Be Your Own Neighborhood: Detecting Adversarial Example by the Neighborhood Relations Built on Self-Supervised Learning

Zhiyuan He[1]*, Yijun Yang[1]*, Pin-Yu Chen[2], Qiang Xu[1], Tsung-Yi Ho[1]
[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] IBM Research
{zyhe, yjyang, qxu, tyho}@cse.cuhk.edu.hk, pin-yu.chen@ibm.com

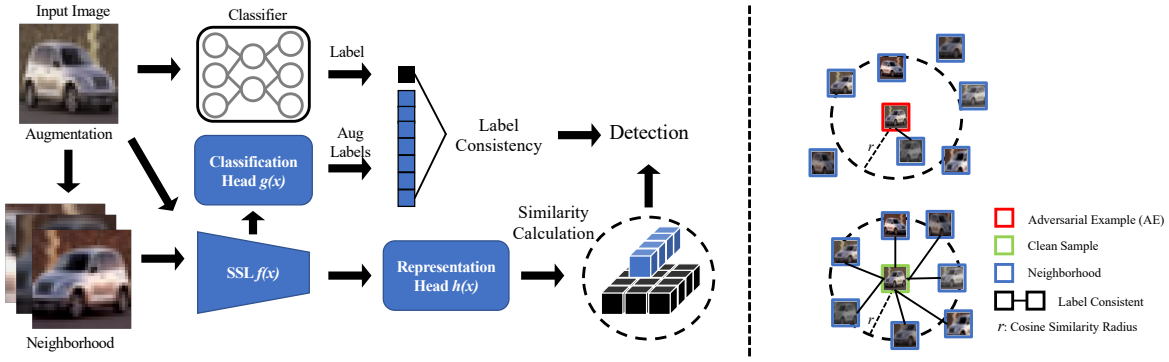


Figure 1. Pipeline of the proposed **BEYOND** framework. First we augment the input image to obtain a bunch of its neighbors. Then, we perform the label consistency detection mechanism on the classifier's prediction on the input image and that of neighbors predicted by SSL's classification head. Meanwhile, the representation similarity mechanism employs *cosine distance* to measure the similarity among the input image and its neighbors (left). The input image with poor label consistency or representation similarity is flagged as AE (right).

Abstract

Deep Neural Networks (DNNs) have achieved excellent performance in various fields. However, DNNs' vulnerability to Adversarial Examples (AE) hinders their deployments to safety-critical applications, e.g., autonomous driving. This paper presents a novel AE detection framework, named **BEYOND**, *for trustworthy predictions.* BEYOND *performs the detection by distinguishing AE's abnormal relation with its augmented versions, i.e. neighbors, from two prospects: representation similarity and label consistency. An off-the-shelf Self-Supervised Learning (SSL) model is used to extract the representation and predict the label for its highly informative representation capacity compared to supervised learning models. For clean samples, their representations and predictions are closely consistent with their neighbors, whereas those of AEs differ greatly. Moreover, we explain this observation and show that by leveraging this discrepancy* BEYOND *can effectively detect AEs. Experiments show that* BEYOND *outperforms baselines by a large margin, especially under adaptive attacks.*

*Zhiyuan He and Yijun Yang contribute equally to this work.

1. Introduction

Deep Neural Networks (DNNs) have been widely adopted in many fields such as computer vision, speech recognition, and natural language processing due to their superior performance. However, DNNs are vulnerable to Adversarial Examples (AEs), which can easily fool DNNs by adding some imperceptible adversarial perturbations. Such vulnerability precludes DNNs from being deployed to safety-critical applications such as autonomous driving and disease diagnosis, where incorrect predictions can lead to catastrophic economic and even loss of life.

Existing defensive countermeasures can be roughly categorized as: adversarial training, input purification, and AE detection [1, 4, 8]. Adversarial training is known as the most effective defense technique [4], but it brings accuracy degradation and extra training cost, which are unacceptable under some application scenarios. By contrast, input transformation techniques without high training/deployed costs, but their defensive ability is limited, i.e easily being defeated by adaptive attacks [4].

