# Achieving Both Model Accuracy and Robustness by Adversarial Training with Batch Norm Shaping

**Brian Zhang** (*Harrison High School, Indiana*),

**Shiqing Ma** (*Rutgers University*)

Supported by **IARPA**

TrojAI

ICTAI 2022

# Deep Learning in Critical Applications

Autonomous Driving: recognize traffic signs, pedestrians, other vehicles

Face Recognition: identity verification, access authorization

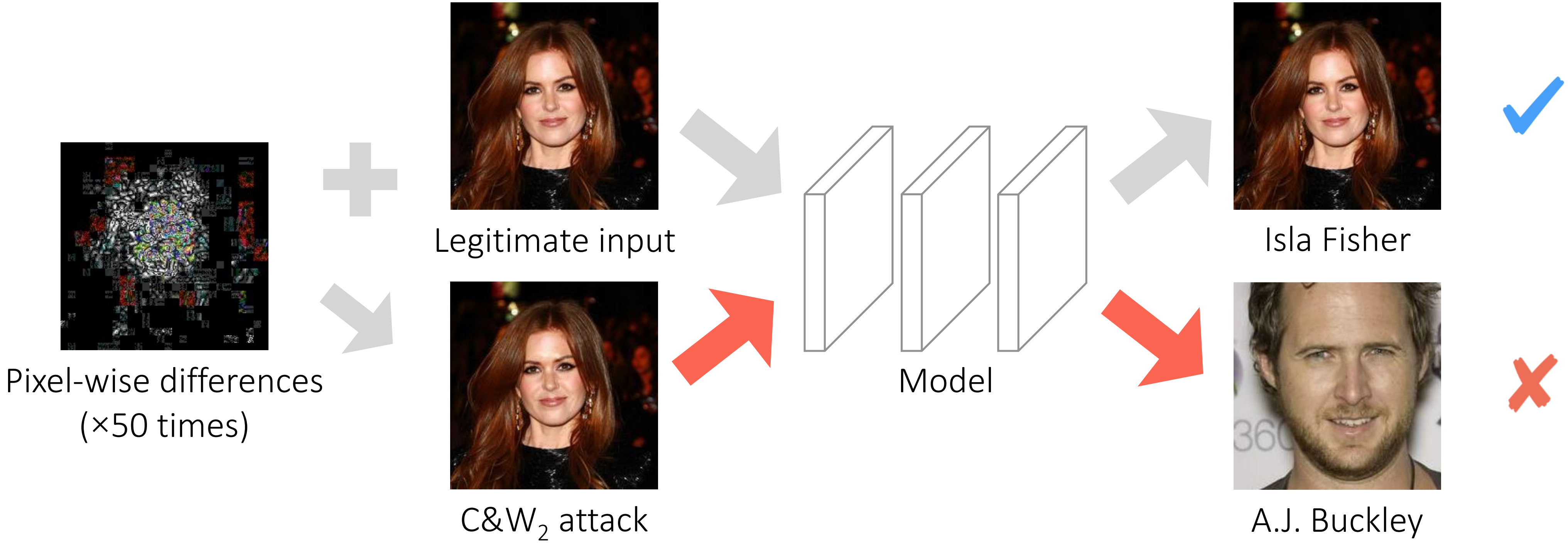
Criminal Identification: validate criminal profile, cross-check history records



