference between conventional supervised adversarial examples and our UAEs. Notably, our UAE generation is supervision-free because it solely uses an information-theoretic similarity measure and the associated unsupervised learning objective function. It does not use any supervision such as label information or prediction from other supervised models.
In this paper, we aim to formalize the notion of UAE, establish an efficient framework for UAE generation, and demonstrate the advantage of UAEs for improving a variety of unsupervised machine learning tasks. We summarize our main contributions as follows.

- We propose a new per-sample based mutual information neural estimator (MINE) between a pair of original and modified data samples as an information-theoretic similarity measure and a supervision-free approach for generating UAE. For instance, see UAEs for data reconstruction in Figure ?? of supplementary material. While our primary interest is generating adversarial examples for unsupervised learning models, we also demonstrate that our per-sample MINE can be used to generate adversarial examples for supervised learning models with improved visual quality.
- We formulate the generation of adversarial examples with MINE as a constrained optimization problem, which applies to both supervised and unsupervised machine learning tasks. We then develop an efficient MinMax optimization algorithm (Algorithm 1) and prove its convergence. We also demonstrate the advantage of our MinMax algorithm over the conventional penalty-based method.
- We show a novel application of UAEs as a simple plug-in data augmentation tool for several unsupervised machine learning tasks, including data reconstruction, representation learning, and contrastive learning on image and tabular datasets. Our extensive experimental results show outstanding performance gains (up to 73.5% performance improvement) by retraining the model with UAEs.

2 Related Work and Background

2.1 Adversarial Attack and Defense

For supervised adversarial examples, the attack success criterion can be either untargeted (i.e. model prediction differs from the true label of the corresponding native data sample) or targeted (i.e. model prediction targeting a particular label or a reference sample). In addition, a similarity metric such as $L_p$-norm bounded perturbation is often used when generating adversarial examples. The projected gradient descent (PGD) attack (Madry et al. 2018) is a widely used approach to find $L_p$-norm bounded supervised adversarial examples. Depending on the attack threat model, the attacks can be divided into white-box (Szegedy et al. 2013; Carlini and Wagner 2017b), black-box (Chen et al. 2017; Brendel, Rauber, and Bethge 2018; Liu et al. 2020), and transfer-based (Nitin Bhagoji et al. 2018; Papernot et al. 2017) approaches.

Although a plethora of defenses were proposed, many of them failed to withstand advanced attacks (Carlini and Wagner 2017a; Athalye, Carlini, and Wagner 2018). Adversarial training (Madry et al. 2018) and its variants aiming to generate worst-case adversarial examples during training are so far the most effective defenses. However, adversarial training on supervised adversarial examples can suffer from undesirable tradeoff between robustness and accuracy (Su et al. 2018; Tsipras et al. 2019). Following the formulation of untargeted supervised attacks, recent studies such as (Cemgil et al. 2020) generate adversarial examples for unsupervised tasks by finding an adversarial example within an $L_p$-norm perturbation constraint that maximizes the training loss. In contrast, our approach aims to find adversarial examples that have low training loss but are dissimilar to the native data (see Table 1), which plays a similar role to the category of “on-manifold” adversarial examples governing generalization errors (Stutz, Hein, and Schiele 2019). In supervised setting, (Stutz, Hein, and Schiele 2019) showed that adversarial training with $L_p$-norm constrained perturbations may find off-manifold adversarial examples and hurt generalization.

