Are Labels Required for Improving Adversarial Robustness?

Jonathan Uesato* **Jean-Baptiste Alayrac*** **Po-Sen Huang***

Robert Stanforth **Alhussein Fawzi** **Pushmeet Kohli**

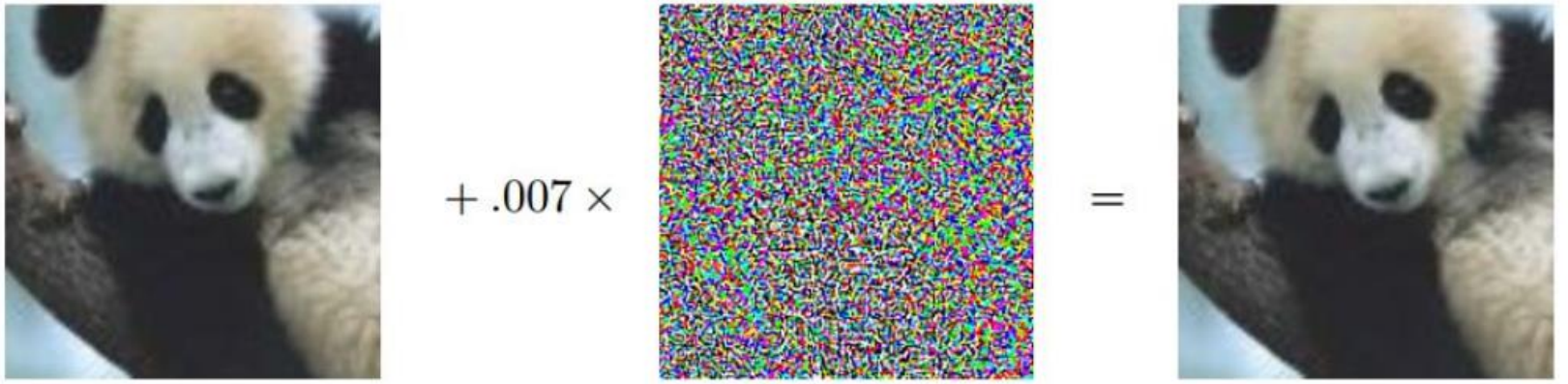
DeepMind
{juesato,jalayrac,posenhuang}@google.com

NIPS-2019

Review

Adversarial examples:

An adversarial example is an instance with small, intentional feature perturbations that cause a machine learning model to make a false prediction.



Review

Adversarial training：

$$\min_{w} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{data}}} \left[\max_{\|\delta\|\leq\delta_{\max}} \ell\big(f(x+\delta;w),y\big) \right]$$

Training with an adversarial objective function based on FGSM：

$$\tilde{J}(\boldsymbol{\theta},\boldsymbol{x},y) = \alpha J(\boldsymbol{\theta},\boldsymbol{x},y) + (1-\alpha) J(\boldsymbol{\theta},\boldsymbol{x}+\epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta},\boldsymbol{x},y)\right))$$

Motivation

Training models to be invariant to adversarial perturbations requires substantially larger datasets than those required for the standard classification task.

This result is a key hurdle in the development and deployment of robust machine learning models in many real world applications where labeled data is expensive.

We ask a simple question: is more labeled data necessary, or is unsupervised data sufficient?

Adversarially Robust Generalization Requires More Data

Method

Notations:

Labeled training set: $\mathcal{S}_n = \{(x_i, y_i)\}_{1 \leq i \leq n}$ Where: $(x_i, y_i) \sim P(X, Y)$

Unlabeled training set: $\mathcal{U}_m = \{x_i\}_{1 \leq i \leq m}$ Where: $x_i \sim P(X)$

Neural network: $f_\theta(x) = \arg\max_{y \in \mathcal{Y}} p_\theta(y|x)$

Method

Evaluation of Adversarial Robustness:

Natural risk:

$$\mathcal{L}_{nat}(\theta) = \mathbb{E}_{(x,y)\sim P(X,Y)}\, \ell(y, f_\theta(x))$$