Loan Authorization: financial record verification, background check

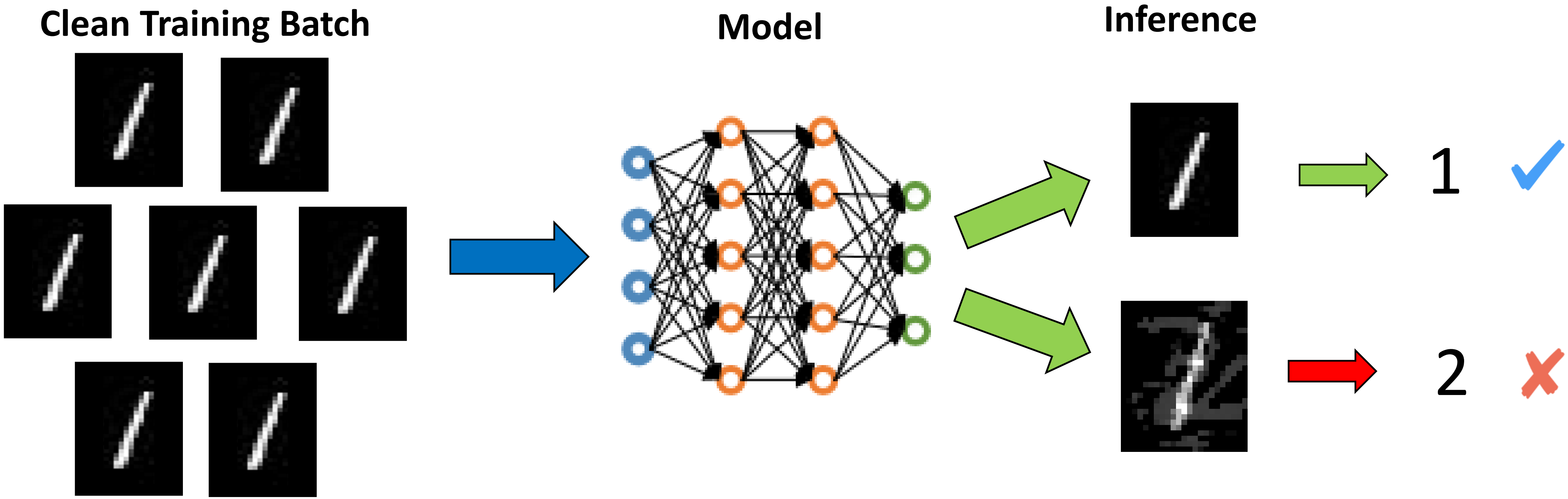
# Adversarial Attacks

- Adversarial attacks perturb model inputs generated to fool neural networks (i.e., unexpected prediction results).



# Normal Training and Clean Training Batch

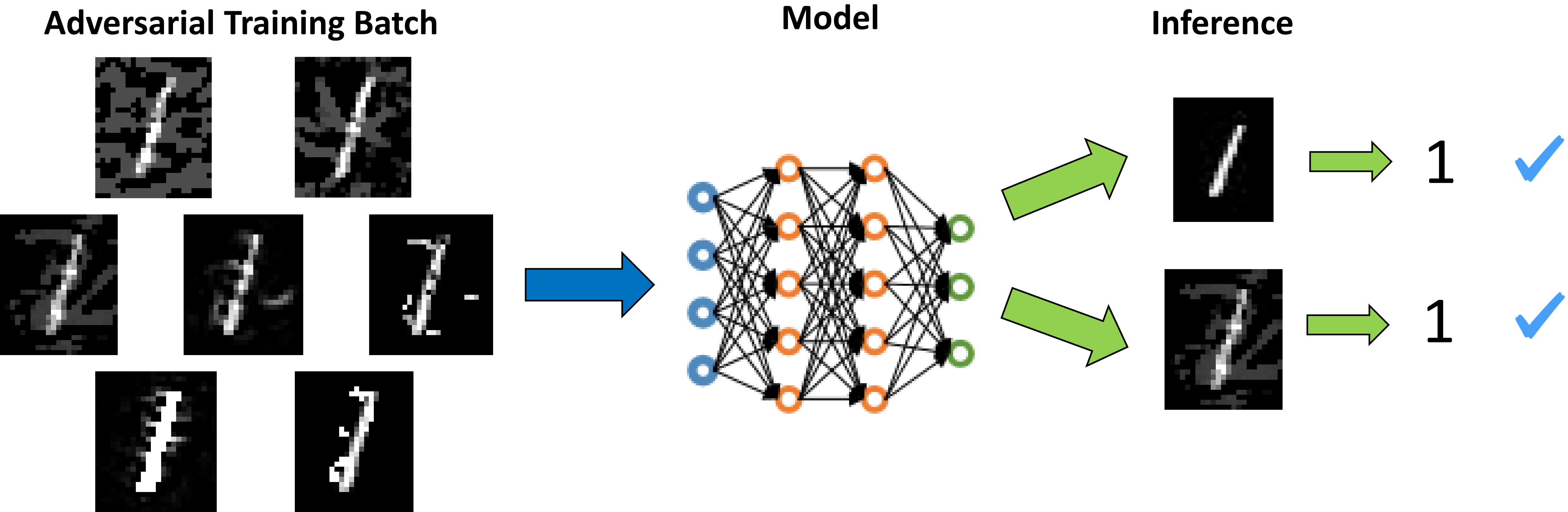
- Training input provided in ***batches*** of clean, unperturbed samples
- Model weights are updated based on the batch
- Normal training techniques can achieve an accuracy of **0.94** (94% of inferences correct) on clean samples, however, the accuracy could degrade to close to **0** on adversarial samples



# Adversarial Training and Adversarial Training Batch

**The improvement in robustness comes at the cost of accuracy**

- Input batches of perturbed, or adversarial samples into a training model
- PGD can achieve an accuracy of **0.87** (down from 0.94) on clean samples and **0.47** on adversarial samples



