Adaptive Batch Normalization Networks for Adversarial Robustness

AVSS 2024

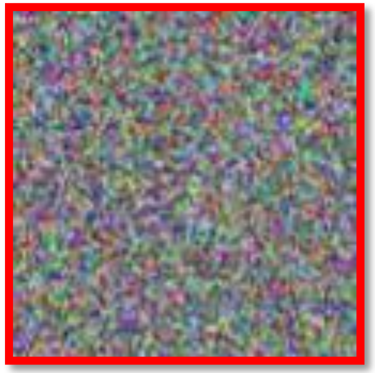
Shao-Yuan Lo Vishal M. Patel

Johns Hopkins University

What are Adversarial Examples

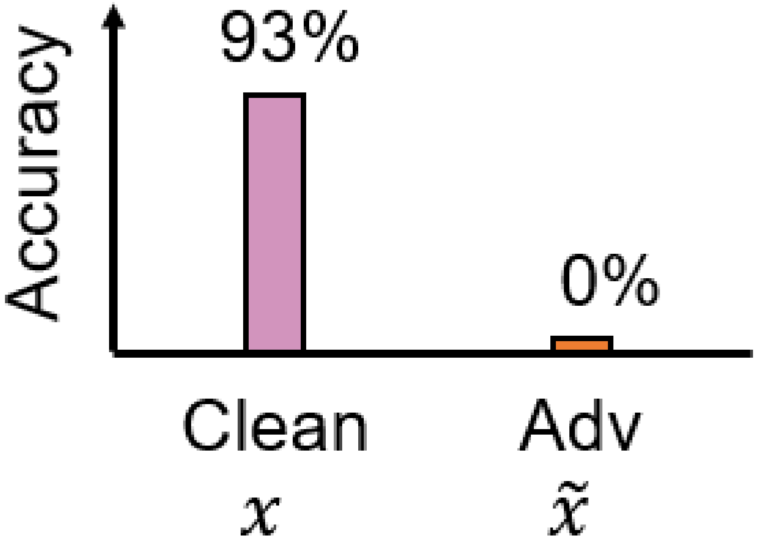
- Deep networks are **vulnerable** to adversarial examples.

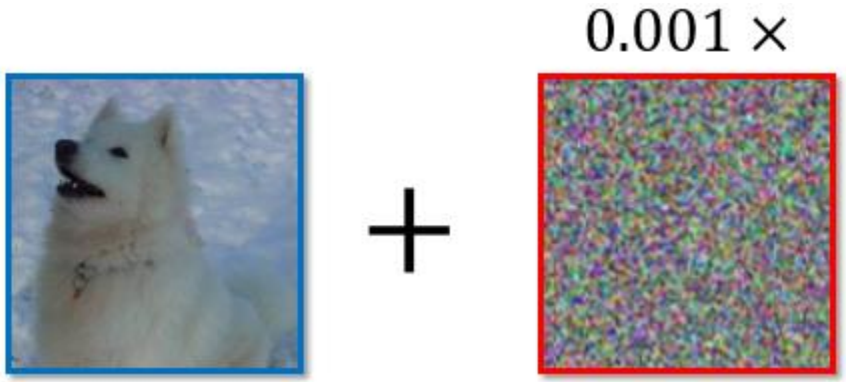
$f_\theta($  $) =$ "Dog"

$f_\theta($  $+ \; 0.001 \times$  $) =$ "Cat"

What are Adversarial Examples

- Dataset: CIFAR-10
- Network: ResNet-50





How to Generate Adversarial Examples

- Train a model
  - **min Loss(f(x), y; θ)**
  - **Minimize** the loss function w.r.t. **model parameters θ**

- Generate adversarial examples
  - Most common method: Gradient-based method, e.g., FGSM.
  - **max Loss(f(x+δ), y; θ)**
  - **Maximize** the loss function w.r.t. **adversarial perturbation δ**

Defense by Adversarial Training

- Adversarial Training (AT) is a strong defense against adversarial examples.
- **Core idea: Train with adversarial examples.**

Standard Training

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}}\, L(x, y; \theta)$$

Adversarial Training

$$\theta^* = \underbrace{\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \Big[\underbrace{\max_{\delta\in\mathbb{S}} L(x+\delta, y; \theta)}_{\text{Generate adversarial examples}} \Big]}_{\text{Train model parameters}}$$

[Madry et al. ICLR'18]

Defense by Adversarial Training

- However, AT involves a **min-max optimization**, which is **extremely expensive**.

