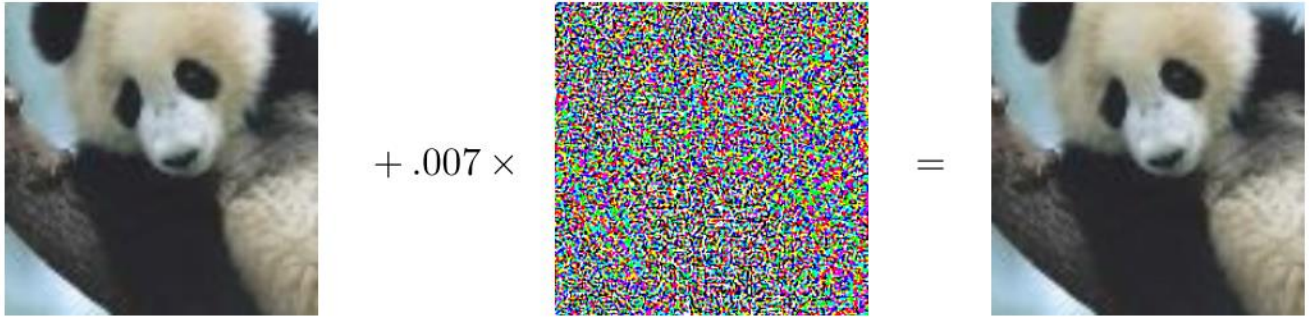
# Learning with Multiplicative Perturbations

Xiulong Yang & Shihao Ji
Georgia State University

# Background

- Adversarial Examples:



$+ .007 \times$ $=$

$\boldsymbol{x}$

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Perturbations are small, imperceptible to human.

Goodfellow et al. 2014

# Adversarial Training

- Improve robustness of DNNs against adversarial examples

AT:

$$\mathcal{L}_{\mathrm{AT}}(\boldsymbol{x}_l, y_l, \boldsymbol{r}_{\mathrm{adv}}, \boldsymbol{\theta}) = D\left[h(y_l|\boldsymbol{x}_l), p(y|\boldsymbol{x}_l + \boldsymbol{r}_{\mathrm{adv}}, \boldsymbol{\theta})\right]$$
$$\text{with } \boldsymbol{r}_{\mathrm{adv}} = \underset{\boldsymbol{r};\|\boldsymbol{r}\|\leq\epsilon}{\arg\max} D\left[h\left(y_l|\boldsymbol{x}_l\right), p\left(y|\boldsymbol{x}_l + \boldsymbol{r}, \boldsymbol{\theta}\right)\right],$$

[Goodfellow et al. 2014]

VAT:

$$\mathcal{L}_{\mathrm{VAT}}(\boldsymbol{x}_*, \boldsymbol{r}_{\mathrm{adv}}, \boldsymbol{\theta}) = D\left[p(y|\boldsymbol{x}_*, \boldsymbol{\theta}), p(y|\boldsymbol{x}_* + \boldsymbol{r}_{\mathrm{adv}}, \boldsymbol{\theta})\right]$$
$$\text{with } \boldsymbol{r}_{\mathrm{adv}} = \underset{\boldsymbol{r};\|\boldsymbol{r}\|_2\leq\epsilon}{\arg\max} D\left[p(y|\boldsymbol{x}_*, \boldsymbol{\theta}), p(y|\boldsymbol{x}_* + \boldsymbol{r}, \boldsymbol{\theta})\right],$$

[Miyato et al. 2018]

# Multiplicative Perturbations

- We propose a new type of adversarial perturbations:

$$x_{adv} = x + r_{adv} \quad \Rightarrow \quad x_{xadv} = x \odot z$$

- Derive new loss functions:

xAT:

$$\mathcal{L}_{\text{xAT}}(\boldsymbol{x}, \boldsymbol{z}_{\text{xadv}}, \boldsymbol{\theta}) = D\left[h(y|\boldsymbol{x}, \boldsymbol{\theta}), p(y|\boldsymbol{x} \odot \boldsymbol{z}_{\text{xadv}}, \boldsymbol{\theta})\right]$$