# Our contributions

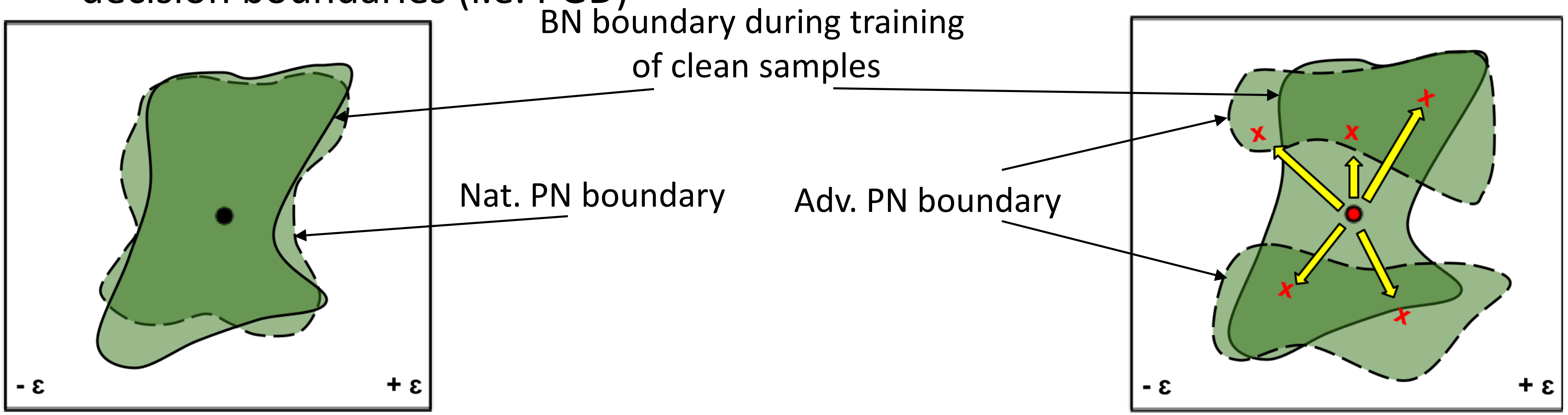
- **Problem Statement:** We want to achieve both model accuracy and robustness in adversarial training
- We conduct an in-depth study on the confounding factors of Batch Normalization in adversarial training.
- We propose a technique that suppresses these confoundings and boost training performance
- We evaluate our technique over two existing adversarial training methods: PGD and TRADES
  - We achieve model accuracy of **0.94**, comparable to normally trained models
  - We improve robustness against PGD attacks from 0.47 to **0.816**
  - We improve robustness against TRADES attacks from 0.46 to **0.817**
  - We have a robustness of **0.51** against the strongest adaptive attack for our model

# Background: Batch Norms (BN) and Population Norms (PN)

- In Neural Network training, internal activation values are usually normalized to achieve quick convergence, by making the mean and STD to be 0 and 1, respectively
- *Batch Norms (BNs)* are a normalization using batch statistics, generally for training
- *Population Norms (PNs)* are a normalization using population statistics, generally for inference
- **Therefore, decision boundaries can be considered parameterized on these norms**

# Confoundings of Batch Normalization in Adversarial Training

- **Confounding I: PN and BN Norm differences degrade model accuracy**
  - PN is modified by adversial training
- Alignment of clean input decision boundaries determines model accuracy:
  - Natural BN decision boundaries align well with naturally trained PN decision boundaries
  - Natural BN decision boundaries do not align well with adversarially trained PN decision boundaries (i.e. PGD)