Iterative adversarial example generation

$$x^{t+1} = \Pi_{x+\mathbb{S}}\left(x^t + \alpha \cdot sign(\nabla_x L(x, y; \theta))\right) \longrightarrow x + \delta$$

Adversarial Training

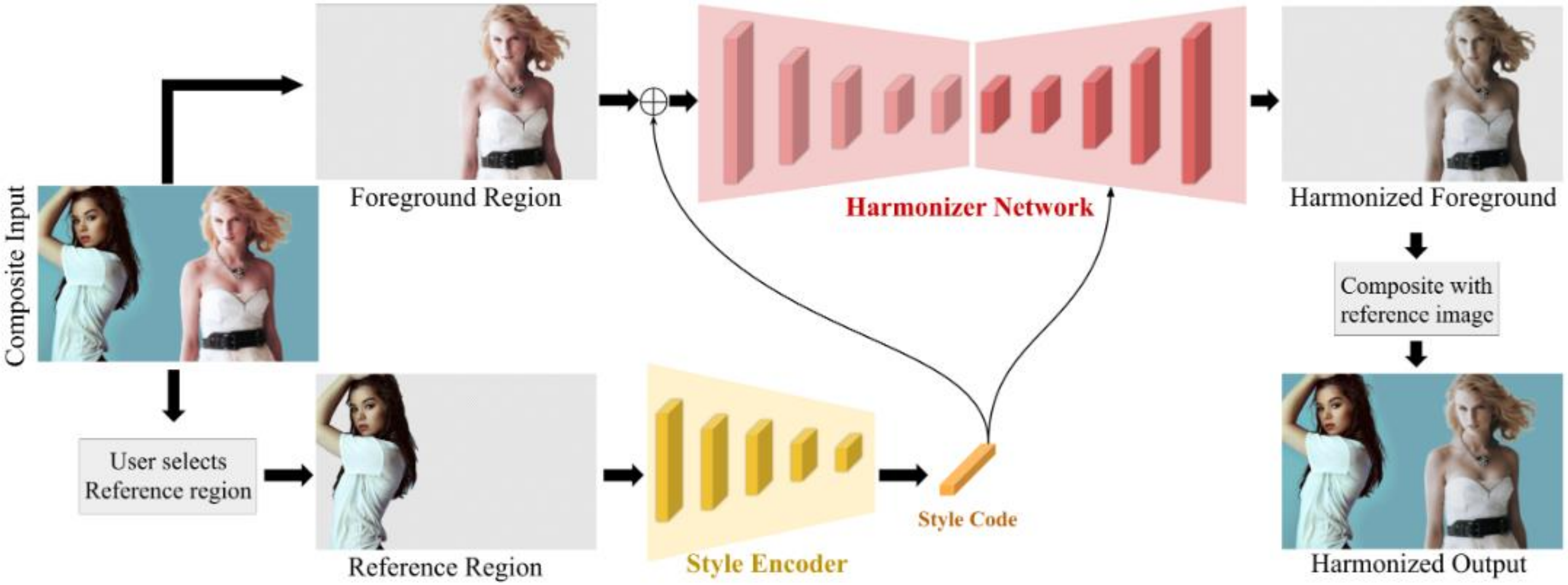
$$\underbrace{\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\underbrace{\max_{\delta \in \mathbb{S}} L(x + \delta, y; \theta)}_{\text{Generate adversarial examples}} \right]}_{\text{Train model parameters}}$$

[Madry et al. ICLR'18]

How to design a defense method that gets rid of AT but is still robust against strong adversarial examples?

Inspiration from Image Harmonization

- Image harmonization: Match a foreground object to a new background scene.
- A style code is extracted by a style encoder and is passed to the adaptive instance norm layers of the harmonizer network.

