CHAPTER 21

Data augmentation for unsupervised machine learning

In addition to studying the failure modes in machine learning models and systems, this chapter introduces how a type of adversarial examples can be used as an efficient data augmentation tool to improve the generalization and robustness for unsupervised machine learning tasks. When using these 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.

21.1 Adversarial examples for unsupervised machine learning models

Despite of 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 *semisupervised* setting by leveraging supervision from a classifier (trained on labeled data) and using the predicted labels on unlabeled data for generating (semisupervised) adversarial examples (Miyato et al., 2018; Zhang et al., 2019b; 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 the study by Hsu et al. (2022) regarding their robustness. The 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 21.1 summarizes the fundamental difference between conventional supervised adversarial examples and our UAEs. Notably, the 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

Adversarial Robustness for Machine Learning
https://doi.org/10.1016/B978-0-12-824020-5.00033-8

information or prediction from other supervised models. The UAEs can be interpreted as "on-manifold" data samples having low training loss but are dissimilar to the training data, causing generalization errors. Therefore data augmentation and retraining with UAEs can improve model generalization (Stutz et al., 2019).

Table 21.1 Illustration of adversarial examples for supervised/unsupervised machine learning tasks. Both settings use a native data sample x as reference. For supervised setting, adversarial examples refer to *similar* samples of x causing inconsistent predictions. For unsupervised setting, adversarial examples refer to *dissimilar* samples yielding smaller loss than x, relating to generalization errors on low-loss samples.

(I) *Mathematical notation*	
$M^{\text{sup}}/M^{\text{unsup}}$: trained supervised/unsupervised machine learning models x/x_{adv}: original/adversarial data sample $\ell_x^{\text{sup}}/\ell_x^{\text{unsup}}$: supervised/unsupervised loss function in reference to x	
(II) *Supervised tasks* (e.g., classification)	(III) *Unsupervised tasks* (e.g., data reconstruction, contrastive learning)
x_{adv} is **similar** to x, but $M^{\text{sup}}(x_{\text{adv}}) \neq M^{\text{sup}}(x)$	x_{adv} is **dissimilar** to x, but $\ell_x^{\text{unsup}}(x_{\text{adv}}\|M^{\text{unsup}}) \leq \ell_x^{\text{unsup}}(x\|M^{\text{unsup}})$

Hsu et al. (2022) propose a 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. 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 ($\mathbb{P}_{\mathbb{X}}$, $\mathbb{P}_{\mathbb{Z}}$, and $\mathbb{P}_{\mathbb{XZ}}$). For efficient computation, the mutual information neural estimator (MINE) with consistency guarantees is proposed by Belghazi et al. (2018). Specifically, MINE aims to maximize the lower bound of the exact MI using a model parameterized by a neural network θ defined as $I_\Theta(X, Z) \leq I(X, Z)$, where Θ is the space of feasible parameters of a neural network, and $I_\Theta(X, Z)$ is the neural information quantity defined as $I_\Theta(X, Z) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Z}[e^{T_\theta}])$. The function T_θ is parameterized by a neural network θ 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 $\mathbb{P}_{\mathbb{XZ}}$ and $\mathbb{P}_{\mathbb{X}} \otimes \mathbb{P}_{\mathbb{Z}}$ (the product of marginals). MINE has been successfully

applied to improve representation learning (Hjelm et al., 2019; Zhu et al., 2020a) given a dataset. However, for 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.

Per-sample MINE. Given a data sample x and its perturbed sample $x+\delta$, we can construct an auxiliary distribution using their random samples or convolution outputs to compute MI via MINE as a similarity measure, denoted 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 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 data (which is an almost granted setting for image data), we propose to use the output of the first convolution layer of a data input, denoted by *conv*, to obtain K feature maps $\{conv(x)_k, conv(x+\delta)_k\}_{k=1}^K$ for computing $I_\Theta(x, x+\delta)$.