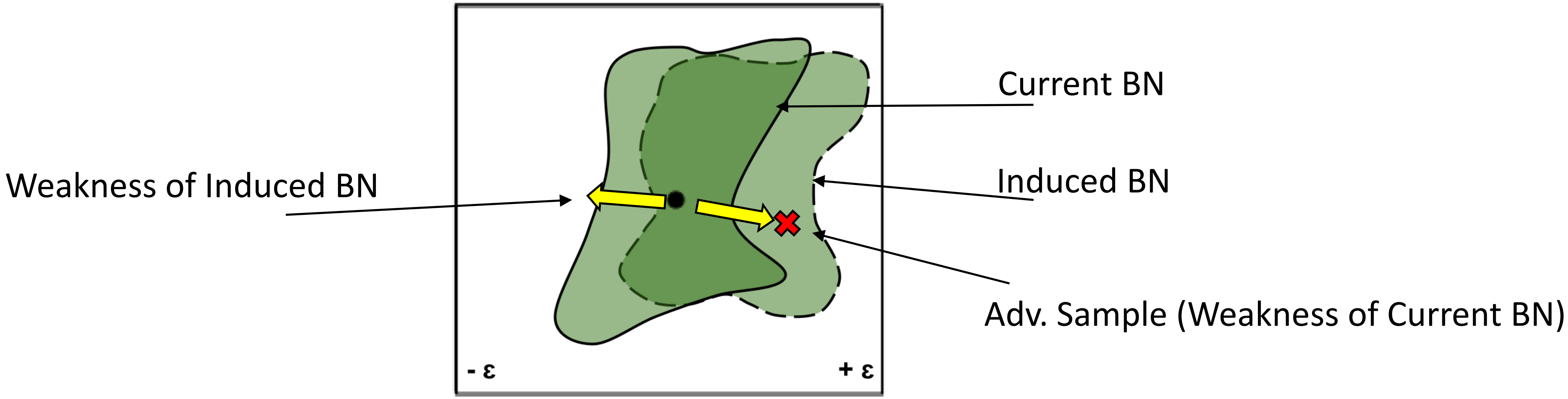



Nat. training

Adv. training

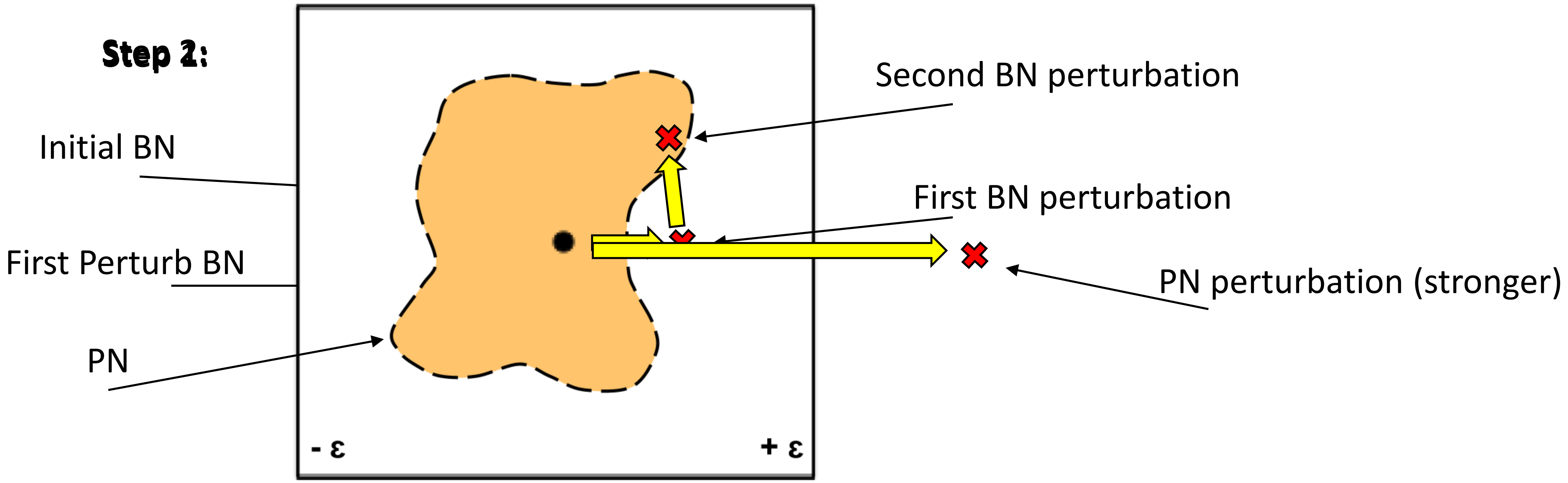
# Confoundings of Batch Normalization in Adversial Training

- **Confounding II: BN hinders model robustness**
  - Adversial training is a minmax problem
  - Proper perturbations made along the weakest point in the BN decision boundary
- Moving Target Effect:
  - Clean BN is used to generate adversial samples
  - However, BNs of adversial samples used in training (instead of those of clean samples) may induce a different decision boundary