Recently, a large number of AE detection methods have been proposed. Some methods detect AE by interrogating

the abnormal relationship between AE and other samples. For example, Deep k-Nearest Neighbors (DkNN) [8] compares the DNN-extracted features of the input image with that of its k nearest neighbors layer by layer to identify AE, leading to high inference cost. Instead of comparing all the features, Latent Neighborhood Graph (LNG) [1] employs a Graph Neural Network to make the comparison on a neighborhood graph, whose nodes are pre-stored embeddings of AEs together with those of the clean ones extracted by a DNN, and the edges are built according to distances between the input node and every reference node. Though more efficient than DkNN, LNG suffers from some weaknesses: some AEs are required to build the graph the detection performance relays on the reference AEs and cannot effectively generalize to unseen attacks. More importantly, both DkNN and LNG can be bypassed by adaptive attacks, in which the adversary knows full knowledge of the detection strategy.

We observe that one cause for the adversarial vulnerability is the lack of feature invariance, i.e. small perturbations may lead to undesired large changes in features or even predicted labels. Self-Supervised Learning (SSL) models learn data representation consistency under different data augmentations, which intuitively can mitigate the issue of lacking feature invariance and thereby improve adversarial robustness [1]. As an illustration, we visualize the SSL-extracted representation of the clean sample, AE and that of their corresponding augmentations in Fig. 1 (right). We can observe that the clean sample has closer ties with its neighbors reflected by the higher label consistency and representation similarity. Whereas AE's representation stays quite far away from its neighbors, and there is a wide divergence in the predicted labels.

Inspired by the above observations, we propose an novel AE detection framework, named **BE Your Own NeighborhooD (BEYOND)**. The contributions of this work are summarized as follows:

- We propose BEYOND, a novel AE detection framework, which takes advantage of SSL model's robust representation capacity to identify AE by referring to its neighbors. To our best knowledge, BEYOND is the first work that leverages SSL model for AE detection without prior knowledge of adversarial attacks or AEs.
- We show that BEYOND can effectively defend against adaptive attacks. To defeat the two adopted detection mechanisms: label consistency and representation similarity simultaneously, attackers have to optimize two objectives that have contradictory directions, resulting in gradients canceling each other out.
- As a plug-and-play method, BEYOND can be applied directly to any image classifier without compromising accuracy or additional retraining costs.

2. Proposed Method

2.1. Method Overview

Compontens. BEYOND consists of three components: a SSL feature exacter $f(\cdot)$, a classification head $g(\cdot)$, and a representation head $h(\cdot)$, as shown in Fig. 1 (left). To be more specific, the SSL feature exacter is a Convolutional Neural Network (CNN), pretrained by specially-designed loss, e.g., contrastive loss, without supervision [*]. A Fully-Connected (FC) layer acts as the classification head $g(\cdot)$, trained by freezing the $f(\cdot)$. The $g(\cdot)$ performs on input image's neighbors for label consistency detection. The representation head $h(\cdot)$ consisting three FC layer, encodes the output of $f(\cdot)$ to a embedding, i.e. representation. We operate the representation similarity detection on representations of the input image and its neighbors.

Core idea. Our approach relies on robust relationships between input sample and its neighbors for AE detection. The key idea is that adversarial attacks may easily attack one sample's representation to another submanifold, but it is difficult to totally shift that of all its neighbors. We employ the SSL model to capture such relationships, since it is trained to project input and its augmentations (neighbor) to the same sub-manifold [3].

Selection of neighbor number. Obviously, the larger the number of neighbors, the more stable relationship between them, but may increase the overhead. We choose 50 neighbors for BEYOND, since larger neighbors no longer enhance the performance significantly , as shown in Fig. 2.

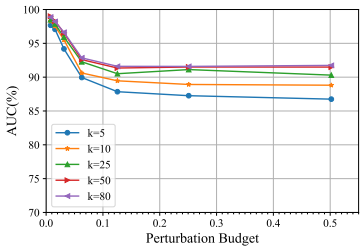


Figure 2. Detection performance with different number of neighbors on CIFAR-10.