Unified attack formulation. We formalize the objectives for supervised and unsupervised adversarial examples using per-sample MINE. As summarized in Table 21.1, the supervised setting aims to find *most similar* examples causing prediction evasion, leading to an MINE *maximization* problem. 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 a unified MinMax algorithm (Algorithm 7).

Supervised adversarial example. 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 δ to x such that the MI estimate $I_\Theta(x, x+\delta)$ is maximized while the prediction of $x+\delta$ is different from y (or being a targeted class $y' \neq y$), which is formulated as

$$\begin{aligned} &\underset{\delta}{\text{Maximize}} \quad I_\Theta(x, x+\delta) \\ &\textit{such that } x+\delta \in [0,1]^d,\ \delta \in [-\epsilon, \epsilon]^d \text{ and } f_x(x+\delta) \leq 0. \end{aligned} \tag{21.1}$$

The constraint $x+\delta \in [0,1]^d$ ensures that $x+\delta$ lies in the (normalized) data space of dimension d, and the constraint $\delta \in [-\epsilon, \epsilon]^d$ corresponds to the typical bounded L_∞ perturbation norm. We include this bounded-norm

constraint to make direct comparisons to other norm-bounded attacks. We can ignore this constraint by setting $\epsilon = 1$. Finally, the function $f_x^{\text{sup}}(x+\delta)$ is an attack success evaluation function, where $f_x^{\text{sup}}(x+\delta) \leq 0$ means that $x+\delta$ is a prediction-evasive adversarial example. For untargeted attack, we can use the attack function f_x^{sup} designed by Carlini and Wagner (2017b), which is $f_x^{\text{sup}}(x') = \text{logit}(x')_y - \max_{j:j\neq y} \text{logit}(x')_j + \kappa$, where $\text{logit}(x')_j$ is the jth class output of the logit (pre-softmax) layer of a neural network, and $\kappa \geq 0$ is a tunable gap between the original prediction $\text{logit}(x')_y$ and the top prediction $\max_{j:j\neq y} \text{logit}(x')_j$ of all classes other than y. Similarly, the attack function for targeted attack with a class label $y' \neq y$ is $f_x^{\text{sup}}(x') = \max_{j:j\neq y'} \text{logit}(x')_j - \text{logit}(x')_{y'} + \kappa$.

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 Φ for data reconstruction to illustrate the unsupervised attack formulation. The design principle can naturally extend to other unsupervised tasks. The autoencoder Φ 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 that 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:

$$\begin{aligned} &\underset{\delta}{\text{Minimize}} \quad I_\Theta(x, x+\delta) \\ &\textit{such that } x+\delta \in [0,1]^d\,,\ \delta \in [-\epsilon, \epsilon]^d \text{ and } f_x(x+\delta) \leq 0. \end{aligned} \tag{21.2}$$

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

$$f_x^{\text{unsup}}(x+\delta) = \|x - \Phi(x+\delta)\|_2 - \|x - \Phi(x)\|_2 + \kappa, \tag{21.3}$$

which means that the attack is successful (i.e., $f_x^{\text{unsup}}(x+\delta) \leq 0$) if the reconstruction loss of $x+\delta$ relative to the original sample x is smaller than the native reconstruction loss minus a nonnegative margin κ, that is, $\|x - \Phi(x+\delta)\|_2 \leq \|x - \Phi(x)\|_2 - \kappa$. 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 relate to generalization errors on low-loss samples because the model is biased toward these unseen samples.

MINE-based MinMax algorithm. Here we introduce a unified MinMax algorithm for solving the aforementioned supervised and unsupervised attack formulations. Its algorithmic convergence proof is given in (Hsu et al., 2022). For simplicity, we will use f_x to denote the attack criterion for f_x^{sup} or f_x^{unsup}. Without loss of generality, we will analyze the supervised attack objective of maximizing I_Θ with constraints. The analysis also holds for the unsupervised case since minimizing I_Θ is equivalent to maximizing I'_Θ, where $I'_\Theta = -I_\Theta$.