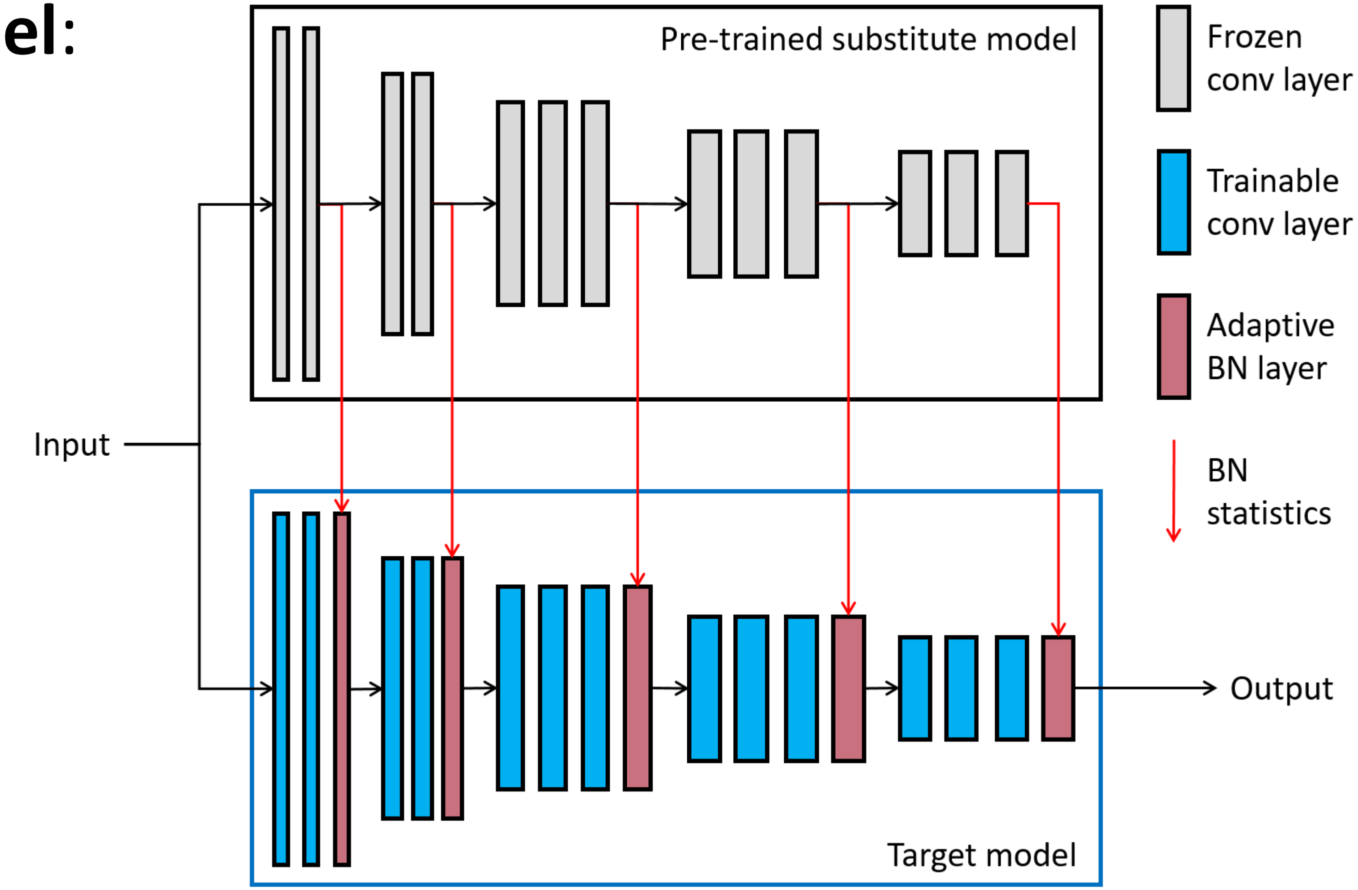


[Valanarasu et al. ICLR’23]