Workflow. Fig. 1 demonstrates the workflow of the proposed BEYOND. When input comes, we first transform it to 50 augmentations, i.e. 50 neighbors. Next, the input along with its 50 neighbors are fed to SSL feature extractor $f(\cdot)$ thereafter the classification head $g(\cdot)$ and representation head $h(\cdot)$, respectively. For the classification branch, $g(\cdot)$ outputs the predicted label for 50 neighbors. Later, the label consistency detection algorithm calculates the consistency level between the input label (predicted by the classifier) and 50 labels of neighbors. When it comes to the representation branch, the generated 51 representations are

[*]Here, we employ the SimSiam [3] as the SSL feature-exactor for its decent performance.

sent to representation similarity detection algorithm for AE detection. If a sample's label consistency or representation similarity is lower than a threshold, BEYOND shall flag AE.

2.2. Detection Algorithms

For enhanced AE detection capability, BEYOND adopted two detection mechanisms: *Label Consistency*, and *Representation Similarity*. The detection performance of the combined two can exceed any of the individuals. More importantly, their contradictory optimization directions hinder the adaptive attacks to bypass both of them, simultaneously.

Label Consistency. We compare the classifier's prediction, $\ell_{cls}(x)$, on input image, x, to SSL classification head's predictions, $\ell_{ssl}(\hat{x}_i), i = 1 \dots k$, where $\hat{x}_i$ denotes the ith neighbor, k is the total number of neighbors. If $\ell_{cls}(x)$ equals $\ell_{ssl_i}(\hat{x}_i)$, the label consistency increases by one, $\texttt{Ind}_{\texttt{Label}} += 1$. Once the final label consistency less than a certain threshold, $\texttt{Ind}_{\texttt{Label}} < \mathcal{T}_{label}$, the *Label Consistency* flags it as AE. We summarize the label consistency detection mechanism in Algorithm. 1.

Representation Similarity. We employ the *cosine distance* as a metric to calculate the similarity between the representation of input sample, $r(x)$ and that of its neighbors, $r(\hat{x}_i), i = 1, ..., k$. Once the similarity, $-cos(r(x), r(\hat{x}_i))$, is higher than a certain value, representation similarity increases by 1, $\texttt{Ind}_{\texttt{Rep}} += 1$. If the final representation similarity less than a threshold, $\texttt{Ind}_{\texttt{ReP}} < \mathcal{T}_{rep}$, the *representation similarity* flag the sample as an AE. Algorithm. 1 concludes the representation similarity detection mechanism.

Note that, we select the thresholds, i.e. $\mathcal{T}_{label}$, $\mathcal{T}_{rep}$, by fix the False Positive Rate (FPR)@5%, which can be determined only by clean sample, and the implementation of our method needs no prior knowledge about AE.

2.3. Resistance to Adaptive Attacks

Attackers may design adaptive attacks to bypass BEYOND, if the attacker knows both the classifier and the detection strategy. BEYOND apply the augmentation on the input, which has a weakening effect on adversarial perturbations. As a result, to mislassified SSL's classification results on neighbors,i.e. bypass label consistency detection, large perturbations are needed. However, to bypass the representation similarity detection, the added perturbation should be small, since a large perturbation can alter the representation significantly. Therefore, to attack BEYOND, attackers have to optimize two objectives that have contradictory directions, resulting in gradients canceling each other out.

3. Evaluation

3.1. Experimental Setting

Gray-box attack & White-box attack. In the grey-box attack setting, the adversary has complete knowledge of the classifier, while the detection strategy is confidential. Whereas in an adaptive attack (white-box) setting, the adversary is aware of the defense strategy.

Algorithm 1 BEYOND detection algorithm

Input: Input image x, target classifier $c(\cdot)$, SSL feature exactor $f(x)$, classification head $g(x)$, projector head $h(x)$, label consistency threshold $\mathcal{T}_{label}$, representation similarity threshold $\mathcal{T}_{rep}$, Augmentation Aug, neighbor indicator i, total neighbor k
Output: `reject / accept`