The attack generation via MINE can be reformulated as the following MinMax optimization problem with simple convex set constraints:

$$\underset{\delta: x+\delta\in[0,1]^d,\ \delta\in[-\epsilon,\epsilon]^d}{\text{Min}} \quad \underset{c\geq 0}{\text{Max}} \quad J(\delta, c) \triangleq c \cdot f_x^+(x+\delta) - I_\Theta(x, x+\delta). \tag{21.4}$$

The outer minimization problem finds the best perturbation δ 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 by 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 that 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 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 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. Hsu et al. (2022) propose to use alternative projected gradient descent between the inner and outer steps to solve the MinMax attack problem, which is summarized in Algorithm 7. The parameters α and β denote the step sizes of the minimization and maximization steps, respectively. The gradient $\nabla f_x^+(x+\delta)$ with respect to δ is set to be 0 when $f_x(x+\delta) \leq 0$. The MinMax algorithm returns the successful adversarial example $x+\delta^*$ with the best MINE value $I_\Theta^*(x, x+\delta^*)$ over T iterations.

21.2 Empirical comparison

With the MinMax attack algorithm and per-sample MINE for similarity evaluation, we can generate MINE-based unsupervised adversarial exam-

Algorithm 7 MINE-based MinMax attack algorithm.

Require: data sample x, attack criterion f_x, step sizes α and β, perturbation bound ϵ, # of iterations T
Initialize $\delta_0 = 0$, $c_0 = 0$, $\delta^* = \text{null}$, $I^*_\Theta = -\infty$, $t = 1$
for t in T iterations **do**
    $\delta_{t+1} = \delta_t - \alpha \cdot (c \cdot \nabla f^+_x(x + \delta_t) - \nabla I_\Theta(x, x + \delta_t))$
    Project δ_{t+1} to $[-\epsilon, \epsilon]$ via clipping
    Project $x + \delta_{t+1}$ to $[0, 1]$ via clipping
    Compute $I_\Theta(x, x + \delta_{t+1})$
    Perform $c_{t+1} = (1 - \frac{\beta}{t^{1/4}}) \cdot c_t + \beta \cdot f^+_x(x + \delta_{t+1})$
    Project c_{t+1} to $[0, \infty]$
    if $f_x(x + \delta_{t+1}) \leq 0$ and $I_\Theta(x, x + \delta_{t+1}) > I^*_\Theta$ **then**
        update $\delta^* = \delta_{t+1}$ and $I^*_\Theta = I_\Theta(x, x + \delta_{t+1})$
    end if
end for
Return δ^*, I^*_Θ

ples (MINE-UAEs). In what follows, we show novel applications of MINE-UAEs as a simple plug-in data augmentation tool to boost the model performance of several unsupervised machine learning tasks. We provide a brief summary of the datasets:

• *MNIST* consists of grayscale images of hand-written digits. The numbers of training/test samples are 60K/10K.

• *SVHN* is a color image dataset set of house numbers extracted from Google Street View images. The number of training/test samples are 73257/26302.

• *Fashion MNIST* contains grayscale images of 10 clothing items. The numbers of training/test samples are 60K/10K.

• *Isolet* consists of preprocessed speech data of people speaking the name of each letter of the English alphabet. The numbers of training/test samples are 6238/1559.

• *Coil-20* contains grayscale images of 20 multiviewed objects. The numbers of training/test samples are 1152/288.

• *Mice protein* consists of the expression levels (features) of 77 protein modifications in the nuclear fraction of cortex. The numbers of training/test samples are 864/216.

• *Human activity recognition* consists of sensor data collected from a smartphone for various human activities. The numbers of training/test samples are 4252/1492.

Data reconstruction. Data reconstruction using an autoencoder Φ 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. All autoencoders use the L_2 reconstruction loss defined as $\|x - \Phi(x)\|_2$. The four autoencoders are summarized below.