2.2 Mutual Information Neural Estimator

Mutual information (MI) measures the mutual dependence between two random variables $X$ and $Z$, defined as $I(X, Z) = H(X) - H(X|Z)$, where $H(X)$ denotes the (Shannon) entropy of $X$ and $H(X|Z)$ denotes the conditional entropy of $X$ given $Z$. Computing MI can be difficult without knowing the marginal and joint probability distributions ($P_X$, $P_Z$, and $P_{XZ}$). For efficient computation, the mutual information neural estimator (MINE) with consistency guarantees is proposed in (Belghazi et al. 2018). Specifically, MINE aims to maximize the lower bound of the exact MI using a model parameterized by a neural network $\theta$, defined as $I_\theta(X, Z) \leq I(X, Z)$, where $\Theta$ is the space of feasible parameters of a neural network, and $I_\theta(X, Z)$ is the neural information quantity defined as $I_\theta(X, Z) = \sup_{\Theta} \mathbb{E}_{P_{XZ}}[T_\theta] - \log(\mathbb{E}_{P_X \otimes P_Z}[e^{T_\theta}])$. The function $T_\theta$ is parameterized by a neural network $\theta$ based on the Donsker-Varadhan representation theorem (Donsker and Varadhan 1983). MINE estimates the expectation of the quantities above by shuffling the samples from the joint distribution along the batch axis or using empirical samples $\{x_i, z_i\}_{i=1}^n$ from $P_{XZ}$ and $P_X \otimes P_Z$ (the product of marginals).

MINE has been successfully applied to improve representation learning (Hjelm et al. 2019; Zhu, Zhang, and Evans 2020) given a dataset. However, for the purpose of generating an adversarial example for a given data sample, the vanilla MINE is not applicable because it only applies to a batch of data samples (so that empirical data distributions can be used for computing MI estimates) but not to single data sample. To bridge this gap, we will propose two MINE-based sampling methods for single data sample in Section 3.1.

3 Methodology

3.1 MINE of Single Data Sample

Given a data sample $x$ and its perturbed sample $x + \delta$, we construct an auxiliary distribution using their random samples or convolution outputs to compute MI via MINE as a similarity measure, which we denote as “per-sample MINE”.

Random Sampling Using compressive sampling (Candès and Wakin 2008), we perform independent Gaussian sampling of a given sample $x$ to obtain a batch of $K$ compressed samples $\{x_k, (x + \delta)_k\}_{k=1}^K$ for computing $I_\theta(x, x + \delta)$ via MINE. We refer the readers to the supplementary material (SuppMat 6.2, 6.3) for more details. We also note that random sampling is agnostic to the underlying machine learning model since it directly applies to the data sample.

Convolution Layer Output When the underlying neural network model uses a convolution layer to process the input
We formalize the objectives for supervised/unsupervised adversarial examples. The unsupervised setting aims to find least similar examples but having smaller training loss, leading to an MINE minimization problem. Both problems can be solved efficiently using our unified MinMax algorithm.

Let \((x, y)\) denote a pair of a data sample \(x\) and its ground-truth label \(y\). The objective of supervised adversarial example is to find a perturbation \(\delta\) to \(x\) such that the MI estimate \(I_\emptyset(x, x + \delta)\) is maximized while the prediction of \(x + \delta\) is different from \(y\) (or being a targeted class \(y' \neq y\)), which is formulated as

\[
\text{Maximize} \quad I_\emptyset(x, x + \delta)
\]

such that \(x + \delta \in [0, 1]^d\), \(\delta \in [-\epsilon, \epsilon]^d\) and \(f_\emptyset(x + \delta) \leq 0\).

The constraint \(x + \delta \in [0, 1]^d\) ensures \(x + \delta\) lies in the (normalized) data space of dimension \(d\), and the constraint \(\delta \in [-\epsilon, \epsilon]^d\) corresponds to the typical bounded \(L_\infty\) perturbation norm. We include this bounded-norm constraint to make direct comparisons to other norm-bounded attacks. One can ignore this constraint by setting \(\epsilon = 1\). Finally, the function \(f_\emptyset^{\text{sup}}(x + \delta)\) is an attack success evaluation function, where \(f_\emptyset^{\text{sup}}(x + \delta) \leq 0\) means \(x + \delta\) is a prediction-evasive adversarial example. For untargeted attack one can use the attack function \(f_\emptyset^{\text{sup}}\) designed in (Carlini and Wagner 2017b), which is \(f_\emptyset^{\text{sup}}(x') = \logit(x')_y - \max_{j: j \neq y} \logit(x')_j + \kappa\), where \(\logit(x')_j\) is the \(j\)-th class output of the logit (soft-max) layer of a neural network, and \(\kappa \geq 0\) is a tunable gap between the original prediction \(\logit(x')_y\) and the top prediction \(\max_{j: j \neq y} \logit(x')_j\) of all classes other than \(y\). Similarly, the attack function for targeted attack with a class label \(y' \neq y\) is \(f_\emptyset^{\text{sup}}(x') = \max_{j: j \neq y'} \logit(x')_j - \logit(x')_{y'} + \kappa\).