$$\text{with } \boldsymbol{z}_{\text{xadv}} = \arg\max_{\boldsymbol{z}} D\left[h(y|\boldsymbol{x}, \boldsymbol{\theta}), p(y|\boldsymbol{x} \odot \boldsymbol{z}, \boldsymbol{\theta})\right]$$

xVAT:

$$\mathcal{L}_{\text{xVAT}}(\boldsymbol{x}, \boldsymbol{z}_{\text{xadv}}, \boldsymbol{\theta}) = D\left[p(y|\boldsymbol{x}, \boldsymbol{\theta}), p(y|\boldsymbol{x} \odot \boldsymbol{z}_{\text{xadv}}, \boldsymbol{\theta})\right]$$

$$\text{with } \boldsymbol{z}_{\text{xadv}} = \arg\max_{\boldsymbol{z}} D\left[p(y|\boldsymbol{x}, \boldsymbol{\theta}), p(y|\boldsymbol{x} \odot \boldsymbol{z}, \boldsymbol{\theta})\right]$$

# Training

- We use the $L_0$-norm of $z$ to regularize the learning:

$$\boldsymbol{z}_{\text{xadv}} = \arg\max_{\boldsymbol{z}} \Delta D(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) + \lambda \|\boldsymbol{z}\|_0$$

$$= \arg\max_{\boldsymbol{z}} \Delta D(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) + \lambda \sum_{j=1}^{P} 1_{[z^j \neq 0]}$$

- However, the discrete essence of $z$ makes it undifferentiable.

# Optimization

- We adopt the Stochastic Variational Optimization and the Hard Concrete Gradient Estimator techniques for optimization.

$$\log \boldsymbol{\alpha}_{\text{xadv}} = \arg\max_{\log \boldsymbol{\alpha}} \mathbb{E}_{\boldsymbol{u}\sim\mathcal{U}(0,1)} \left[\Delta D(g(f(\log \boldsymbol{\alpha}, \boldsymbol{u})), \boldsymbol{x}, \boldsymbol{\theta})\right] + \lambda \sum_{j=1}^{P} \sigma\left(\log \alpha^j - \beta \log \frac{-\gamma}{\zeta}\right) \tag{12}$$

with

$$f(\log \boldsymbol{\alpha}, \boldsymbol{u}) = \sigma\left((\log \boldsymbol{u} - \log(1-\boldsymbol{u}) + \log \boldsymbol{\alpha})/\beta\right)(\zeta-\gamma)+\gamma,$$

$$g(\cdot) = \min(1, \max(0, \cdot)),$$

- Generate the mask: $\hat{\boldsymbol{z}}_{\text{xadv}} = g(f(\log \boldsymbol{\alpha}_{\text{xadv}}, \boldsymbol{u})), \qquad \boldsymbol{u} \sim \mathcal{U}(0,1).$