# Confoundings of Batch Normalization in Adversarial Training

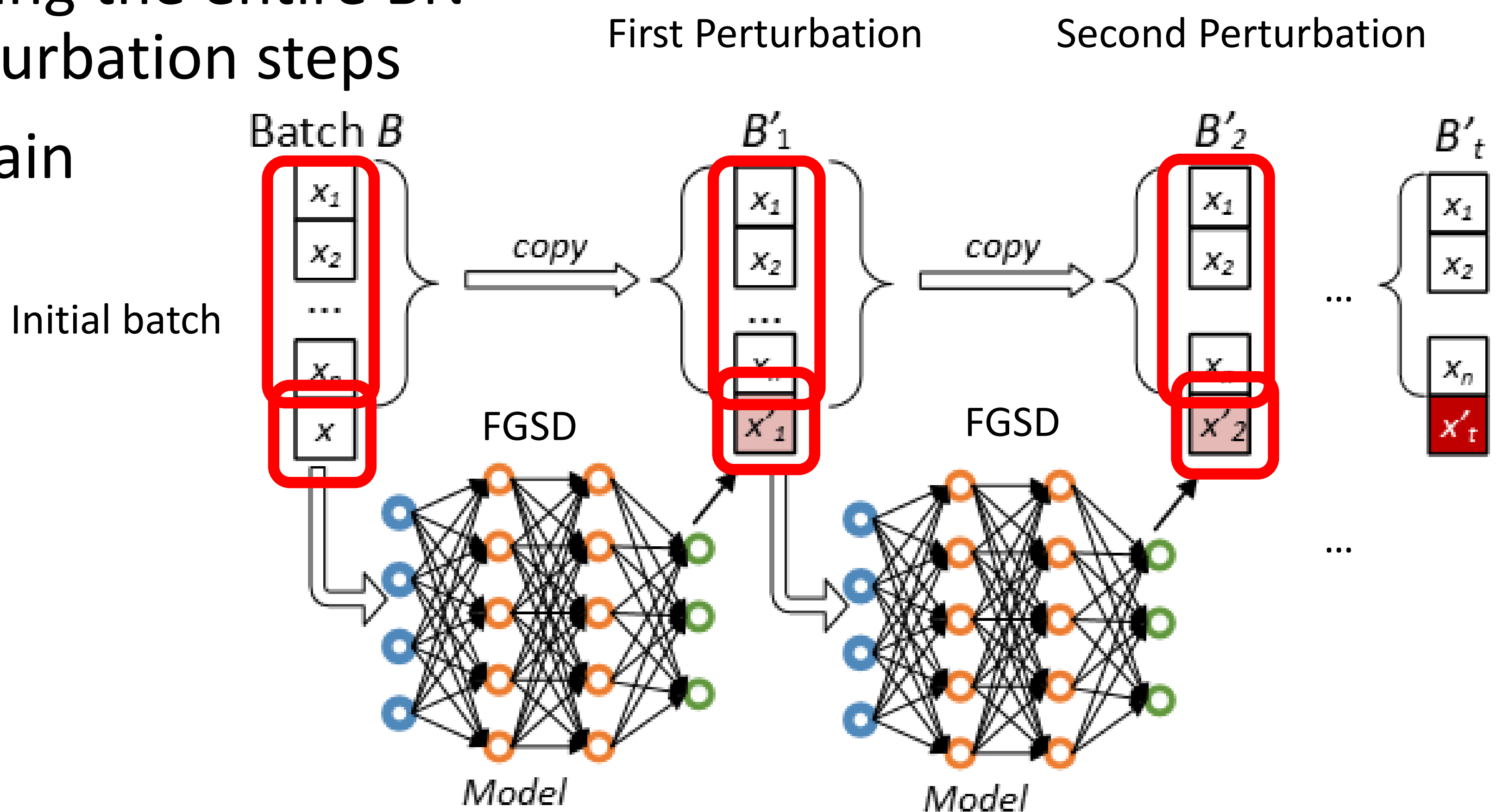
- **Confounding III: Norm differences weaken BN attacks**
  - Adversarial samples are generated by multiple perturbation steps (i.e., PGD)
  - The Moving Target Effect can cause irregular perturbation development
  - Attacks using PNs (constant) are generally much stronger