**Unsupervised Adversarial Example**  
Many machine learning tasks such as data reconstruction and unsupervised representation learning do not use data labels, which prevents the use of aforementioned supervised attack functions. Here we use an autoencoder \(\Phi(\cdot)\) for data reconstruction to illustrate the unsupervised attack formulation. The design principle can naturally extend to other unsupervised tasks. The autoencoder \(\Phi\) takes a data sample \(x\) as an input and outputs a reconstructed data sample \(\Phi(x)\). Different from the rationale of supervised attack, for unsupervised attack we propose to use MINE to find the least similar perturbed data sample \(x + \delta\) with respect to \(x\) while ensuring the reconstruction loss of \(\Phi(x + \delta)\) is no greater than \(\Phi(x)\) (i.e., the criterion of successful attack for data reconstruction). The unsupervised attack formulation is as follows:

\[
\text{Minimize} \quad I_\emptyset(x, x + \delta)
\]

such that \(x + \delta \in [0, 1]^d\), \(\delta \in [-\epsilon, \epsilon]^d\) and \(f_\emptyset(x + \delta) \leq 0\). The first two constraints regulate the feasible data space and the perturbation range. For the \(L_2\)-norm reconstruction loss, the unsupervised attack function is

\[
f_\emptyset^{\text{unsup}}(x + \delta) = \|x - \Phi(x + \delta)\|_2 - \|x - \Phi(x)\|_2 + \kappa
\]

which means the attack is successful (i.e., \(f_\emptyset^{\text{unsup}}(x + \delta) \leq 0\)) if the reconstruction loss of \(x + \delta\) relative to the original sample \(x\) is smaller than the native reconstruction loss minus a nonnegative margin \(\kappa\). That is, \(\|x - \Phi(x + \delta)\|_2 \leq

---

**Table 2**: Frechet and kernel inception distances (FID/KID) between the generated adversarial examples of 1000 test samples and the training data in CIFAR-10.

<table>
<thead>
<tr>
<th>Per-sample MINE Method</th>
<th>FID</th>
<th>KID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Sampling (10 runs, (K = 96))</td>
<td>339.47 ± 8.07</td>
<td>14.86 ± 1.45</td>
</tr>
<tr>
<td>1st Convolution Layer Output ((K = 96))</td>
<td>344.231</td>
<td>10.78</td>
</tr>
</tbody>
</table>

**Evaluation**  
We use the CIFAR-10 dataset and the same neural network as in Section 4.2 to provide qualitative and quantitative evaluations on the two per-sample MINE methods for image classification. Figure 1 shows their visual comparisons, with the objective of finding the most similar perturbed sample (measured by MINE with the maximal scaled \(L_\infty\) perturbation bound \(\epsilon = 1\)) leading to misclassification. Both random sampling and convolution-based approaches can generate high-similarity prediction-evasive adversarial examples despite of large \(L_\infty\) perturbation.

Table 2 compares the Frechet inception distance (FID) (Heusel et al. 2017) and the kernel inception distance (KID) (Bińkowski et al. 2018) between the generated adversarial examples versus the training data (lower value is better). Both per-sample MINE methods have comparable scores. The convolution-based approach attains lower KID score and is observed to have better visual quality as shown in Figure 1. We also tested the performance using the second convolution layer output but found degraded performance. In this paper we use convolution-based approach whenever applicable and otherwise use random sampling.