# Transductive vs. Inductive Training

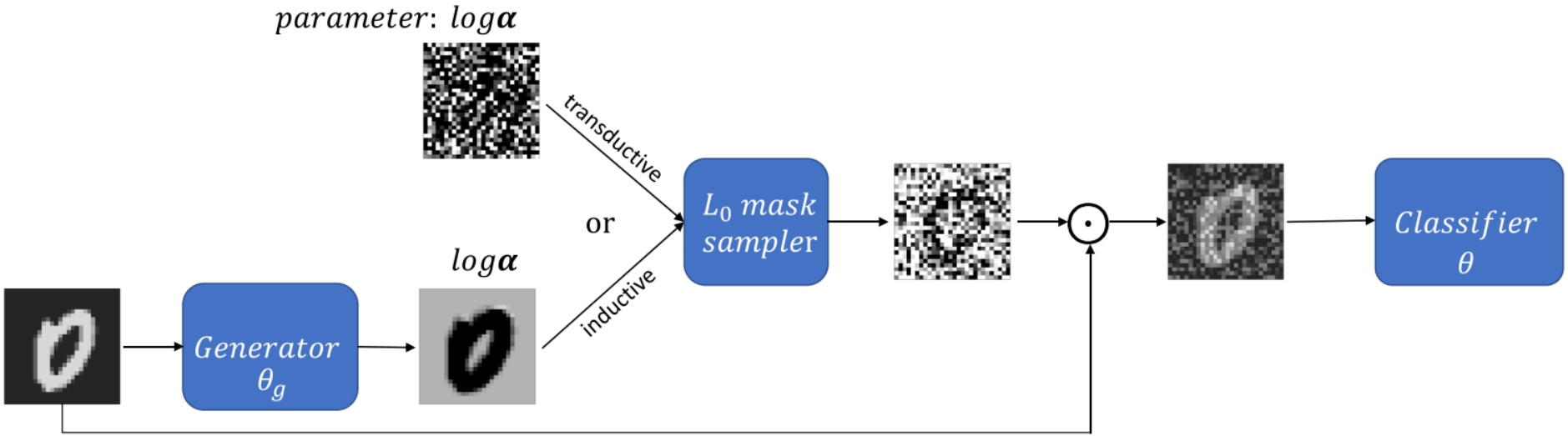
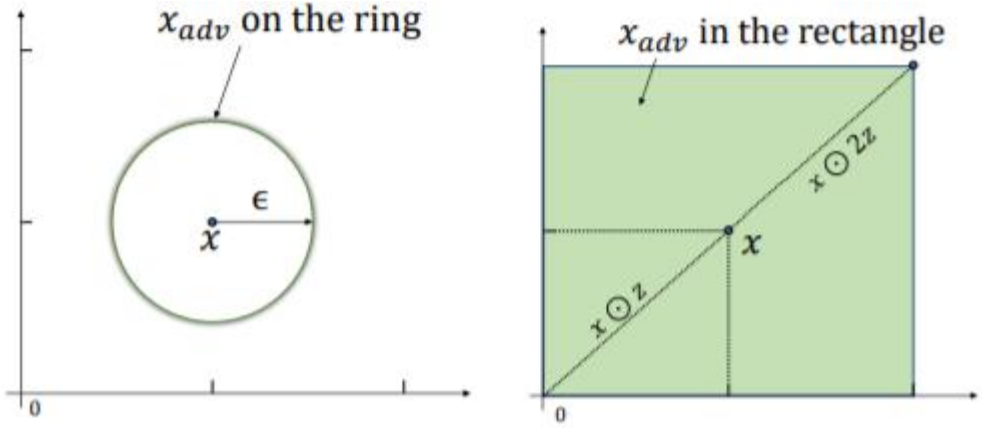



Fig. 2. The pipeline of the transductive and inductive implementations of multiplicative adversarial training.

# Shrink or Expand

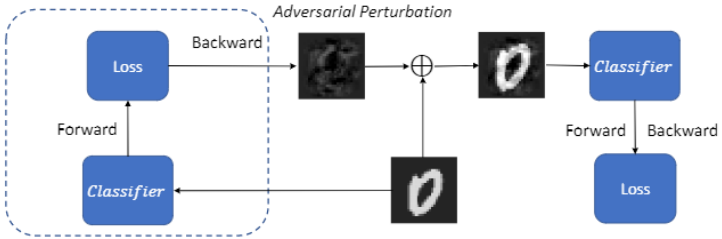



(a) Additive Perturbation (b) Multiplicative Perturbation

Fig. 3. The effect of $\epsilon$ on different perturbations. (a) shows that the additive perturbations are on the surface of a ball with the radius $\epsilon$. (b) demonstrates that our multiplicative perturbations are distributed within the rectangle.

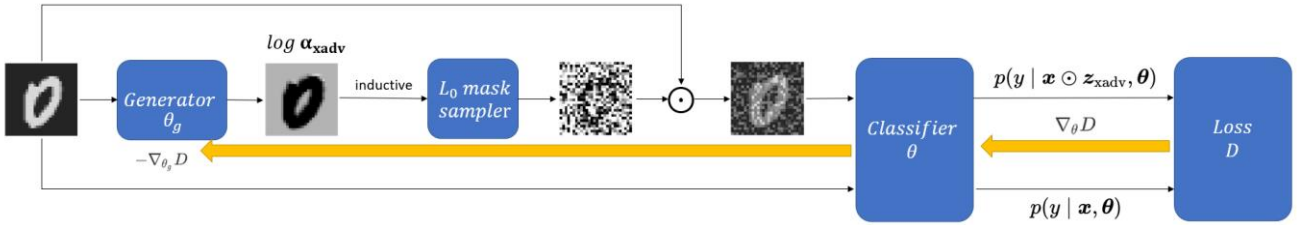
# Efficient Computing

- Both AT and VAT resort to optimizing additive perturbations and classifier parameter alternatively in two steps.