Adaptive Batch Normalization Network

- **Pre-trained substitute model**: A public model trained on large-scale datasets (e.g., ImageNet)

- **Target model**: The model that we are training for a downstream task



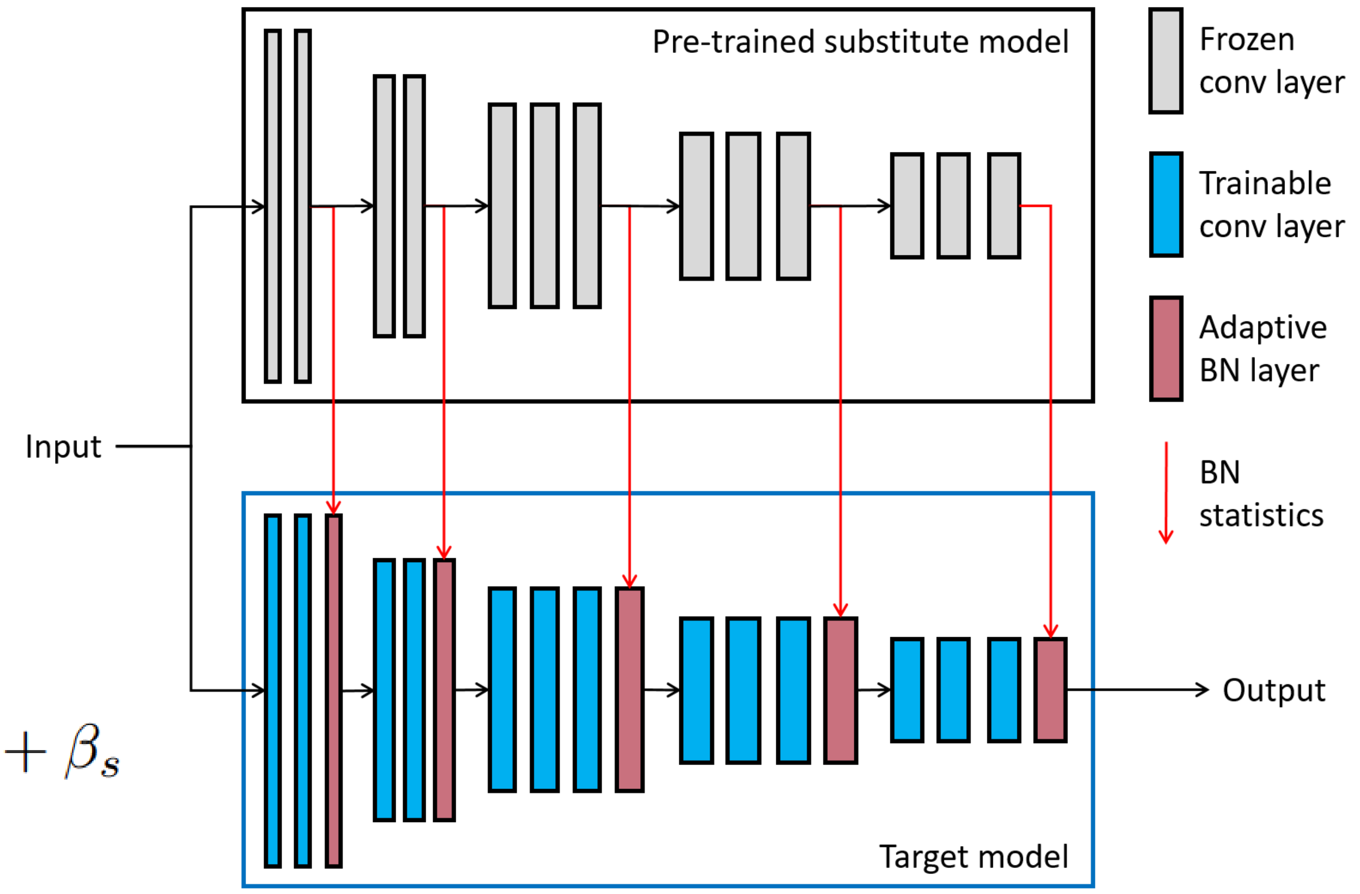
Adaptive Batch Normalization Network