### 3.2 MINE-based Attack Formulation

We formalize the objectives for supervised/unsupervised adversarial examples using per-sample MINE. As summarized in Table 1, the supervised setting aims to find most similar examples causing prediction evasion, leading to an MINE maximization problem.

[Figure 1: Visual comparison of MINE-based untargeted supervised adversarial examples (with \(\epsilon = 1\)) on CIFAR-10.]
\[ ||x - \Phi(x)||_2^2 - \kappa. \] In other words, our unsupervised attack formulation aims to find that most dissimilar perturbed sample \( x + \delta \) to \( x \) measured by MINE while having smaller reconstruction loss (in reference to \( x \)) than \( x \). Such UAEs thus relates to generalization errors on low-loss samples because the model is biased toward these unseen samples.

### 3.3 MINE-based Attack Algorithm

Here we propose a unified MinMax algorithm for solving the aforementioned supervised and unsupervised attack formulations, and provide its convergence proof in Section 3.4. For simplicity, we will use \( f_x \) to denote the attack criterion for \( f_x^{\text{sup}} \) or \( f_x^{\text{unsup}} \). Without loss of generality, we will analyze the supervised attack objective of maximizing \( I_{\Theta} \) with constraints. The analysis also holds for the unsupervised case since minimizing \( I_{\Theta} \) is equivalent to maximizing \( I_{\Theta} \), where \( I_{\Theta}' = -I_{\Theta} \). We will also discuss a penalty-based algorithm as a comparative method to our proposed approach.

**MinMax Algorithm (proposed)** We reformulate the attack generation via MINE as the following MinMax optimization problem with simple convex set constraints:

\[
\min_{\delta \in \mathbb{R}^d} \max_{c \in \Theta} \mathcal{F}(\delta, c) = c \cdot f_x^+(x + \delta) - I_{\Theta}(x, x + \delta)
\]

The outer minimization problem finds the best perturbation \( \delta \) with data and perturbation feasibility constraints \( x + \delta \in [0, 1]^d \) and \( \delta \in [-\epsilon, \epsilon]^d \), which are both convex sets with known analytical projection functions. The inner maximization associates a variable \( c \geq 0 \) with the original attack criterion \( f_x(x + \delta) \leq 0 \), where \( c \) is multiplied to the ReLU activation function of \( f_x \), denoted as \( f_x^+(x + \delta) = \text{ReLU}(f_x(x + \delta)) = \max\{f_x(x + \delta), 0\} \). The use of \( f_x^+ \) means when the attack criterion is not met (i.e., \( f_x(x + \delta) > 0 \)), the loss term \( c \cdot f_x(x + \delta) \) will appear in the objective function \( \mathcal{F} \). On the other hand, if the attack criterion is met (i.e., \( f_x(x + \delta) \leq 0 \)), then \( c \cdot f_x(x + \delta) = 0 \) and the objective function \( \mathcal{F} \) only contains the similarity loss term \(-I_{\Theta}(x, x + \delta)\). Therefore, the design of \( f_x^+ \) balances the tradeoff between the two loss terms associated with attack success and MINE-based similarity. We propose to use alternative projected gradient descent between the inner and outer steps to solve the MinMax attack problem, which is summarized in Algorithm 1. The parameters \( \alpha \) and \( \beta \) denote the step sizes of the minimization and maximization steps, respectively. The gradient \( \nabla f_x^+(x + \delta) \) with respect to \( \delta \) is set to be 0 when \( f_x(x + \delta) \leq 0 \). Our MinMax algorithm returns the successful adversarial example \( x + \delta^* \) with the best MINE value \( I_{\Theta}(x, x + \delta^*) \) over \( T \) iterations.