Adversarial risk:

$$\mathcal{L}_{g}(\theta) = \mathbb{E}_{P(X,Y)}\, \ell(f_\theta(x'), y)$$

Where:

$$x' = g(x, y, \theta)$$

Method

Strategy 1: Unsupervised Adversarial Training with Online Targets (UAT-OT)

$$\mathcal{L}^{OT}_{\text{unsup}}(\theta) = \mathop{\mathbb{E}}_{x\sim P(X)} \sup_{x'\in\mathcal{N}_\epsilon(x)} \mathcal{D}(p_{\hat{\theta}}(.|x), p_\theta(.|x'))$$

Strategy 2: Unsupervised Adversarial Training with Fixed Targets (UAT-FT)

$$\mathcal{L}^{FT}_{\text{unsup}}(\theta) = \mathop{\mathbb{E}}_{x\sim P(X)} \sup_{x'\in\mathcal{N}_\epsilon(x)} \texttt{xent}(\hat{y}(x), p_\theta(.|x'))$$

Overall training

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{sup}}(\theta) + \lambda\mathcal{L}_{\text{unsup}}(\theta)$$

The unsupervised loss can be either $\mathcal{L}^{OT}_{\text{unsup}}$ (UAT-OT), $\mathcal{L}^{\text{FT}}_{\text{unsup}}$ (UAT-FT) or both (UAT++).

Method

Algorithm 1 UAT-OT update

Input: Weight hyperparameter λ, batch sizes b_s and b_u
Sample b_s labeled examples $(\boldsymbol{x}_s, \boldsymbol{y}_s) \sim \mathcal{S}_n$ and b_u unlabeled examples $(\boldsymbol{x}_u, \boldsymbol{y}_u) \sim \mathcal{U}_m$
Compute loss $L = \hat{\mathcal{L}}^{adv}(\boldsymbol{x}_s, \boldsymbol{y}_s; \theta) + \lambda(\frac{b_s}{b_u})\hat{\mathcal{L}}^{OT}(\boldsymbol{x}_u, \boldsymbol{y}_u; \theta)$
Update with gradient $g = \nabla_\theta L$

$$\hat{\mathcal{L}}^{adv}(\boldsymbol{x}, \boldsymbol{y}; \theta) = \frac{1}{|(\boldsymbol{x}, \boldsymbol{y})|} \sum_{i=1}^{|(\boldsymbol{x}, \boldsymbol{y})|} \sup_{x_i' \in N_\epsilon(\boldsymbol{x}_i)} \texttt{xent}(\boldsymbol{y}_i, p_\theta(\cdot|x_i'))$$

$$\hat{\mathcal{L}}^{OT}(\boldsymbol{x}, \boldsymbol{y}; \theta) = \frac{1}{|(\boldsymbol{x}, \boldsymbol{y})|} \sum_{i=1}^{|(\boldsymbol{x}, \boldsymbol{y})|} \sup_{x_i' \in N_\epsilon(\boldsymbol{x}_i)} \mathcal{D}(p_{\hat{\theta}}(.|\boldsymbol{x}_i), p_\theta(.|x_i'))$$