- **Standard BN**

$$z' = \gamma \left[\frac{z - \mu(z)}{\sigma(z)} \right] + \beta$$

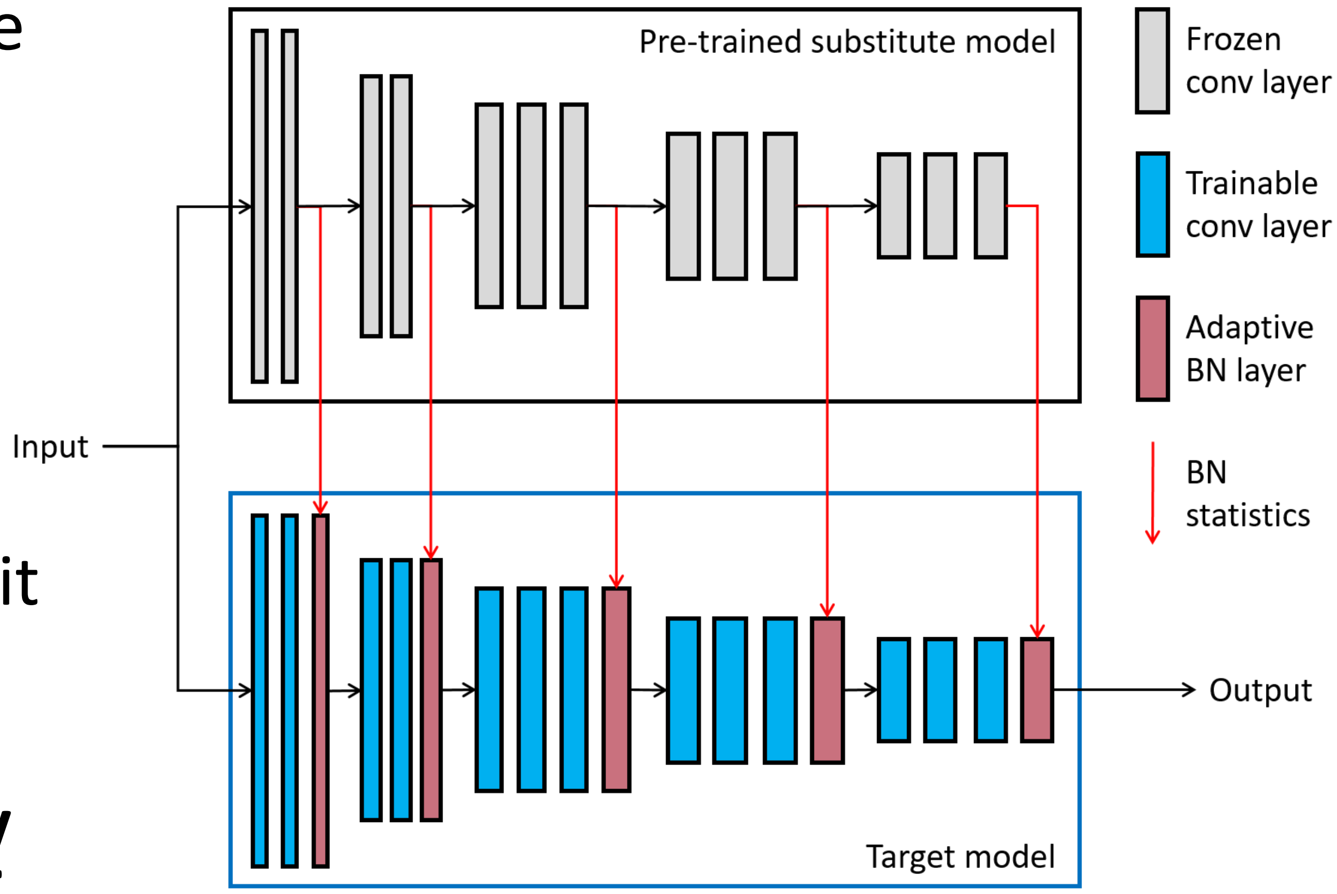
- **Proposed adaptive BN**

$$z'_t = \gamma_s \left[\sigma(z_s) \left[\frac{z_t - \mu(z_t)}{\sigma(z_t)} \right] + \mu(z_s) \right] + \beta_s$$