**Penalty-based Algorithm (baseline)** An alternative approach to solving the MINE-based attack formulation is the penalty-based method with the objective:

\[
\min_{\delta \in \mathbb{R}^d} \max_{c \in \Theta} \mathcal{F}(\delta, c) = c \cdot f_x^+(x + \delta) - I_{\Theta}(x, x + \delta)
\]

where \( c \) is a fixed regularization coefficient instead of an optimization variable. Prior arts such as (Carlini and Wagner 2017b) use a binary search strategy for tuning \( c \) and report the best attack results among a set of \( c \) values. In contrast, our MinMax attack algorithm dynamically adjusts the \( c \) value in the inner maximization stage (step 8 in Algorithm 1).

**Algorithm 1: MinMax Attack Algorithm**

1. **Require**: data sample \( x \), attack criterion \( f_x(\cdot) \), step sizes \( \alpha \) and \( \beta \), perturbation bound \( \epsilon \), # of iterations \( T \).
2. **Initialize**: \( \delta_0 = 0 \), \( c_0 = 0 \), \( \delta^* = 0 \), \( I_{\Theta} = -\infty \), \( t = 1 \).
3. **for** \( t \) in \( T \) **iterations**
   4. \( \delta_{t+1} = \delta_t - \alpha \cdot c_t \cdot \nabla f_x^+(x + \delta_t) - \nabla I_{\Theta}(x, x + \delta_t) \)
   5. **Project** \( \delta_{t+1} \) to \([-\epsilon, \epsilon]^d\) via clipping
   6. **Project** \( x + \delta_{t+1} \) to \([0, 1]^d\) via clipping
   7. Compute \( I_{\Theta}(x, x + \delta_{t+1}) \)
   8. Perform \( c_{t+1} = (1 - \frac{\alpha}{\beta c_t}) \cdot c_t + \beta \cdot f_x^+(x + \delta_{t+1}) \)
   9. **Project** \( c_{t+1} \) to \([0, \infty]^d\)
   10. **if** \( f_x(x + \delta_{t+1}) \leq 0 \) and \( I_{\Theta}(x, x + \delta_{t+1}) > I_{\Theta}^* \) **then**
       11. **update** \( \delta^* = \delta_{t+1} \) and \( I_{\Theta}^* = I_{\Theta}(x, x + \delta_{t+1}) \)
12. **Return** \( \delta^*, I_{\Theta}^* \)

Section 4.2, we will show that our MinMax algorithm is more efficient in finding MINE-based adversarial examples than the penalty-based algorithm. The details of the binary search process are given in SuppMat 6.5. Both methods have similar computation complexity involving \( T \) iterations of gradient and MINE computations.

### 3.4 Convergence Proof of MinMax Attack

As a theoretical justification of our proposed MinMax attack algorithm (Algorithm 1), we provide a convergence proof with the following assumptions on the considered problem:

- **A.1**: The feasible set \( \Delta \) for \( \delta \) is compact, and \( f_x \) has (well-defined) gradients and Lipschitz continuity (with respect to \( \delta \)) with constants \( L_f \) and \( I_f \). That is, \( ||f_x^+(x + \delta) - f_x^+(x + \delta')|| \leq L_f ||\delta - \delta'|| \) and \( ||\nabla f_x^+(x + \delta) - \nabla f_x^+(x + \delta')|| \leq L_f ||\delta - \delta'|| \), \( \forall \delta, \delta' \in \Delta \). Moreover, \( I_{\Theta}(x, x + \delta) \) also has gradient Lipschitz continuity with constant \( L_f \).

- **A.2**: The per-sample MINE is \( \eta \)-stable over iterations for the same input, \( I_{\Theta}(x, x + \delta_{t+1}) - I_{\Theta}(x, x + \delta_{t+1}) \leq \eta \).

A.1 holds in general for neural networks since the numerical gradient of ReLU activation can be efficiently computed and the sensitivity (Lipschitz constant) against the input perturbation can be bounded (Weng et al. 2018). The feasible perturbation set \( \Delta \) is compact when the data space is bounded. A.2 holds by following the consistent estimation proof of the native MINE in (Belghazi et al. 2018).

