Appendix for: LiBRe: A Practical Bayesian Approach to Adversarial Detection

Zhijie Deng[1], Xiao Yang[1], Shizhen Xu[2], Hang Su[1*], Jun Zhu[1*]
[1] Dept. of Comp. Sci. and Tech., BNRist Center, Institute for AI, Tsinghua-Bosch Joint ML Center, THBI Lab
[1] Tsinghua University, Beijing, 100084, China   [2] RealAI
{dzj17,yangxiao19}@mails.tsinghua.edu.cn, shizhen.xu@realai.ai, {suhangss,dcszj}@tsinghua.edu.cn

A. Attack Methods

In this part, we outline the details of the adopted attack methods in this paper. For simplicity, we use $l(\boldsymbol{x}, y)$ to notate the loss function for attack which inherently connects to the negative log data likelihood, e.g., the cross entropy in image classification, the pairwise feature distance in open-set face recognition, and the weighted sum of the bounding-box regression loss and the classification loss in object detection.

FGSM [5] crafts an adversarial example under the ℓ_∞ norm as

$$\boldsymbol{x}^{\text{adv}} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, y)), \tag{1}$$

FGSM can be extended to an ℓ_2 attack as

$$\boldsymbol{x}^{\text{adv}} = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, y)\|_2}. \tag{2}$$

In all experiments, we set the perturbation budget ϵ as $16/255$.

BIM [7] extends FGSM by taking iterative gradient updates:

$$\boldsymbol{x}^{\text{adv}}_{t+1} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}^{\text{adv}}_t + \eta \cdot \text{sign}(\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^{\text{adv}}_t, y))\big), \tag{3}$$

where $\text{clip}_{\boldsymbol{x},\epsilon}$ guarantees the adversarial example to satisfy the ℓ_∞ constraint. For all the iterative attack methods, we set the number of iterations as 20 and the step size η as $1/255$.

PGD [8] complements BIM with a random initialization for the adversarial examples (i.e., $\boldsymbol{x}^{\text{adv}}_0$ is uniformly sampled from the neighborhood of $\boldsymbol{x}$).

MIM [2] introduces a momentum term into BIM as

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^{\text{adv}}_t, y)}{\|\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^{\text{adv}}_t, y)\|_1}, \tag{4}$$

where μ refers to the decay factor and is set as 1 in all experiments. Then, the adversarial example is calculated by

$$\boldsymbol{x}^{\text{adv}}_{t+1} = \text{clip}_{\boldsymbol{x},\epsilon}(\boldsymbol{x}^{\text{adv}}_t + \eta \cdot \text{sign}(\boldsymbol{g}_{t+1})). \tag{5}$$

We adopt the same hyper-parameters as in BIM.

DIM [10] relies on a stochastic transformation function to craft adversarial examples, which can be represented as

$$\boldsymbol{x}^{\text{adv}}_{t+1} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}^{\text{adv}}_t + \eta \cdot \text{sign}(\nabla_{\boldsymbol{x}} l(T(\boldsymbol{x}^{\text{adv}}_t; p), y))\big), \tag{6}$$

where $T(\boldsymbol{x}^{\text{adv}}_t; p)$ refers to some transformation to diversify the input with probability p.

TIM [3] integrates the translation-invariant method into BIM by convolving the gradient with the pre-defined kernel $\boldsymbol{W}$ as

$$\boldsymbol{x}^{\text{adv}}_{t+1} = \text{clip}_{\boldsymbol{x},\epsilon}\big(\boldsymbol{x}^{\text{adv}}_t + \eta \cdot \text{sign}(\boldsymbol{W} * \nabla_{\boldsymbol{x}} l(\boldsymbol{x}^{\text{adv}}_t, y))\big). \tag{7}$$

C&W adopts the original C&W loss [1] based on the iterative mechanism of BIM to perform attack in classification tasks. In particular, the loss takes the form of

$$l_{cw} = \max(Z(\boldsymbol{x}^{\text{adv}}_t)_y - \max_{i \neq y} Z(\boldsymbol{x}^{\text{adv}}_t)_i, 0), \tag{8}$$

where $Z(\boldsymbol{x}^{\text{adv}}_t)$ is the logit output of the classifier.