Method

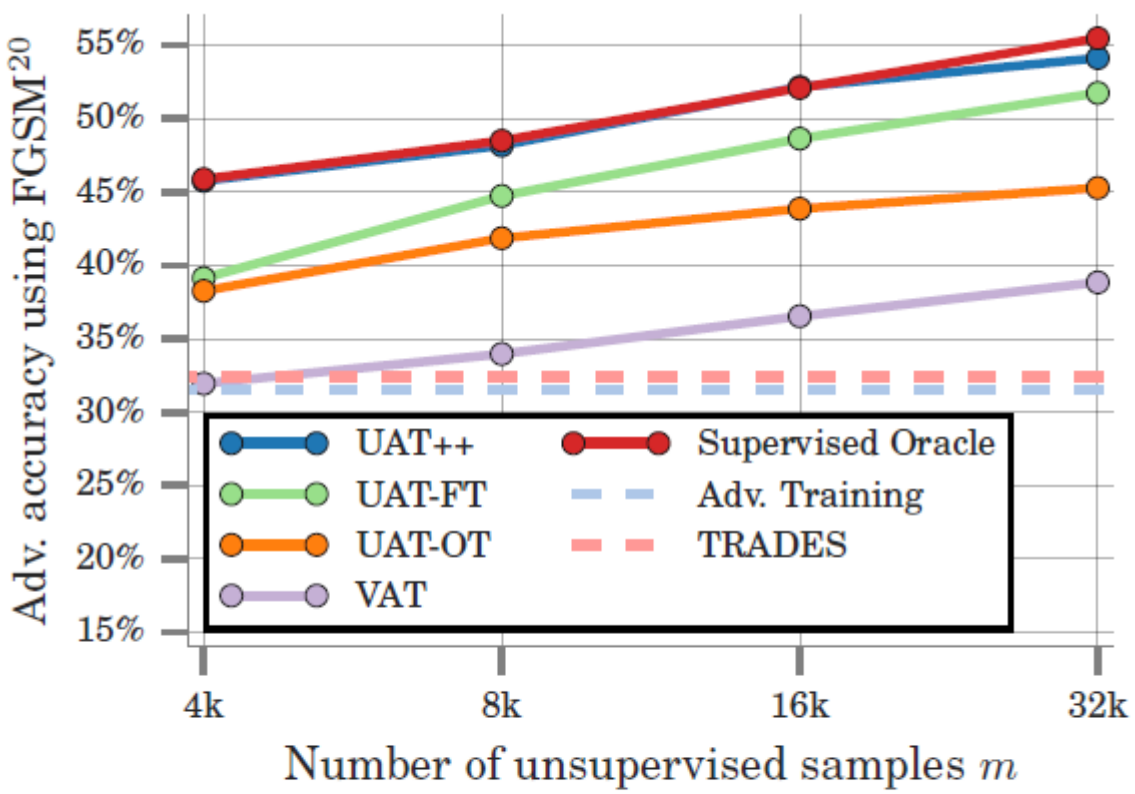
Algorithm 2 UAT-FT update

Input: Batch sizes b_s and b_u
Sample b_s labeled examples $(\boldsymbol{x}_s, \boldsymbol{y}_s) \sim \mathcal{S}_n$ and b_u unlabeled examples $(\boldsymbol{x}_u, \boldsymbol{y}_u) \sim \mathcal{U}_m$
Merge $\boldsymbol{x} = [\boldsymbol{x}_s; \boldsymbol{x}_u]$; $\boldsymbol{y} = [\boldsymbol{y}_s; \boldsymbol{y}_u]$
Compute loss $L = \hat{\mathcal{L}}^{adv}(\boldsymbol{x}, \boldsymbol{y}; \theta)$
Update with gradient $g = \nabla_\theta L$

Algorithm 3 UAT++ update

Input: Weight hyperparameter λ, batch size b_s and b_u
Sample b_s labeled examples $(\boldsymbol{x}_s, \boldsymbol{y}_s) \sim \mathcal{S}_n$ and b_u unlabeled examples $(\boldsymbol{x}_u, \boldsymbol{y}_u) \sim \mathcal{U}_m$
Merge $\boldsymbol{x} = [\boldsymbol{x}_s; \boldsymbol{x}_u]$; $\boldsymbol{y} = [\boldsymbol{y}_s; \boldsymbol{y}_u]$
Compute loss $L = \hat{\mathcal{L}}^{adv}(\boldsymbol{x}, \boldsymbol{y}; \theta) + \lambda \hat{\mathcal{L}}^{OT}(\boldsymbol{x}, \boldsymbol{y}; \theta)$
Update with gradient $g = \nabla_\theta L$