• *Dense autoencoder (Cavallari et al., 2018).* The encoder and decoder have 1 dense layer separately, and the latent dimension is 128/256 for MNIST/SVHN.

• *Sparse autoencoder.* It has a sparsity enforcer (L_1 penalty on the training loss) that directs a network with a single hidden layer to learn the latent representations minimizing the error in reproducing the input while limiting the number of code words for reconstruction. We use the same architecture as Dense Autoencoder for MNIST and SVHN.

• *Convolutional autoencoder.*[1] The encoder uses convolution+relu+pooling layers. The decoder has reversed layer order with the pooling layer replaced by an upsampling layer.

• *Adversarial autoencoder (Makhzani et al., 2016).* It is composed of an encoder, a decoder, and a discriminator. The rationale is to force the distribution of the encoded values to be similar to the prior data distribution.

We also compare the performance of our proposed MINE-based UAE (MINE-UAE) with two baselines: (i) *L_2-UAE,* which replaces the objective of minimizing $I_\Theta(x, x+\delta)$ 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), which zero-mean Gaussian noise with a diagonal covariance matrix of the same constant σ^2 to the training data.

Table 21.2 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

[1] https://github.com/shibuiwilliam/Keras_Autoencoder.

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.

Table 21.3 demonstrates that UAEs can further improve data reconstruction when the original model already involves conventional augmented training data such as flip, rotation, and Gaussian noise.

Representation learning. The concrete autoencoder (Balın et al., 2019) is an unsupervised feature selection method, which recognizes a subset of the most informative features through an additional *concrete select layer* with M nodes in the encoder for data reconstruction. We apply MINE-UAE for data augmentation and use the same post-hoc classification evaluation procedure as in (Balın et al., 2019).

The six datasets and the resulting classification accuracy are reported in Table 21.4. We select $M = 50$ features for every dataset except for Mice Protein (we set $M = 10$) owing to its small data dimension. MINE-UAE can attain up to 11% improvement for data reconstruction and up to 1.39% increase in accuracy among five out of six datasets, corroborating the utility of MINE-UAE in representation learning and feature selection. The exception is Coil-20. A closer inspection shows that MINE-UAE has low ASR (<10%) for Coil-20 and the training loss after data augmentation is significantly higher than the original training loss. Therefore we conclude that the degraded performance in Coil-20 after data augmentation is likely due to the limitation of feature selection protocol and the model learning capacity.

Contrastive learning. The SimCLR algorithm (Chen et al., 2018e) 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 21.5 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 additional data augmentation and model retraining can yield 7.8% improvement in contrastive loss and 1.58% increase in classification accuracy. Comparing to (Ho and

Table 21.2 Comparison of data reconstruction by retraining the autoencoder on UAE-augmented data. The error is the average L_2 reconstruction loss of the test set. The improvement (in green (light gray in print version)/red (dark gray in print version)) 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 21.1 for definition).

