



# Aegis: Mitigating Targeted Bit-flip Attacks against Deep Neural Networks

Jialai Wang\*, Ziyuan Zhang<sup>†</sup>, Meiqi Wang\*, Han Qiu\*<sup>✉</sup>, Tianwei Zhang<sup>‡</sup>,  
Qi Li\*<sup>§</sup>, Zongpeng Li\*<sup>♣</sup><sup>✉</sup>, Tao Wei<sup>♠</sup>, and Chao Zhang\*<sup>§</sup>

\*Tsinghua University, <sup>†</sup>Beijing University of Posts and Telecommunications,  
<sup>‡</sup>Nanyang Technological University, <sup>♣</sup>Hangzhou Dianzi University, <sup>♠</sup>Ant Group,  
<sup>§</sup>Zhongguancun Laboratory

{wang-jl22, wang-mq22}@mails.tsinghua.edu.cn, zhangziy0421@bupt.edu.cn,  
{qiuhan, qli01, zongpeng, chaoz}@tsinghua.edu.cn, tianwei.zhang@ntu.edu.sg,  
lenx.wei@antgroup.com

## A Artifact Appendix

### A.1 Abstract

We teach you how to run our experiments in this appendix. If you have any questions, please let us know.

### A.2 Description & Requirements

#### A.2.1 Security, privacy, and ethical concerns

None.

#### A.2.2 How to access

<https://github.com/wjl123wjl/Aegis.git>

The GitHub hash is:

95ded45538bf358ce5122a6e3ff920db25525186

#### A.2.3 Hardware dependencies

You need GPUs to train or run models.

#### A.2.4 Software dependencies

None.

#### A.2.5 Benchmarks

**Datasets.** Our source code could automatically download the datasets, i.e., CIFAR-10, CIFAR-100, STL-10. **For Tiny-ImageNet, please download it by yourself.**

**Models.** Our source code could train all models, and you can directly run the scripts to train the models.

<sup>✉</sup> Corresponding authors.

## A.3 Set-up

### A.3.1 Installation

We list the python packages and the corresponding versions to install. Note other versions may work as well, but we haven't tried it.

- (1) Please install python 3.6.9.
- (2) Please install pytorch 1.7.0.
- (3) Please install torchvision 0.8.1.
- (4) Please install tensorboardX 2.5.
- (5) Please install matplotlib 3.3.4.
- (6) Please install tqdm 4.60.0.
- (7) Please install pandas 1.1.5.
- (8) Please install numpy 1.18.5.

### A.3.2 Basic Test

## A.4 Evaluation workflow

### A.4.1 Major Claims

- (C1):** Aegis could effectively mitigate TBT attacks, and adaptive TBT attacks.
- (C2):** Aegis could effectively mitigate TA-LBF attacks, and adaptive TA-LBF attacks.
- (C3):** Aegis could effectively mitigate ProFlip attacks, and adaptive ProFlip attacks.

### A.4.2 Experiments

Before conducting experiments, you need to train all models.

- **CIFAR-10: train resnet32.**

- (1) cd cifar10/resnet32
- (2) Train the base model: sh train\_CIFAR.sh.
- (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh

- **CIFAR-10: train vgg16.**
  - (1) cd cifar10/vgg16
  - (2) Train the base model: sh train\_CIFAR.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **CIFAR-100: train resnet32.**
  - (1) cd cifar100/resnet32
  - (2) Train the base model: sh train\_CIFAR.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **CIFAR-100: train vgg16.**
  - (1) cd cifar100/vgg16
  - (2) Train the base model: sh train\_CIFAR.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **STL-10: train resnet32.**
  - (1) cd stl10/resnet32
  - (2) Train the base model: sh train\_STL.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **STL-10: train vgg16.**
  - (1) cd stl10/vgg16
  - (2) Train the base model: sh train\_STL.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **Tiny-ImageNet: train resnet32.**
  - (1) cd tinyimagenet/resnet32
  - (2) Train the base model: sh train\_tinyimagenet.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh
- **Tiny-ImageNet: train vgg16.**
  - (1) cd tinyimagenet/vgg16
  - (2) Train the base model: sh train\_tinyimagenet.sh.
  - (3) After finishing training the base model, then train the enhanced model: sh train\_finetune\_branch.sh

**(E1):** For TBT attacks.

- (1) First enter a folder to attack the target model, e.g.,  
*cd ./Aegis/TBT/resnet32-cifar10/*
- (2) If you want to conduct the TBT attack, run the instruction: *python3 TBT\_noadaptive.py*. Then, you can observe the ASR.
- (3) If you want to conduct the adaptive TBT attack, run the instruction: *python3 TBT\_adaptive.py*. Then, you can observe the ASR.

**(E2):** For non-adaptive TA-LBF attacks. Please enter the folder: cd TA-LBF/non-adaptive. For adaptive TA-LBF attacks. Please enter the folder: cd TA-LBF/adaptive.

- (1) on cifar10 and resnet32, run the instruction:  
*sh ./attack\_reproduce\_k=50\_resnet32\_cifar10.sh*
- (2) on cifar10 and vgg16, run the instruction:  
*sh ./attack\_reproduce\_k=50\_vgg16\_cifar10.sh*
- (3) on cifar100 and resnet32, run the instruction:  
*sh ./attack\_reproduce\_k=50\_resnet32\_cifar100.sh*
- (4) on cifar100 and vgg16, run the instruction:

*sh ./attack\_reproduce\_k=50\_vgg16\_cifar100.sh*

(5) on stl10 and resnet32, run the instruction:

*sh ./attack\_reproduce\_k=50\_resnet32\_stl10.sh*

(6) on stl10 and vgg16, run the instruction:

*sh ./attack\_reproduce\_k=50\_vgg16\_stl10.sh*

(7) on tinyimagenet and resnet32, run the instruction:

*sh ./attack\_reproduce\_k=50\_resnet32\_tinyimagenet.sh*

(8) on tinyimagenet and vgg16, run the instruction:

*sh ./attack\_reproduce\_k=50\_vgg16\_tinyimagenet.sh*

**(E3):** For ProFlip attacks.

(1) First enter a folder to attack the target model, e.g., *cd cd ./Aegis/ProFlip/resnet32-cifar10/*

(2) If you want to conduct the ProFlip attack, run the instruction to generate a trigger: *python3 trigger\_noadaptive.py*. Then, run the instruction to attack: *python3 CSB\_noadaptive.py*. Then, you can observe the ASR.

(3) If you want to conduct the adaptive ProFlip attack, run the instruction to generate a trigger: *python3 trigger\_adaptive.py*. Then, run the instruction to attack: *python3 CSB\_adaptive.py*. Then, you can observe the ASR.

## A.5 Notes on Reusability

None.

## A.6 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at <https://secartifacts.github.io/usenixsec2023/>.