Stage1: Collect labels and representations.
$\ell_{cls}(x) = c(x)$
for i in k **do**
  $\hat{x}_i = Aug(x)$
  $\ell_{ssl}(\hat{x}_i) = f(g(\hat{x}_i))$;$r(x) = f(h(x))$; $r(\hat{x}_i) = f(h(\hat{x}_i))$
Stage2: Label consistency detection mechanism.
for i in k **do**
  if $\ell(\hat{x}_i) == \ell(x)$ **then** $\texttt{Ind}_{\texttt{label}} += 1$
Stage3: Representation similarity detection mechanism.
for i in k **do**
if $cos(r(x), r(\hat{x}_i)) < \mathcal{T}_{cos}$ **then** $\texttt{Ind}_{rep} += 1$
Stage4: AE detection.
if $\texttt{Ind}_{label} < \mathcal{T}_{label}$ or $\texttt{Ind}_{rep} < \mathcal{T}_{rep}$ **then** `reject`
else `accept`

Table 1. Information of Datasets and Models.

Dataset	Classifier SSL	Accuracy↑	
		Classifier	SSL
CIFAR-10	ResNet18	91.53%	90.74%
CIFAR-100	ResNet18	75.34%	66.04%
IMAGENET	ResNet50	80.86%	68.30%

Datasets & Target models. We conduct experiments on three popular datasets: CIFAR-10, CIFAR-100, and IMAGENET. The details of the target models (classifiers), and the SSL models along with their original classification accuracy on clean samples are summarized in Tab. 1.

Attacks. Evaluations are conducted on FGSM, PGD, CW, and AutoAttack [4]. AutoAttack includes APGD, APGD-T, FAB-T, and Square, where APGD-T and FAB-T are targeted attacks, and Square is a black-box attack.

Metrics. TPR@FPR@n%: TPR@FPR indicates the true positive rate (TPR) at a false positive rate (FPR) $\leq n\%$, which reflects the detection ability while ensuring the classification precision of clean samples. **ROC curve & AUC:** ROC curves describe the impact of various thresholds on detection performance, and the AUC is an overall metric.

Baselines. We choose five detection-based defense methods as baselines: kNN [5], DkNN [8], LID [7], [6] and LNG [1], which also consider the relationship among the input and its neighbors to some extent.

3.2. Detection Performance

Tab. 3 reports TPR@FPR5% to demonstrate BEYOND's AE detection performance. It can be seen that BEYOND maintains a high detection performance in various attacks

Table 2. The AUC of Different Adversarial Detection Approaches on CIFAR-10. The **bolded** values are the best performance. To align with baselines, classifier: ResNet110, FGSM: $\epsilon = 0.05$, PGD: $\epsilon = 0.02$. Note that **BEYOND needs no AE for training**, leading to the same value on both *seen* and *unseen* settings.

Methods	*Unseen*: Attacks used in training are preclude from test.				*Seen*: Attacks used in training are included in test.				
	FGSM	PGD	AutoAttack	Square	FGSM	PGD	CW	AutoAttack	Square
DkNN	61.50%	51.18%	52.11%	59.51%	61.50%	51.18%	61.46%	52.11%	59.21%
kNN	61.80%	54.46%	52.64%	73.39%	61.80%	54.46%	62.25%	52.64%	73.39%
LID	71.15%	61.27%	55.57%	66.11%	73.56%	67.95%	55.60%	56.25%	85.93%
Hu	84.44%	58.55%	53.54%	95.83%	84.44%	58.55%	90.99%	53.54%	95.83%
LNG	98.51%	63.14%	58.47%	94.71%	**99.88%**	91.39%	89.74%	84.03%	98.82%
BEYOND	**98.89%**	**99.29%**	**99.18%**	**99.29%**	98.89%	**99.29%**	**99.20%**	**99.18%**	**99.29%**

Table 3. TPR@FPR 5% of BEYOND against Gray-box Attack. All attacks have a perturbation budget of an $L_\infty = 8/255$.

Dataset	CIFAR-10	CIFAR-100	IMAGENET
Attack	TPR@FPR5%↑		
FGSM	86.16%	89.80%	61.05%
PGD	82.80%	85.90%	89.80%
CW	91.48%	91.96%	76.69%
APGD	83.70%	85.50%	90.70%
APGD-T	98.40%	94.80%	90.70%
FAB-T	97.00%	92.20%	80.60%
Square	94.57%	91.10%	75.00%

and datasets. Tab. 2 compares the AUC of BEYOND with five baselines on CIFAR-10. Since LID and LNG rely on AEs for reference during training, we report detection performance on both *seen* and *unseen* attacks. In the *seen* setting, LID and LNG are trained with all types of attacks, while using only the CW attack in the *unseen* attack setting. Note that the detection performance for BEYOND is consistent, since BEYOND needs no AE for training. Experimental results show that BEYOND consistently outperforms the SOTA AE detection methods on CIFAR-10, and the performance advantage is significant when under the *unseen* setting.