(a) *Additive* Perturbation Pipeline

- xAT/xVAT can update them simultaneously in one step.



# Experiments: Semi-supervised learning

TEST ACCURACIES OF SEMI-SUPERVISED LEARNING ON MNIST, SVHN AND CIFAR-10. THE RESULTS ARE AVERAGED OVER 5 RUNS.

| Method | Test Accuracy (%) | | |
|---|---|---|---|
| | MNIST $N_l$=100 | SVHN $N_l$=1000 | CIFAR-10 $N_l$=4000 |
| GAN with feature match [22] | **99.07** | 91.89 | 81.37 |
| CatGAN [23] | 98.09 | - | 80.42 |
| Ladder Networks [24] | 98.94 | - | 79.60 |
| Π-model [14] | - | 94.57 | 83.45 |
| Mean Teacher [16] | - | **94.79** | 82.26 |
| VAT [6] | 98.64 | 94.23 | 85.18 |
| xVAT (Transductive) | 98.02 | 93.99 | 85.82 |
| xVAT (Inductive) | 97.82 | 94.22 | **86.59** |

# Experiments: Supervised learning

TEST ACCURACIES OF SUPERVISED LEARNING ON CIFAR-10 AND CIFAR-100. THE RESULTS ARE AVERAGED OVER 5 RUNS.

| Method | Test Accuracy (%) | |
|---|---|---|
| | CIFAR-10 | CIFAR-100 |
| Baseline (MLE) [14] | 93.24 | 73.58 |
| Π-model [14] | **94.44** | 73.68 |
| Temporal ensembling [14] | 94.40 | 73.70 |
| AT, $L_\infty$(ours)* | 93.90 | 74.04 |
| VAT [6] | 94.19 | 75.02 |
| xAT (Inductive) | 93.70 | 74.62 |
| xVAT (Inductive) | 93.88 | **75.30** |