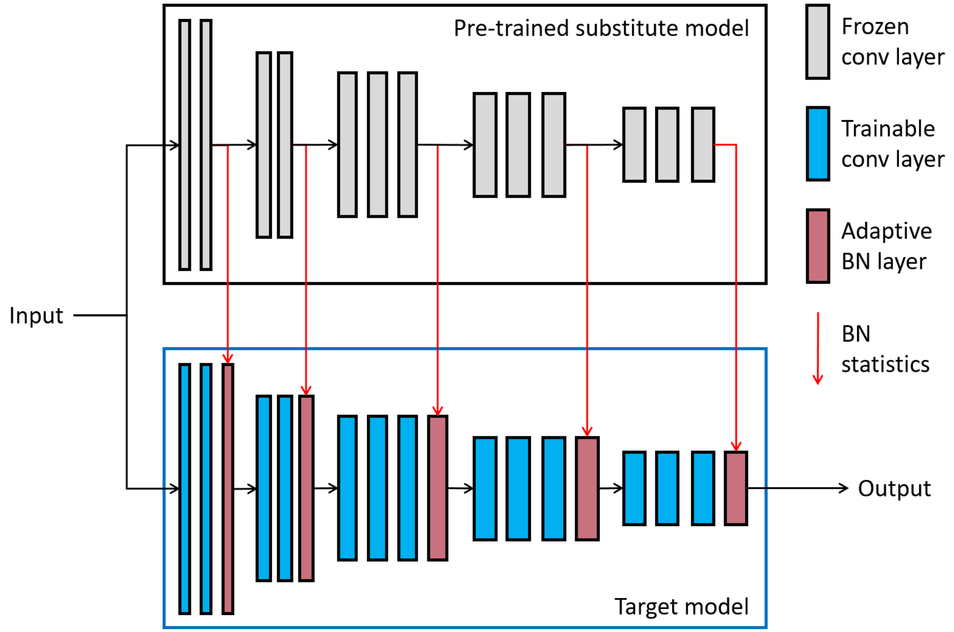


Adaptive Batch Normalization Network

- Adversary would **perturb** the **target model's BN**.
- The **substitute model's BN** are relatively **unaffected**.
- The substitute model is trained on **large-scale datasets different from the target task dataset**, making it harder for adversary to transfer the attack.
- The model is trained on **only clean data without using AT**.



Training Time Complexity



- Let us set that each network pass (i.e., a forward pass or a backward pass) has **N** computational complexity.

- ABNN:
  2N + N = 3N

- PGT-AT:
  2N x tmax + 2N = 2N (tmax + 1)

- PGT-AT has
  2N (tmax + 1) / 3N = ***0.67 (tmax + 1)***
  times more training complexity than ABNN

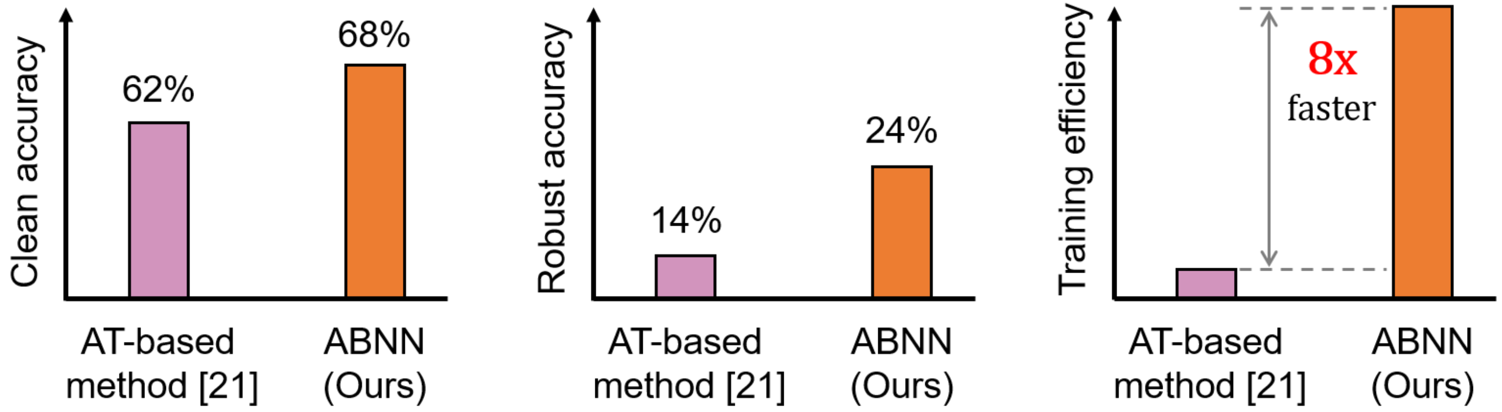
$$x^{t+1} = \Pi_{x+\mathbb{S}}\left(x^t + \alpha \cdot sign(\nabla_x L(x, y; \theta))\right)$$