To state our main theoretical result, we first define the proximal gradient of the objective function as \( \mathcal{L}(\delta, c) := [\delta - P_{\Delta} (\delta - \nabla \mathcal{F}(\delta, c)), c - P_{\mathcal{C}} (c + \nabla \mathcal{F}(\delta, c))] \), where \( P_{\mathcal{X}} \) denotes the projection operator on convex set \( \mathcal{X} \), and \( ||\mathcal{L}(\delta, c)|| \) is a commonly used measure for stationarity of the obtained solution. In our case, \( \Delta := \{\delta : x + \delta \in [0, 1]^d \cap \delta \in [-\epsilon, \epsilon]^d\} \) and \( \mathcal{C} := \{c : 0 \leq c \leq \hat{c}\} \), where \( \hat{c} \) can be an arbitrary large value. When \( ||\mathcal{L}(\delta^*, c^*)|| = 0 \), then the point \( (\delta^*, c^*) \) is refereed as a game stationary point of the min-max problem (Razaviyayn et al. 2020). Next, we now present our main theoretical result.

**Theorem 1.** Suppose Assumptions A.1 and A.2 hold and the sequence \( \{\delta_t, c_t, \forall t \geq 1\} \) is generated by the MinMax attack algorithm. For a given small constant
With the proposed MinMax attack algorithm and per-sample
penalization. Please see the supplemental material (SuppMat 6.8).

Proof. Please see the supplemental material (SuppMat 6.8).

Theorem 1 states the rate of convergence of our proposed
MinMax attack algorithm when provided with sufficient sta-
bility of MINE and proper selection of the step sizes. We
also remark that under the assumptions and conditions of
step-sizes, this convergence rate is standard in non-convex
min-max saddle point problems (Lu et al. 2020).

3.5 Data Augmentation Using UAE

With the proposed MinMax attack algorithm and per-sample
MINE for similarity evaluation, we can generate MINE-based
supervised and unsupervised adversarial examples (UAEs).
Section 4 will show novel applications of MINE-based UAEs
as a simple plug-in data augmentation tool to boost the model
performance of several unsupervised machine learning tasks.
We observe significant and consistent performance improve-
ment in data reconstruction (up to 73.5% improvement), rep-
resentation learning (up to 1.39% increase in accuracy), and
contrastive learning (1.58% increase in accuracy). The ob-
served performance gain can be attributed to the fact that our
UAEs correspond to “on-manifold” data samples having low
training loss but are dissimilar to the training data, causing
generalization errors. Therefore, data augmentation and re-
training with UAEs can improve generalization (Stutz, Hein,
and Schiele 2019).

4 Performance Evaluation

In this section, we conduct extensive experiments on a va-
rity of datasets and neural network models to demonstrate
the performance of our proposed MINE-based MinMax ad-
versarial attack algorithm and the utility of its generated
UAEs for data augmentation, where a high attack success
rate using UAEs suggests rich space for data augmentation
to improve model performance. Codes are available at
https://github.com/IBM/UAE.

4.1 Experiment Setup and Datasets

Datasets and Computing Resource We provide a brief sum-
mary of the datasets and computing resource in SuppMat
6.18.

Supervised Adversarial Example Setting Both data sam-
ple and their labels are used in the supervised setting. We
select 1000 test images classified correctly by the pretrained
MNIST and CIFAR-10 deep neural network classifiers used in
(Carlini and Wagner 2017b) and set the confidence gap
parameter $\kappa = 0$ for the designed attack function $f^{sup}_x$ defined
in Section 3.2. The attack success rate (ASR) is the fraction
of the final perturbed samples leading to misclassification.