Experiments : Adversarial robustness with few labels



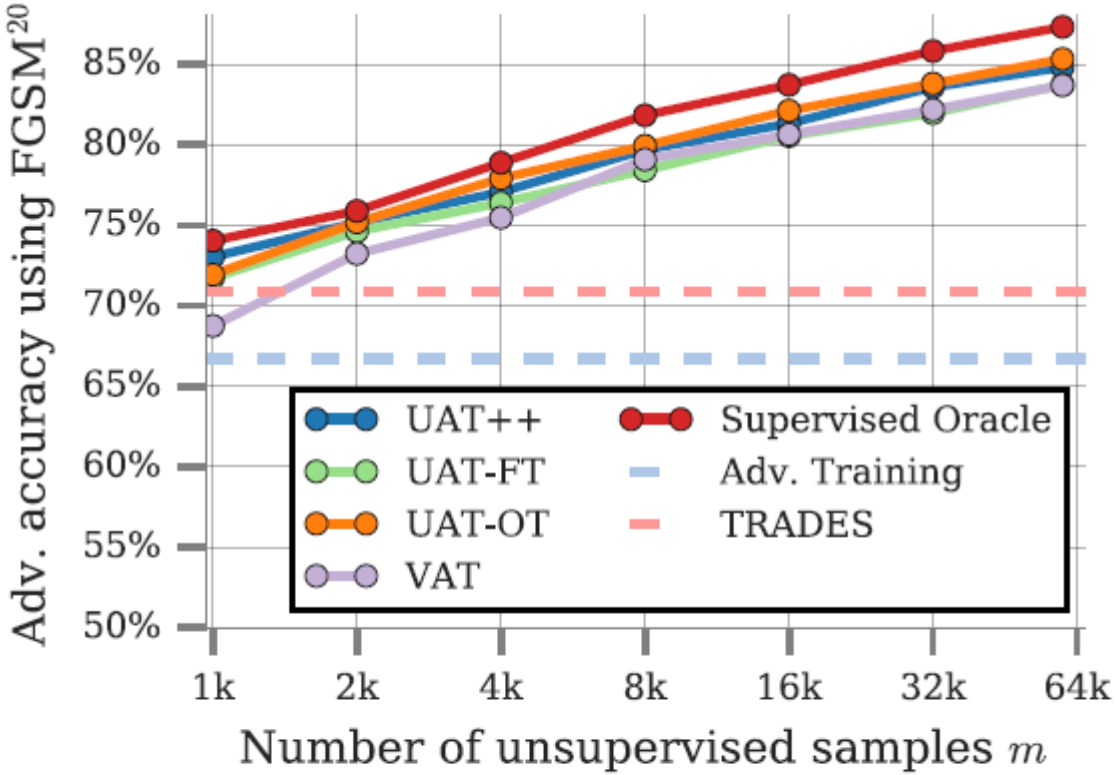
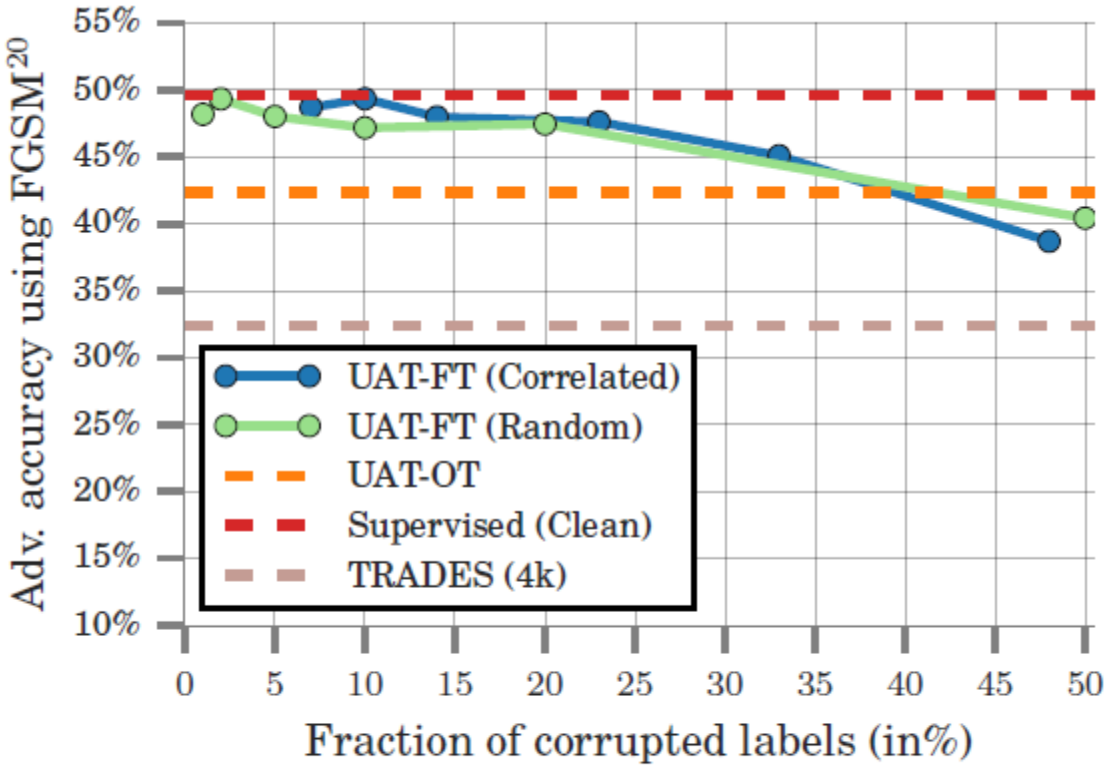