# Norm Shaping

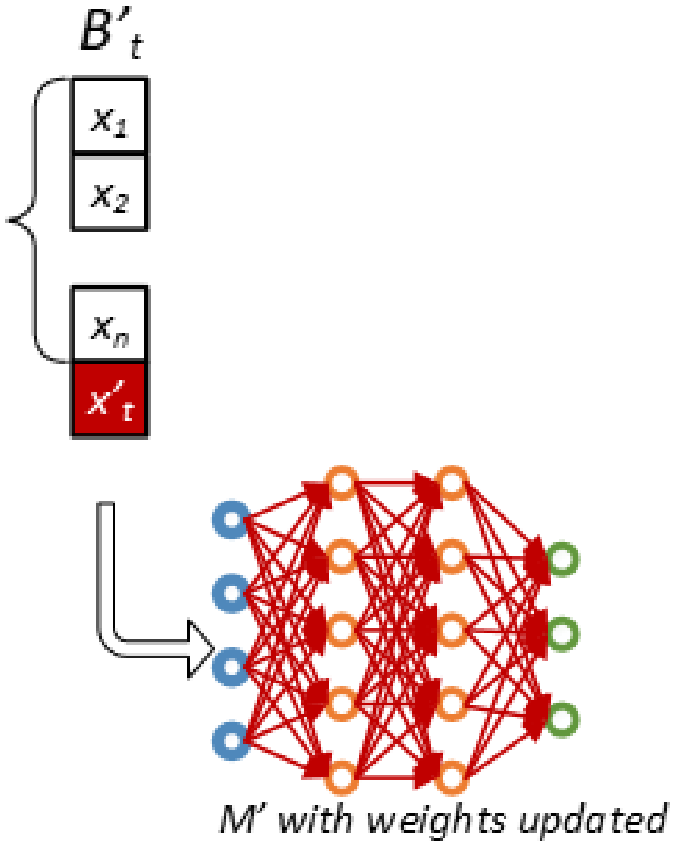
# Norm Shaping during Model training

- Divide a training batch to $n$+1 parts
- No perturbation on first $n$ parts
- Perturb last part as using the entire BN through all of the perturbation steps
- Pass entire batch to train



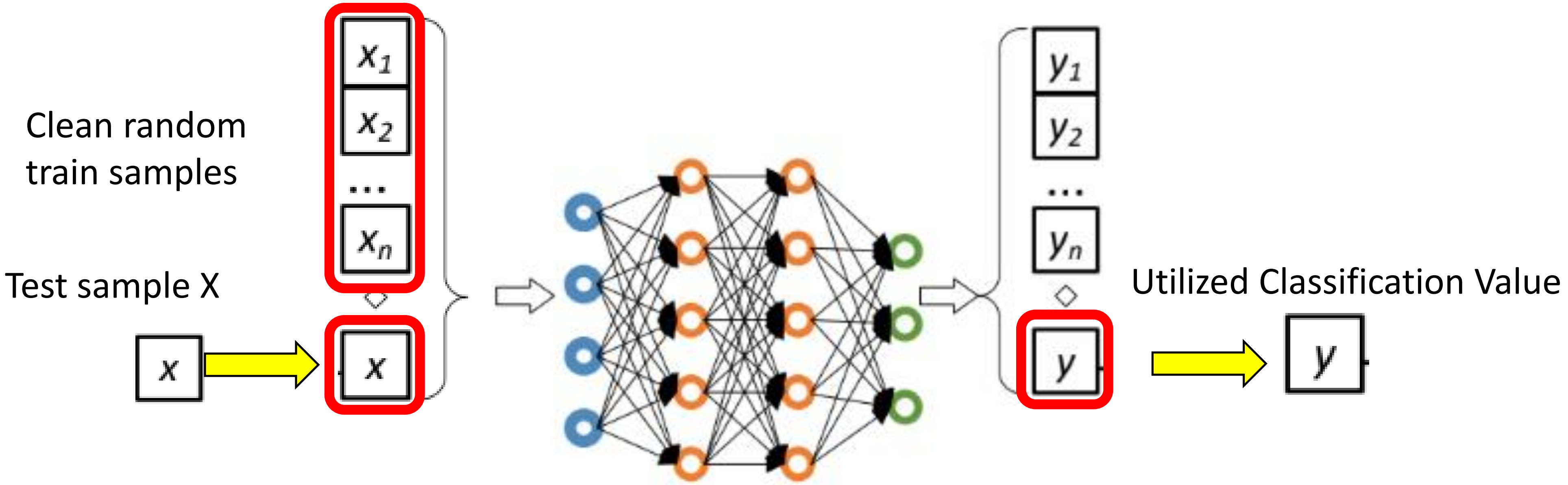
# Norm Shaping during Model training

- Train the model as usual
- Model weights updated by training with BNs



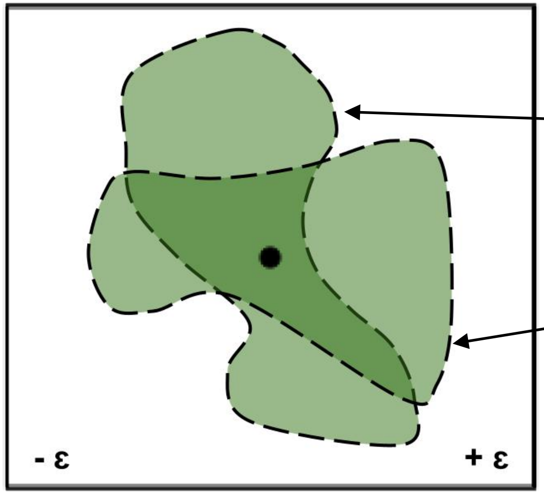
# Norm Shaping During Model Inference

- Evaluate with BNs instead of PNs
  - Combine an input sample $x$ with $n$ clean training samples
  - Use BNs of the batch in classification
  - Only utilize the classification value of the test sample $x$