Unsupervised Adversarial Example Setting Only the train-
ing data samples are used in the unsupervised setting. Their
true labels are used in the post-hoc analysis for evaluating
the quality of the associated unsupervised learning tasks. All

\[ \epsilon' \] and positive constant $\beta$, let $T(\epsilon')$ denote the first
iteration index such that the following inequality is satisfied:

\[ T(\epsilon') := \min \{ t \mid \| L(\delta_t, \alpha_t) \|_2^2 \leq \epsilon', \ t \geq 1 \}. \]

Then, when the step-size and approximation error achieved by Algorithm 1
satisfy $\alpha \sim \eta \sim \sqrt{1/T(\epsilon')}$, there exists some constant $C$
such that $\| L(\delta_{T(\epsilon')}, \alpha_{T(\epsilon')}) \|_2^2 \leq C/\sqrt{T(\epsilon')}$.

Table 3: Comparison between MinMax and penalty-based
algorithms on MNIST and CIFAR-10 datasets in terms of
attack success rate (ASR) and mutual information (MI) value
averaged over 1000 adversarial examples.

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Penalty-based</td>
<td>100%</td>
<td>28.28</td>
</tr>
<tr>
<td>MinMax</td>
<td>100%</td>
<td>51.29</td>
</tr>
</tbody>
</table>

Figure 2: Mean and standard deviation of mutual information
(MI) value versus attack iteration over 1000 samples.

training data are used for generating UAEs individually by
setting $\kappa = 0$. A perturbed data sample is considered as a suc-
cessful attack if its loss (relative to the original sample) is no
greater than the original training loss (see Table 1). For data
augmentation, if a training sample fails to find a successful
attack, we will replicate itself to maintain data balance. The
ASR is measured on the training data, whereas the reported
model performance is evaluated on the test data. The training
performance is provided in SuppMat 6.10.

MinMax Algorithm Parameters We use consistent parame-
ters by setting $\alpha = 0.01, \beta = 0.1,$ and $T = 40$ as the default
values. The vanilla MINE model (Belghazi et al. 2018) is used
in our per-sample MINE implementation.

Models and Codes We defer the summary of the considered
machine learning models to the corresponding sections.

4.2 MinMax v.s. Penalty-based Algorithms

We use the same untargeted supervised attack formulation
and a total of $T = 9000$ iterations to compare our proposed
MinMax algorithm with the penalty-based algorithm using 9
binary search steps on MNIST and CIFAR-10. Table 3 shows
that while both methods can achieve 100% ASR, MinMax
algorithm attains much higher MI values than penalty-based
algorithm. The results show that the MinMax approach is
more efficient in finding MINE-based adversarial examples,
which can be explained by the dynamic update of the coeffi-
cient $c$ in Algorithm 1.

Figure 2 compares the statistics of MI values over attack
iterations. One can find that as iteration count increases, Min-
Max algorithm can continue improving the MI value, whereas
penalty-based algorithm saturates at a lower MI value due
to the use of fixed coefficient $c$ in the attack process. In the
remaining experiments, we will report the results using Min-
Max algorithm due to its efficiency.
We only use the training samples and the associated training data. We suggest not using the \( L_\infty \) similarity regulation while PGD attack only uses \( L_\infty \) norm constraint on the perturbation. The main difference is that MinMax attack uses MINE as an additional similarity regulation while PGD attack only uses \( L_\infty \) norm constraint. We use the concrete autoencoder (Balın, Abid, and Zou 2019) for data augmentation and train the model from scratch on the augmented dataset, and report the resulting reconstruction error on the original test set.

We also compare the performance of our proposed MINE-based UAE (MINE-UAE) with two baselines: (i) \( L_2 \)-UAE that replaces the objective of minimizing \( I_0(x, x + \delta) \) with maximizing the \( L_2 \) reconstruction loss \( \|x - \Phi(x + \delta)\|_2 \) in the MinMax attack algorithm while keeping the same attack success criterion; (ii) Gaussian augmentation (GA) that adds zero-mean Gaussian noise with a diagonal covariance matrix of the same constant \( \sigma^2 \) to the training data.