SPSA [9] estimates the gradients by

$$\hat{\boldsymbol{g}} = \frac{1}{q} \sum_{i=1}^{q} \frac{l(\boldsymbol{x} + \sigma \boldsymbol{u}_i, y) - l(\boldsymbol{x} - \sigma \boldsymbol{u}_i, y)}{2\sigma} \cdot \boldsymbol{u}_i, \tag{9}$$

where $\{\boldsymbol{u}_i\}_{i=1}^q$ are samples from a Rademacher distribution, and we set $\sigma = 0.001$ and $q = 64$. Besides, $l(\boldsymbol{x}, y) = Z(\boldsymbol{x})_y - \max_{i \neq y} Z(\boldsymbol{x})_i$ is used in our experiments rather than the cross entropy loss. We take an Adam [6] optimizer with 0.01 learning rate to apply the estimated gradients.

B. More Experiment Details

For ℓ_2 threat model, we adopt the normalized ℓ_2 distance $\bar{\ell}_2(\boldsymbol{a}) = \frac{\|\boldsymbol{a}\|_2}{\sqrt{d}}$ as the measurement, where d is the dimension of a vector $\boldsymbol{a}$. The decay factors of MIM, TIM, and DIM are 1.0.

In ImageNet classification, we apply Gaussian blur upon the sampled uniform noise with 0.03 probability, and then

[*] Corresponding author

use the outcome to perturb the training data. The technique can enrich the training perturbations with low-frequency patterns, promoting the adversarial detection sensitiveness against diverse kinds of adversarial perturbations.

To attack the open-set face recognition system in the evaluation phase, we find every face pair belonging to the same person, and use one of the paired faces as $\boldsymbol{x}$ and the feature of the other as y to perform attack. The loss function for such an attack is the ℓ_2 distance between y and the feature of $\boldsymbol{x}$ (as mentioned in Sec A). As the *posterior predictive* is not useful in such an open-set scenario, we perform *Bayes ensemble* on the output features and then leverage the outcomes to make decision. Due to the limited GPU memory, we attack the deterministic features of the *MC dropout* baseline instead of the features from *Bayes ensemble*, while the uncertainty estimates are still estimated based on 20 stochastic forward passes with dropout enabled.

In object detection, we adopt the YOLOV5-s architecture, there are three feature output heads (`BottleneckCSP` modules) to deliver features in various scales. Thus, we make these three heads be Bayesian when implementing *LiBRe*. During inference, we average the features calculated given different parameter candidates to obtain an assembled feature to detect objects, which assists us to bypass the potential difficulties of directly assembling the object detection results.

C. Generalization to Score-based Attack

We additionally concern whether *LiBRe* can generalize to the adversarial examples generated by score-based attacks, which usually present different characteristics from the gradient-based ones. We leverage the typical SPSA [9] to conduct experiments on ImageNet, getting 0.969 detection AUROC. This further evidences our *attack agnostic* designs.

D. Detect More Ideal Attacks

At last, we evaluate the adversarial detection ability of *LiBRe* on more ideal attacks. We add the constraint that the generated adversarial examples should also have small predictive uncertainty into the existing attacks. This means that the attacks can jointly fool the decision making and uncertainty quantification aspects of the model. We add an uncertainty minimization term upon the original attack objective to implement this. We feed the crafted adversarial examples into *LiBRe* to assess if they can be identified. On ImageNet, we obtain the following adversarial detection AUROCs: 0.9996, 0.2374, 0.0363, 0.2211, 0.1627, 0.1990, 0.9998, 0.9627, 0.2537, and 0.2213 for FGSM, BIM, C&W, PGD, MIM, TIM, DIM, FGSM-ℓ_2, BIM-ℓ_2, and PGD-ℓ_2, respectively.

The results reveal that *LiBRe* is likely to be defeated if being fully exposed to the attackers. But it is also no doubt that *LiBRe* is powerful enough if keeping opaque to the attack algorithms as the pioneering work [4]. We believe that introducing adversarial training mechanism into *LiBRe* would significantly boost the ability of detecting these ideal attacks, and we leave it as future work.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017. 1

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[4] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 2

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1

[8] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[9] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning (ICML)*, 2018. 1, 2

[10] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1