MNIST

	Reconstruction Error (test set)					ASR (training set)			
Autoencoder	**Original**	**MINE-UAE**	**L_2-UAE**	**GA ($\sigma = 0.01$)**	**GA ($\sigma = 10^{-3}$)**	**MINE-UAE**	**L_2-UAE**	**GA ($\sigma = 0.01$)**	**GA ($\sigma = 10^{-3}$)**
Sparse	0.00561	**0.00243** (↑ 56.7%)	0.00348 (↑ 38.0%)	0.00280±2.60e-05 (↑ 50.1%)	0.00280±3.71e-05 (↑ 50.1%)	100%	99.18%	54.10%	63.95%
Dense	0.00258	**0.00228** (↑ 11.6%)	0.00286 (↓ 6.0%)	0.00244±0.00014 (↑ 5.4%)	0.00238±0.00012 (↑ 7.8%)	92.99%	99.94%	48.53%	58.47%
Convolutional	0.00294	**0.00256** (↑ 12.9%)	0.00364 (↓ 23.8%)	0.00301±0.00011 (↓ 2.4%)	0.00304±0.00015 (↓ 3.4%)	99.86%	99.61%	68.71%	99.61%
Adversarial	0.04785	**0.04581** (↑ 4.3%)	0.06098 (↓ 27.4%)	0.05793±0.00501(↓ 21%)	0.05544±0.00567 (↓ 15.86%)	98.46%	43.54%	99.79%	99.83%
SVHN									
Sparse	0.00887	**0.00235** (↑ 73.5%)	0.00315 (↑ 64.5%)	0.00301±0.00137 (↑ 66.1%)	0.00293±0.00078 (↑ 67.4%)	100%	72.16%	72.42%	79.92%
Dense	0.00659	**0.00421** (↑ 36.1%)	0.00550 (↑ 16.5%)	0.00858±0.00232 (↓ 30.2%)	0.00860±0.00190 (↓ 30.5%)	99.99%	82.65%	92.3%	93.92%
Convolutional	0.00128	**0.00095** (↑ 25.8%)	0.00121 (↑ 5.5%)	0.00098 ± 3.77e-05 (↑ 25.4%)	0.00104±7.41e-05 (↑ 18.8%)	100%	56%	96.40%	99.24%
Adversarial	0.00173	**0.00129** (↑ 25.4%)	0.00181 (↓ 27.4%)	0.00161±0.00061 (↑ 6.9%)	0.00130±0.00037 (↑ 24.9%)	94.82%	58.98%	97.31%	99.85%

Table 21.3 Performance of data reconstruction when retraining with MINE-UAE and additional augmented training data.

SVNH – Convolutional AE		
Augmentation	**Aug. (test set)**	**Aug.+MINE-UAE (test set)**
Flip + Rotation	0.00285	**0.00107** (↑ 62.46%)
Gaussian noise ($\sigma = 0.01$)	0.00107	**0.00095** (↑ 11.21%)
Flip + Rotation + Gaussian noise	0.00307	**0.00099** (↑ 67.75%)

Table 21.4 Performance of representation learning by the concrete autoencoder and the resulting classification accuracy.

	Reconstruction Error (test set)		**Accuracy (test set)**		**ASR**
Dataset	**Original**	**MINE-UAE**	**Original**	**MINE-UAE**	**MINE-UAE**
MNIST	0.01170	**0.01142** (↑ 2.4%)	94.97%	95.41%	99.98%
Fashion MMIST	0.01307	**0.01254** (↑ 4.1%)	84.92%	85.24%	99.99%
Isolet	0.01200	**0.01159** (↑ 3.4%)	81.98%	82.93%	100%
Coil-20	**0.00693**	0.01374 (↓ 98.3%)	98.96%	96.88%	9.21%
Mice Protein	0.00651	**0.00611** (↑ 6.1%)	89.81%	91.2%	40.24%
Activity	0.00337	**0.00300** (↑ 11.0%)	83.38%	84.45%	96.52%

Table 21.5 Comparison of contrastive loss and the resulting accuracy on CIFAR-10 using SimCLR (Chen et al., 2018e) (ResNet-18 with batch size = 512). The attack success rate (ASR) is the fraction of augmented training data having smaller contrastive loss than original loss. For CLAE (Ho and Vasconcelos, 2020), we use the reported accuracy improvement (it shows negative gain in our implementation), though its base SimCLR model only has 83.27% test accuracy.

CIFAR-10			
Model	**Loss (test set)**	**Accuracy (test set)**	**ASR**
Original	0.29010	91.30%	–
MINE-UAE (Hsu et al., 2022)	**0.26755** (↑ 7.8%)	**+1.58%**	100%
CLAE (Ho and Vasconcelos, 2020)	–	+0.05%	–

Vasconcelos, 2020) using adversarial examples to improve SimCLR (named CLAE), the accuracy increase of MINE-UAE is 30 times higher. Moreover, MINE-UAE data augmentation also significantly improves adversarial robustness.