3.3. Adaptive Attacks

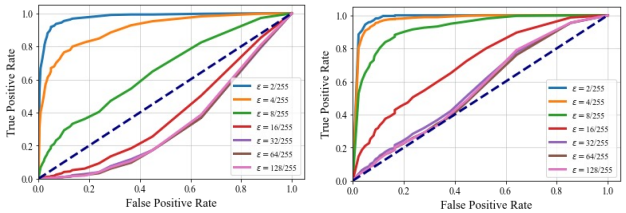
The Design of Adaptive Attack. To adaptively attack BEYOND, the adversary needs to deceive the target model while guaranteeing the label consistency and representation similarity. Note that BEYOND is not based on random transformations. For multiple augmentations employed in BEYOND, we estimate their impact on label consistency and representation similarity during the adaptive attack following Expectation over Transformation (EoT) [2] as:

$$\begin{aligned} Sim_l &= \frac{1}{k}\sum_{i=1}^{k}\mathcal{L}\left(f(g\left(W_{aug}^{i}(x+\delta)\right)), y_t\right), \\ Sim_r &= \frac{1}{k}\sum_{i=1}^{k}(\mathcal{S}(f(h(W_{aug}^{i}(x+\delta))), f(h(x+\delta)))), \end{aligned} \tag{1}$$

where $\mathcal{S}$ represents the cosine similarity, W_{aug} represents data augmentations, and the adaptive adversaries perform gradient descent on the following combined objective:

$$\min_{\delta} \mathcal{L}_C(x+\delta, y_t) + Sim_l - Sim_r, \tag{2}$$

where $\mathcal{L}_C$ indicates classifier's loss function, and y_t is the targeted class.



(a) CIFAR-10 (b) CIFAR-100

Figure 3. ROC Curve of Perturbation Budgets.

ROC v.s. Perturbation Budgets. Fig. 3 summarizes the ROC curve varying with different perturbation budgets on CIFAR-10 and CIFAR-100. Our analysis regarding Fig. 3 is as follows: 1) BEYOND can be bypassed when perturbations are large enough, which is caused by the large perturbation circumventing the input transformation. This shows that BEYOND is not gradient masking and our adaptive attack design is effective. However, large perturbations, while bypassing BEYOND, are easier to perceive. 2) When the perturbation is small, the detection performance of BEYOND for adaptive attacks remains high. This is because small perturbations cannot guarantee both label consistency and representation similarity. 3) Under the same perturbation budget, the performance of adaptive attack on CIFAR-100 is weaker than CIFAR-10, which is because the complex label space of CIFAR-100 makes the optimization of label consistency more difficult than CIFAR-10.

4. Conclusion

We propose BEYOND, a novel detection framework, which focuses on identifying abnormal relations between AEs and their augmented neighbors. Samples have low label consistency and representation similarity with their neighbors is detected as AE. We empirically demonstrate the effectiveness of BEYOND through grey-box and adaptive attacks. Experimental results show that BEYOND outperforms SOTA AE detectors.

References

[1] Ahmed Abusnaina, Yuhang Wu, Sunpreet Arora, Yizhen Wang, Fei Wang, Hao Yang, and David Mohaisen. Adversarial example detection using latent neighborhood graph. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7687–7696, 2021. 1, 2, 3

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 4

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2

[4] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 1, 3

[5] Abhimanyu Dubey, Laurens van der Maaten, I. Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. Defense against adversarial images using web-scale nearest-neighbor search. *computer vision and pattern recognition*, 2019. 3

[6] Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q. Weinberger. A new defense against adversarial images: Turning a weakness into a strength. *neural information processing systems*, 2019. 3

[7] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018. 3

[8] Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018. 1, 2, 3