Table 4 shows the reconstruction loss and the ASR. The improvement of reconstruction error is measured with respect to the reconstruction loss of the original model (i.e., without data augmentation). We find that MINE-UAE can attain much higher ASR than \( L_2 \)-UAE and GA in most cases. More importantly, data augmentation using MINE-UAE achieves consistent and significant reconstruction performance improvement across all models and datasets (up to 56.7% on MNIST and up to 73.5% on SVHN), validating the effectiveness of MINE-UAE for data augmentation. On the other hand, in several cases \( L_2 \)-UAE and GA lead to notable performance degradation. The results suggest that MINE-UAE can be an effective plug-in data augmentation tool for boosting the performance of unsupervised machine learning models.

### 4.4 UAE Improves Data Reconstruction

Data reconstruction using an autoencoder \( \Phi(\cdot) \) that learns to encode and decode the raw data through latent representations is a standard unsupervised learning task. Here we use the default implementation of the following four autoencoders to generate UAEs based on the training data samples of MNIST and SVHN for data augmentation, retrain the model from scratch on the augmented dataset, and report the resulting reconstruction error on the original test set.

Figure 3 presents a visual comparison of MNIST supervised adversarial examples crafted by MinMax attack and the PGD attack with 100 iterations (Madry et al. 2018) given different \( \epsilon \) values governing the \( L_\infty \) perturbation bound. The main difference is that MinMax attack uses MINE as an additional similarity regulation while PGD attack only uses \( L_\infty \) norm constraint. Given the same \( \epsilon \) value, MinMax attack yields adversarial examples with better visual quality. The results validate the importance of MINE as an effective similarity metric. In contrast, PGD attack aims to make full use of the \( L_\infty \) perturbation bound and attempts to modify every data dimension, giving rise to lower-quality adversarial examples. Similar results are observed for adversarially robust models (Madry et al. 2018; Zhang et al. 2019), as shown in SuppMat 6.16.

Moreover, the results also suggest that for MINE-based attacks, the \( L_\infty \) norm constraint on the perturbation is not critical for the resulting visual quality, which can be explained by the fact that MI is a fundamental information-theoretic similarity measure. When performing MINE-based attacks, we suggest not using the \( L_\infty \) norm constraint (by setting \( \epsilon = 1 \)) so that the algorithm can fully leverage the power of MI to find a more diverse set of adversarial examples.

Next, we study three different unsupervised learning tasks. We only use the training samples and the associated training loss to generate UAEs. The post-hoc analysis reports the performance on the test data and the downstream classification accuracy. We report their improved adversarial robustness after data augmentation with MINE-UAEs in SuppMat 6.17.

### 4.5 UAE Improves Representation Learning

The SimCLR algorithm (Chen et al. 2018) is a popular contrastive learning framework for visual representations. It uses self-supervised data modifications to efficiently improve several downstream image classification tasks. We use the default implementation of SimCLR on CIFAR-10 and generate MINE-UAEs using the training data and the defined training loss for SimCLR. Table 5 shows the loss, ASR and the resulting classification accuracy by training a linear head on the learned representations. We find that using MINE-UAE for
Table 4: Comparison of data reconstruction by retraining the autoencoder on UAE-augmented data. The error is the average reconstruction loss of the test set. The improvement is relative to the original model. The attack success rate (ASR) is the fraction of augmented training data having smaller reconstruction loss than the original loss (see Table 1 for definition).

Table 5: Comparison of contrastive loss and the resulting accuracy of MINE-based supervised and unsupervised adversarial examples and establish its convergence guarantees. As a novel application, we show that MINE-based UAEs can be used as a simple yet effective plug-in data augmentation tool and achieve significant performance gains in data reconstruction, representation learning, and contrastive learning.

Acknowledgments
This work was primarily done during Chia-Yi’s visit at IBM Research. Chia-Yi Hsu and Chia-Mu Yu were supported by
References