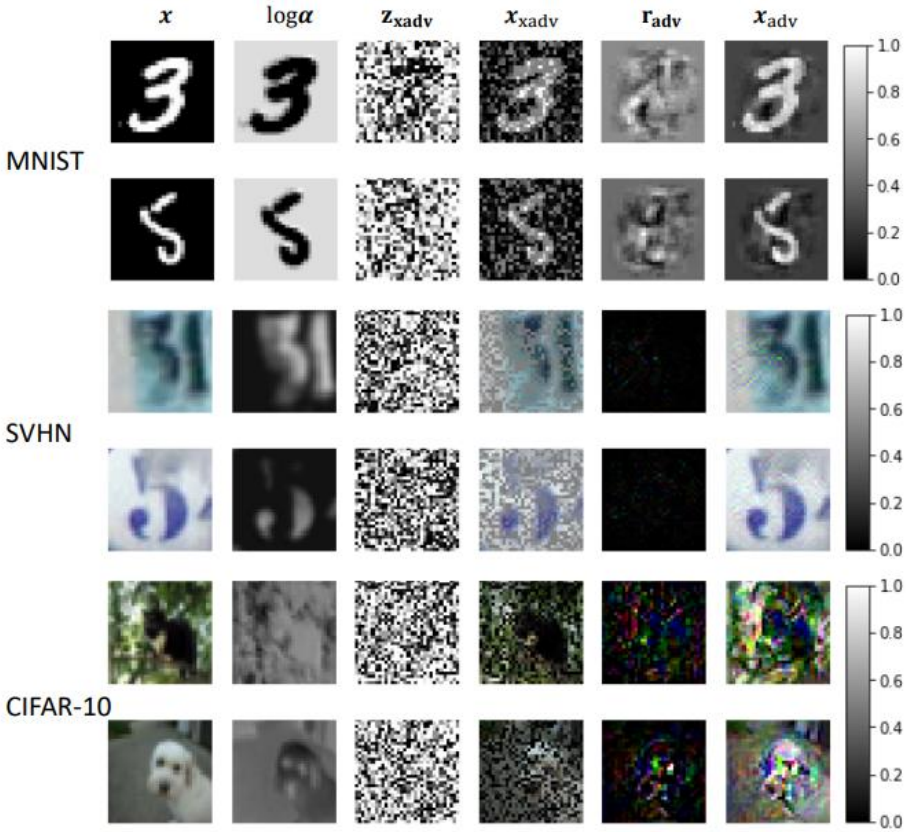
# Experiments: Speed

THE TRAINING SPEEDS OF VAT AND XVAT ON THE FOUR BENCHMARK DATASETS. THE RESULTS ARE AVERAGED OVER 5 RUNS.

| Method | Seconds per epoch | | | |
|---|---|---|---|---|
| | MNIST | SVHN | CIFAR-10 | CIFAR-100 |
| VAT (ours)* | 4.31 | 54.3 | 51.3 | 51.5 |
| xVAT (Transductive) | 4.54 | 36.6 | 34.1 | 39.3 |
| xVAT (Inductive) | 4.33 | 35.7 | 33.6 | 34.4 |

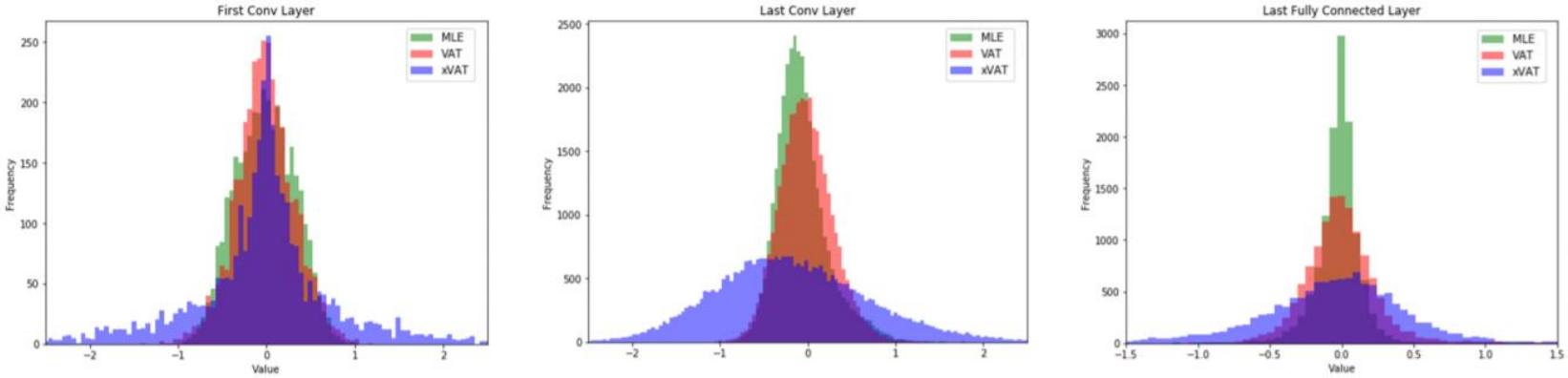
# Multiplicative vs. Additive Perturbations

- The multiplicative perturbations are
  (1) More perceptible
  (2) More interpretable



# Robustness and Sparsity

Histograms of weights shows that xVAT learns a denser classifier from multiplicative perturbations with more non-zero weights than VAT and MLE, which may indicate the adversarially trained DNNs need more capacities (active neurons) to against multiplicative perturbations.




(a) First convolutional layer (b) Last convolutional layer (c) Last fully connected layer

Fig. 6. Histograms of the classifier weights learned by MLE, VAT and xVAT on CIFAR-100. The histograms are computed from different CNN layers.

# Thank you!