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\max_{\delta\in\mathbb{S}} L(x+\delta, y; \theta)\right]$$

Results

Method	Clean	ROA	Training cost
No Defense [14]	93.0	7.0	2N
OUDefend [21]	62.0	13.6	24N
ABNN (Ours)	68.3	24.4	3N

- **Dataset**: UCF-101
- **Target model**: 3D ResNeXt-101
- **Substitute model**: 3D ResNet-18 pre-trained on Kinetics-400
- **Attack**: ROA with 10% area, tmax=5



(a) Better clean data performance (b) Better robustness generalization (c) Better training efficiency

Results

Table 1. Evaluation results (%) under the PGD attack on the CIFAR-10 dataset.

Method	Clean	PGD	Training cost
No Defense	93.4	0.0	2N
PGD-AT [23]	83.3	51.6	12N
ABNN (Ours)	87.5	31.5	3N

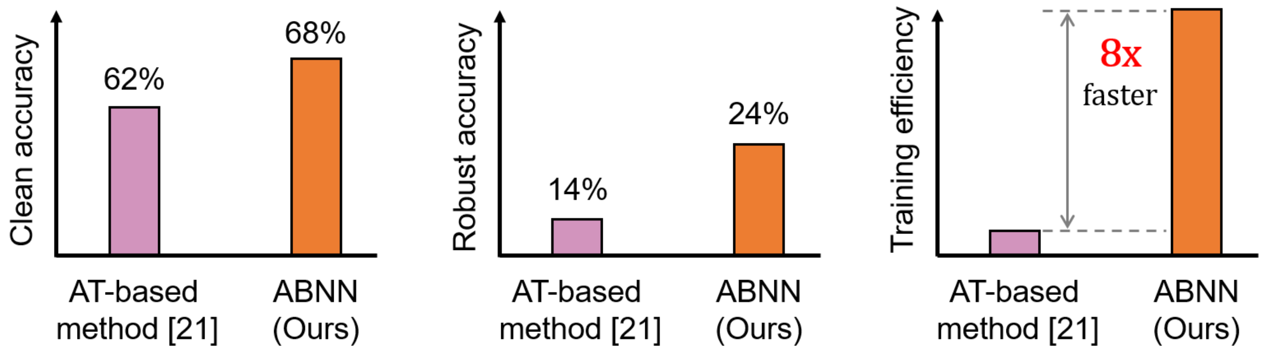
Table 2. Evaluation results (%) under the PGD attack on the UCF-101 dataset.

Method	Clean	PGD	Training cost
No Defense [14]	93.0	0.0	2N
OUDefend [21]	62.0	58.6	24N
ABNN (Ours)	68.3	43.4	3N

Conclusion

- The proposed adversarial defense **ABNN** is a non-AT method that **gets rid of the extremely time-consuming AT**.

- Compared to traditional AT-based approaches, the proposed ABNN achieves **higher clean data performance**, **better robustness generalization**, and **significantly lower training time complexity**.



(a) Better clean data performance (b) Better robustness generalization (c) Better training efficiency