# Design Justification

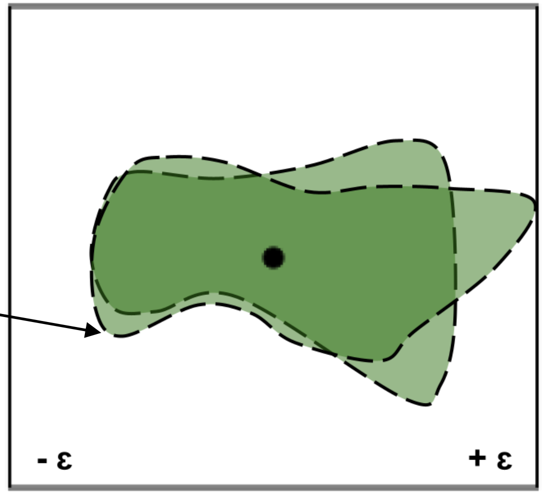
- Clean to adversarial ratio in batches (with the former dominating) keeps adversarial BN boundaries similar to clean BN boundaries
  - Model operates on BNs resembling those of natural samples- > improves accuracy
  - Stabilization of BN boundary focuses perturbations - > stronger attacks
  - Resemblance to clean BNs allow for robustness improvement for the mixed batch- > more robust model




Adversial BN boundary **without** Norm shaping

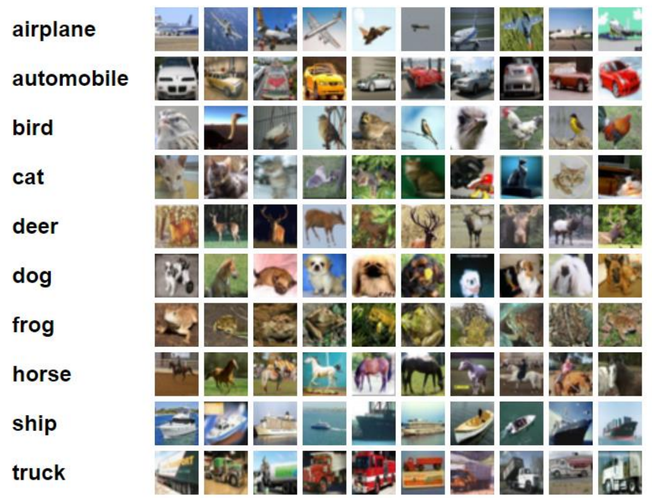
Clean BN boundary

Adversial BN boundary **with** Norm shaping



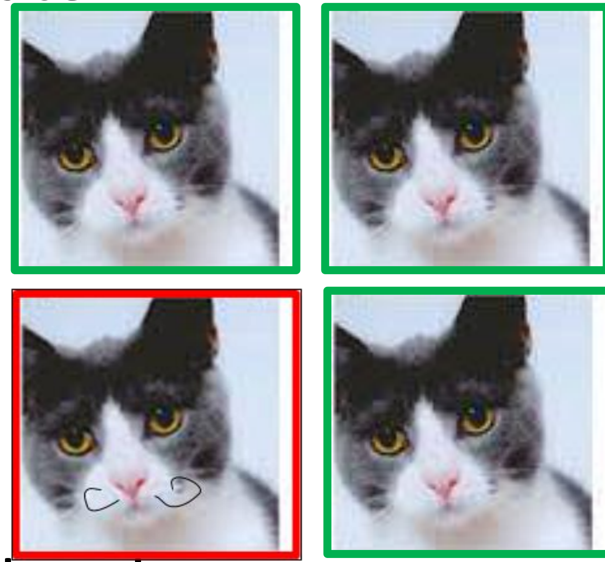
# Experiment Setup: Data and Configuration

- We use CIFAR10 as our data set
  - Same ResNet w32-10 structure utilized by PGD
- We implement our algorithm on PGD and TRADES
  - TRADES computes loss via *KL-Divergence* between adversarial and benign samples
- We define the constant $n$ used in our norm shaping technique as 3
- We mostly reuse default training configuration from PGD
  - **80,000** training steps
  - **8/255** perturbation bound
  - Step size of **2**
  - **10** attack steps (in adversial sample generation)
  - We use a batch size of **64**
  - More information can be found in the `config.json` file of PGD



