A Artifact Appendix

A.1 Abstract

This artifact includes the code necessary to reproduce the experimental results presented in our paper titled "Transferring Adversarial Robustness Through Robust Representation Matching". It is made available in the form of a GitHub repository (final stable URL: https://github.com/Ethos-lab/robust-representation-matching/releases/tag/final). Our experiments involve training neural network based image classifiers that are robust against adversarial attacks. Therefore, we provide the necessary training and evaluation scripts, along with all the supporting code. The expected results, as reported in the paper, are : (1) total training time, and (2) accuracy of the trained classifier on clean and adversarial test sets. All our code is written in Python.

On the hardware side, the code requires a machine with at least one GPU with 12 GB memory and storage space > 150 GB to run. We recommend at least 8 GB of RAM. On the software side, the code requires Python compiler, pip, conda and several other 3rd party Python libraries like PyTorch, IBM's Adversarial Robustness Toolbox, Nvidia's apex *etc.* Detailed instructions regarding setting up the run-time environment are provided in Section A.4 and the repository README.

A.2 Artifact check-list (meta-information)

- **Algorithm:** Our paper presents a novel algorithm called Robust Representation Matching (RRM). The purpose of this algorithm is to speed up the process of adversarially training neural network based image classifiers.
- **Data set:** We perform experiments using two image datasets: CIFAR-10 and Restricted-ImageNet. The CIFAR-10 dataset downloads itself if not available. It requires 341MB storage space. For experiments involving Restricted-ImageNet, the full ImageNet dataset needs to be downloaded. Instructions for this are provided in the README of the code repository. It requires 145 GB storage space.
- **Model:** The CIFAR-10 experiments are conducted using the following neural networks: VGG11, VGG19, ResNet18, ResNet50. The Restricted-ImageNet experiments use the following neural networks: AlexNet, VGG16, ResNet50. All the code associated with these networks is provided in the repository. We also make available weights of pre-trained classifiers for quick evaluation.
- **Run-time environment:** Our code has been tested on a Linux machine. To prepare the run-time environment, one needs to create a Python virtual environment and install all required Python libraries. The instructions for setting up the run-time environment are provided in Section A.4 and the README in the repository.
- **Hardware:** The code requires a machine with at least one GPU with 12 GB memory and storage space > 150 GB. We recommend running the Restricted-ImageNet training scripts on 4 GPUs. Also, we recommend 8 GB of RAM.
- **Execution:** Here we provide estimated time taken by different components of our experiments. These estimates were computed on our machine. We ran our experiments on two different machines. The CIFAR-10 experiments were run on a machine with an Intel Xenon(R) Gold 6136 CPU, 16 GB RAM, and an Nvidia Titan V GPU. The training scripts took $\sim$ 5 hours on average. In total 19 classifiers need to be trained using different methods. The Restricted-ImageNet experiments were run on a second machine with an Intel Xenon(R) E5-2690 CPU, 16 GB RAM, and an Nvidia V100 GPU. The training scripts took $\sim$ 1 week to run on average. In total 5 classifiers need to be trained using different methods. For both the datasets, the evaluation scripts take $\sim$ 3 hours to run in the worst case, with every trained model needing to be evaluated once.
- **Metrics:** We report two metrics in our paper: (1) Training run time and (2) Accuracy of clean and adversarial test sets. Note that due to differences in hardware, the absolute training times will be different than what is reported in the main paper. However, the speedup (ratio of train times) should be approximately the same.
- **Output:** All the expected output will be printed out as stdout on running the evaluation script. The following quantities will be outputted: (1) average time per training epoch, (2) its 95% confidence interval, (3) total training time, (4) accuracy on clean test set, and (5) accuracy on adversarial test set.
- **Experiments:** The step-by-step instructions to reproduce the experimental results are provided in the GitHub READMEs. The accuracy numbers will be within a few percentage points of the numbers reported in the main paper. The absolute training time numbers will vary from what is reported in the main paper due to hardware differences. However, the speedup numbers (ratio of training times) will be approximately the same.
- **How much disk space required (approximately)?:** 150 GB
- **How much time is needed to complete experiments (approximately)?:** On our machine, training all the reported classifiers on CIFAR-10 took $\sim$ 4 days. Training all the reported Restricted-ImageNet classifiers took $\sim$ 5 weeks. Evaluating all the classifiers (corresponding to both datasets) took $\sim$ 3 days. During reproduction, expect significant variations in these times because of hardware differences.
- **Publicly available (explicitly provide evolving version reference)?:** Yes. https://github.com/Ethos-lab/robust-representation-matching
- **Code licenses (if publicly available)?:** MIT License
- **Data licenses (if publicly available)?:** CIFAR-10: no license. ImageNet: https://www.image-net.org/download.php.

A.3 Description

A.3.1 How to access

Clone GitHub repository, available here (final stable URL): https://github.com/Ethos-lab/robust-representation-matching/releases/tag/final

A.3.2 Hardware dependencies

The code requires a machine with at least one GPU with 12 GB memory and storage space > 150 GB. We recommend running the Restricted-ImageNet training scripts on 4 GPUs. Also, we recommend 8 GB of RAM.

A.3.3 Software dependencies

Our code is written in Python and requires a Python compiler installed along with the python package managers pip and conda. In addition, our code makes use of several 3rd party Python libraries. For instructions regarding how to install all the software dependencies and set up the run-time environment, refer to Section A.4 and the GitHub README.

A.3.4 Data sets

We use two datasets in our experiments: CIFAR-10 and Restricted-ImageNet. CIFAR-10 will download itself if not available. For Restricted-ImageNet, the entire ImageNet dataset needs to be downloaded. Instructions for this are provided in the GitHub README.

A.3.5 Models

The CIFAR-10 experiments are conducted using the following neural networks: VGG11, VGG19, ResNet18, ResNet50. The Restricted-ImageNet experiments use the following neural networks: AlexNet, VGG16, ResNet50. All the code associated with these networks is provided in the repository. We also make available weights of pre-trained classifiers for quick evaluation.

A.3.6 Security, privacy, and ethical concerns

All the data we use is publicly available for research. The work presented in our paper introduces no security, privacy, or ethical concerns.

A.4 Installation

Follow the following steps to set up the run-time enviroment required to run our code:

1. Clone the github repository and navigate into it:
   ```
   git clone https://github.com/pratik18v/
   robust-representation-matching.git &&
   cd robust-representation-matching
   ```
2. Create a Python virtual environment and activate it:
   ```
   conda create -n rrm python=3.6 &&
   conda activate rrm
   ```
3. Install dependencies:
   ```
   pip install -r requirements.txt
   ```
4. Install apex using instructions here:
   https://github.com/NVIDIA/apex#quick-start

All the instructions to setup the run-time enviroment are also provided in the GitHub README.

A.5 Evaluation and expected results

We demonstrate that our proposed algorithm (RRM) trains adversarially robust image classifiers faster than previous state-of-the-art method, at the same time attaining better robustness. For this, we train neural networks using several prior methods and compare them to our method. We perform comparison using two metrics: (1) total training time, and (2) accuracy on clean and adversarial test sets. We show that our method has the lowest total training time. Compared to the previous fastest method, our method trains classifier with higher adversarial accuracy. The accuracy numbers can be reproduced within a few percentage points of the numbers reported in the main paper. The absolute training time numbers will vary from what is reported in the main paper due to hardware differences. However, the speedup numbers (ratio of training times) can be reproduced to a value approximately similar to the reported value. The detailed steps to reproduce our results are laid out in the READMEs available in our GitHub repository.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20220119.