Figure 1: Comparison of labeled data and unsupervised data for improving adversarial generalization on CIFAR-10 (**left,a**) and SVHN (**right,b**)

Experiments : Label noise analysis



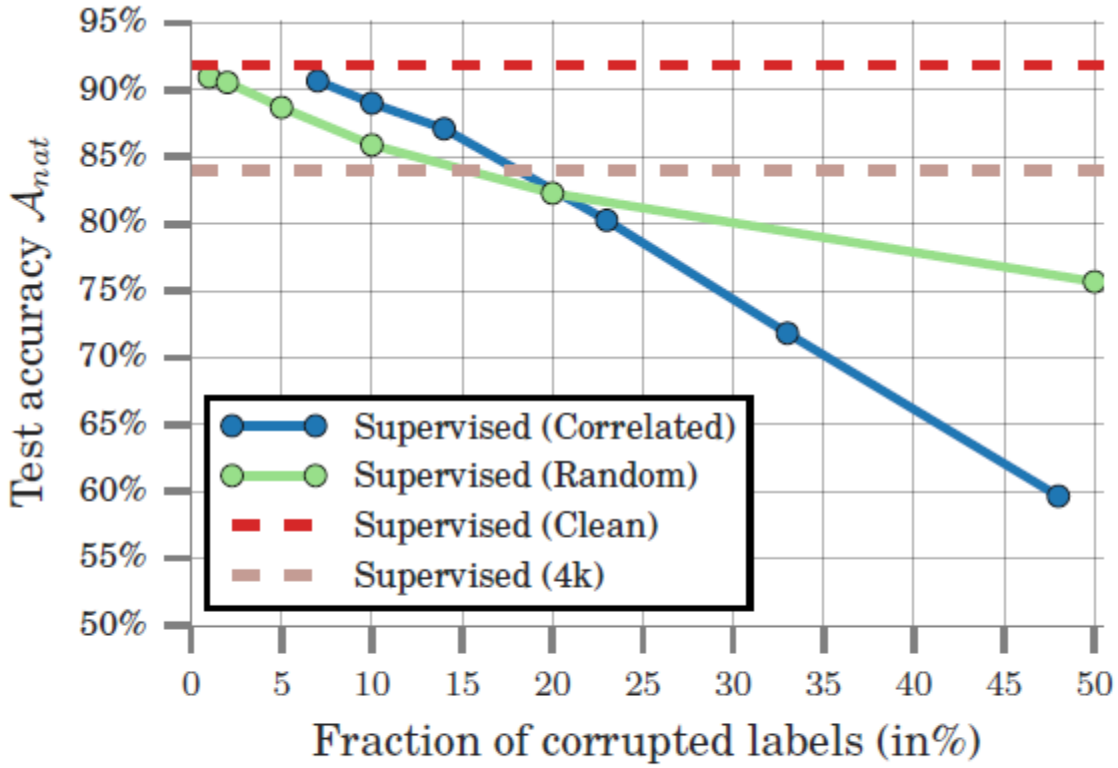


Figure 2: Effects of label noise on adversarial **(left, a)** and natural **(right, b)** accuracies, on CIFAR-10