# Experiment Setup: Attacks

- We use four existing attacks and our own attack to evaluate robustness
  - *PGD*
  - *C&W L2 attack*
  - *FGSM*
    - Perturbation bound of 8 pixels changed to 16 pixels
  - *Deepfool*
  - *Our own norm shaping attack*
- There are always 100 adversarial samples generated per batch
  - The first four attacks rely on PNs
  - Our last attack relies on BNs, with the ratio of clean to unclean being 3:1



# Results: PGD-based adv. training

- Compared to default PGD, our method achieves better clean accuracy (0.942 vs 0.869)
- Our model receives nearly the best robustness in all attacks but our own

**Default PGD**

| Attack | Clean Acc | Robust Acc |
|---|---|---|
| PGD | 0.869 | 0.47 |
| CW | | 0.821 |
| FGSM | | 0.376 |
| Deepfool | | 0.071 |
| Shaping | | 0.614 |

**Our technique**

| Attack | Clean Acc | Robust Acc |
|---|---|---|
| PGD | 0.942 | 0.816 |
| CW | | 0.904 |
| FGSM | | 0.740 |
| Deepfool | | 0.911 |
| Shaping | | 0.533 |

# Results: TRADES-based adv. training

- Similar results when using norm shaping in TRADES adv. training
- Compared to default TRADES, our method achieves better clean accuracy (0.944 vs 0.888)
- Our model receives nearly the best robustness in all attacks but our own

**Our method is effective regardless of the underlying adversarial training method**

**Default TRADES**

| Attack | Acc | R.Acc |
|---|---|---|
| PGD | 0.888 | 0.462 |
| CW | | 0.845 |
| FGSM | | 0.347 |
| Deepfool | | 0.066 |
| Shaping | | 0.533 |

**Our technique**

| Attack | Acc | R.Acc |
|---|---|---|
| PGD | 0.944 | 0.817 |
| CW | | 0.899 |
| FGSM | | 0.737 |
| Deepfool | | 0.914 |
| Shaping | | 0.542 |

# Results: Adaptive Attack

- These adaptive attacks are under the assumption that the attacker knows:
  - All the training samples but not the specific ones used in norm shaping (Slightly weaker adaptive attack)
  - The exact training samples used in norm shaping (Adaptive). It is the strongest attack to our method
- Notice that our models still have over **0.5** robustness

| | PGD+Shaping | TRADES+Shaping |
|---|---|---|
| Slightly weaker attack | 0.518 | 0.514 |
| Adaptive attack | 0.508 | 0.505 |

TABLE III: Adaptive attack results

# Results: Ablation Study

With training ratio **1:3** and up, increasing the number of clean samples does not cause significant effect, implying that the *BN is sufficient*

- We used different norm shaping ratios in inference and training to test our model's effectiveness
- We use the PGD attack from CleverHans to determine our robust accuracy

**1:2 training** has the best robustness against 1:1 and [illegible] (0.92) and worse results against more clean samples

**1:3 training** (our default setting) has the best overall results

| | | adv:clean in inference | | | | | |
|---|---|---|---|---|---|---|---|
| | | **1:0** | **1:1** | **1:3** | **1:9** | **1:19** | **1:49** |
| in training | **1:1** | 0.477 | 0.478 | 0.476 | 0.476 | 0.477 | 0.476 |
| | **1:2** | 0.385 | 0.837 | 0.835 | 0.753 | 0.668 | 0.573 |
| | **1:3** | 0.281 | 0.692 | 0.820 | 0.817 | 0.813 | 0.815 |
| | **1:7** | 0.039 | 0.138 | 0.706 | 0.781 | 0.776 | 0.779 |

TABLE IV: Impact of the ratio between adversarial and clean samples on robustness against PGD attack

# Related Work

- Batch Normalization in Adversarial Training
  - Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. …
  - Philipp Benz, Chaoning Zhang, and In So Kweon. Batch normal…
  - Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W. Taylor. …
  - Cihang Xie and Alan Loddon Yuille. Intriguing properties of adv…
  - …
- Adversarial Training
  - Harini Kannan, Alexey Kurakin, and Ian Goodfellow. …
  - Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. …
  - Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. …
  - Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. …
  - …

# Conclusion

- We develop a novel training method that addresses the confoundings caused by Batch Normalization
  - Batch Normalizations lower accuracy, robustness, and attack strength
- We propose a norm shaping technique that stabilizes the batch norms by enforcing a set ratio of clean training samples in the batches
- Our experiment show that the technique can improve existing adversarial training methods such as PGD and TRADES
  - **0.94** model accuracy compared to the **0.88** baseline
  - **0.81** robustness against the PGD attack compared to the **0.47** baseline

# Thank you!