Experiments : Unsupervised data with distribution shift

80 Million Tiny Images dataset (80m) :
a large dataset obtained by web queries for 75,062 words.

80m@100K dataset : top 10k images per class

80m@200K dataset : top 20k images per class

80m@500K dataset : top 50k images per class

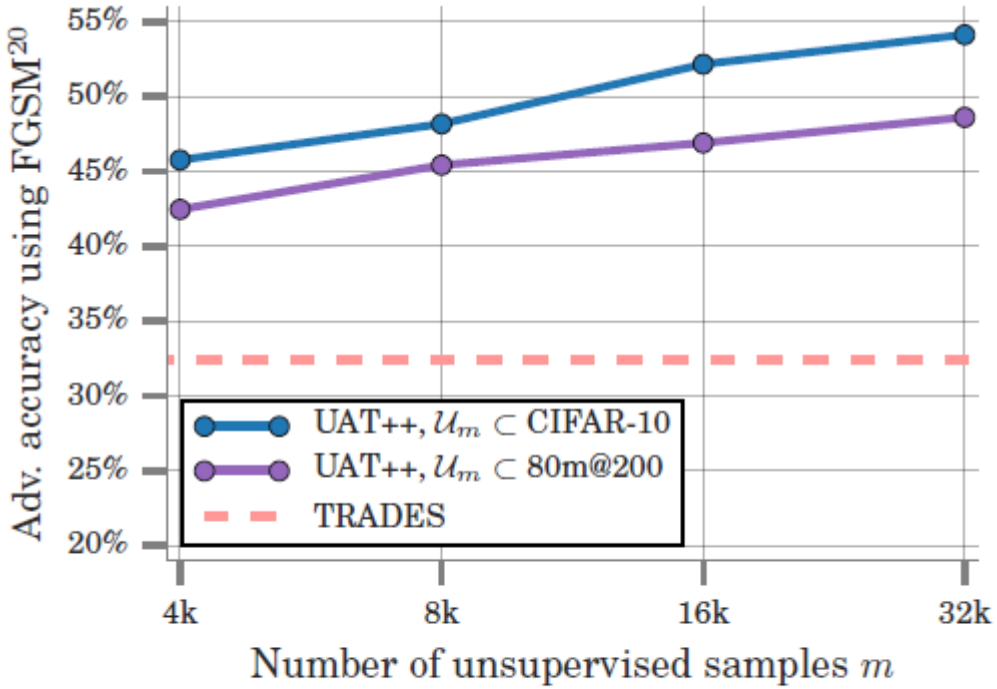


Figure 3: Distribution shift on CIFAR-10

Experiments : Unsupervised data with distribution shift

Method	Sup. Data	Unsup. Data	Network	$\mathcal{A}_{nat}$	$\mathcal{A}_{FGSM^{20}}$	$\mathcal{A}_{MultiTar.}$
[48]	CIFAR-10	✗	-	27.07%	23.54%	-
AT [30]	CIFAR-10	✗	WRN-28	87.30%	47.04%	44.54%
[55]	CIFAR-10	✗	-	94.64%	0.15%	-
[26]	CIFAR-10	✗	-	85.25%	45.89%	-
[21]	ImageNet + CIFAR-10	✗	WRN-28	87.1%	57.40%	$\leq$52.9%*
AT-Reimpl. [30]	CIFAR-10	✗	WRN-34	87.08%	52.93%	47.10%
TRADES [54]	CIFAR-10	✗	WRN-34	84.92%	57.11%	52.58%
UAT++	CIFAR-10	80m@100K	WRN-34	86.04%	59.41%	52.64%
UAT++	CIFAR-10	80m@200K	WRN-34	85.85%	**62.18%**	**53.35%**
UAT++	CIFAR-10	80m@500K	WRN-34	78.34%	58.04%	48.99%
UAT++	CIFAR-10	80m@200K	WRN-70	86.75%	62.89%	55.04%
UAT++	CIFAR-10	80m@200K	WRN-106	86.46%	**63.65%**	**56.30%**

Table 1: Experimental results using 80m Tiny Images dataset (as a unsupervised data) and CIFAR-10 (as supervised data), where $\mathcal{A}_{nat}$ represents the original test accuracy, $\mathcal{A}_{FGSM^{20}}$ represents the adversarial accuracy under 20 step FGSM, and $\mathcal{A}_{MultiTar.}$ represents the adversarial accuracy under the strong `MultiTargeted` attack. WRN-k denotes the Wide-ResNet with depth k. '*' indicates it is from [21] using 100 PGD steps with 1000 random restarts, an attack that we have found to be weaker than the `MultiTargeted` attack.