Supplementary Material of Trust, but Verify: Using Self-Supervised Probing to Improve Trustworthiness

Ailin Deng, Shen Li, Miao Xiong, Zhirui Chen, and Bryan Hooi

National University of Singapore
{ailin, shen.li, miao.xiong, zhiruichen}@u.nus.edu
bhooi@comp.nus.edu.sg

A Experimental Setup

We implement the based models and self-supervised probing framework in Pytorch (GPU) 1.10.1+cu113 and train them on a server with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz and a GeForce RTX 3090 graphics card. The implementation is based on the publicly released codes[1].

B Dataset

We conduct experiments on the benchmark image datasets: CIFAR-10 [11], CINIC-10 [3] and STL-10 [1]. We use the default validation split from CINIC-10 and split 20% data from the labeled training data as the validation sets for CIFAR-10 and STL-10, respectively. All the models and baselines share the same dataset setting for fair comparison.

- CIFAR-10 [11]: CIFAR-10 consists of 50000 training images and 10000 testing images at a resolution of 32×32. The dataset covers ten object classes, with each class having an equal number of images. We split 10000 images out of the training images as the validation set.
- CINIC-10 [3]: CINIC-10 dataset is designed to be a middle option relative to CIFAR-10 and ImageNet [4]: it contains images at a resolution of 32×32 as CIFAR10 but at a large scale total of 270000 images, which is closer to that of ImageNet. The dataset has default data splits: equal 90000 images for training, validation and test set.
- STL-10: STL-10 is an image dataset derived from ImageNet with a resolution of 96×96. It contains 100000 unlabeled images and 13000 labeled images from 10 object classes. Among the labeled images, 5000 images are partitioned for training while the remaining 8000 images are for testing. We split 1000 images for each class from the training images as validation data.

[1] https://github.com/google/TrustScore
https://github.com/valeoai/ConfidNet
https://github.com/markus93/NN_calibration

C Classification Network Architecture

For classification, we adopted the architectures VGG16 [10] and ResNet-18 [6]. The specific architectures we used are publicly released[2]. To enable comparing with the MCdropout baseline, we add dropout after each block in the ResNet models. The VGG16 models are trained using the Adam optimizer with learning rate 1×10^{-4} and $(\beta_1, \beta_2) = (0.9, 0.99)$. As the adopted public VGG16 models can not support the images with 96×96 resolution directly, we only train VGG16 for CIFAR-10 and CINIC-10. The ResNet models are trained using the SGD optimizer with cosine annealing scheduler beginning with learning rate of 0.1. The models for CIFAR-10 and CINIC-10 are trained for 300 epochs and the model for STL-10 are trained for 100 epochs as training with the dataset at a smaller scale can reach convergence faster. We choose the model at the final epoch as our base trained classification model. The test accuracies for the trained model are reported in the Table 1.

Table 1: Test accuracies (%) for each trained classification model.

	CIFAR-10 VGG16	**CIFAR-10** ResNet-18	**CINIC-10** VGG16	**CINIC-10** ResNet-18	**STL-10** ResNet-18
Test Accuracy (%)	92.02	93.34	82.10	84.75	69.30

D Evaluation

D.1 Misclassification Detection

FPR at 95% TPR measures the False Positive Rate (FPR) when the True Positive Rate (FPR) is equal to 95%. True Positive Rate is computed by $\texttt{TPR} = \texttt{TP}/(\texttt{TP} + \texttt{FN})$, where `TP` and `FN` denote the occurrences of true positives and false negatives, respectively. The False Positive Rate can be computed by $\texttt{FPR} = \texttt{FP}/(\texttt{FP} + \texttt{TN})$, where `FP` and `TN` denote the occurrences of false positives and true negatives, respectively. One can interpret the metric as the probability that a sample is misclassified when the True Positive Rate (TPR) is equal to 95%.

AUROC measures the Area Under the Receiver Operating Characteristic curve (AUROC). It is a threshold-agnostic performance evaluation metric, as the curve shows the trade-off between TPR and FPR across different decision thresholds.

[2] https://github.com/valeoai/ConfidNet/blob/master/confidnet/models/vgg16.py
https://github.com/weiaicunzai/pytorch-cifar100/blob/master/models/resnet.py

AUPR measures the Area Under the Precision-Recall (PR) curve. The PR curve is a graph showing precision = TP/(TP + FP) versus recall = TP/(TP + FN) across different decision thresholds. Similar to AUROC, AUPR is also a threshold-agnostic performance evaluation. In our tests, AUPR-SUC indicates that correct predictions are used as the positive class, while AUPR-ERR indicates that errors are used as the positive class.

For we train our probing framework only on train set and select hyperparameter λ on validation set, our baselines are also implemented on train data except that we train TCP for CIFAR10 with validation set as TCP [2] relies on a larger absolute number of errors while our obtained classifiers can achieve nearly 100% accuracy on a training set and get less errors compared to using validation set.

D.2 OOD

For Out-of-Distribution Detection (OOD), we use SVHN [8], ImageNet [4] and LSUN [12] -related datasets as OOD datasets and use CIFRA-10 and CINIC-10 as the normal datasets. We use SVHN from the Pytorch library; and all the other datasets are publicly released[3]. As our goal is to verify that the self-supervised probing can benefit the existing commonly used OOD methods, we compare with the Maximum Softmax Probability (MSP) and the entropy. We adopt AUROC as our general OOD performance evaluation metric.

D.3 Calibration

ECE measures Expected Calibration Error. We partition predictions into M equally spaced bins and compute the accuracy for each bin. ECE is the average of the bins' accuracy/confidence difference. We use $M = 15$ in our experiments.

MCE measures Maximum Calibration Error. Unlike ECE, MCE is the largest calibration error across all bins. It measures the worse-case deviation between confidence and accuracy.

NLL is Negative Log Likelihood and a standard measure of a probabilistic model's quality [5]. It is also referred to as the cross entropy loss [7].

Brier Score is equivalent to the mean squared error when applied to predicted probabilities for unidimensional predictions. For multi-class classification, it can be defined as:

$$BS = \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{R} \left(f_{ti} - o_{ti} \right)^2 , \tag{1}$$

where R is the number of possible classes and N denotes the total number of instances across all classes. f_{ti} is the predictive probability of class i and o_{ti} is 1 if the example t has a ground-truth class label i.

[3] `https://github.com/alinlab/CSI`

E Self-Supervised Probing

In our experiments, we design rotation and translation as the probing tasks. Specifically, we design rotation prediction tasks with 4 degrees $\{0°, 90°, 180°, 270°\}$ and translation tasks with 5 different translated pixels $\{(0, 0), (-8, 0), (8, 0), (0, -8), (0, 8)\}$ for both CIFAR-10 and CINIC-10 across different backbone models. As for the smaller dataset STL-10, we design easier probing tasks: rotation prediction tasks with 2 degrees $\{0°, 90°\}$ and translation tasks with 3 different translated pixels $\{(0, 0), (0, -32), (0, 32)\}$. To avoid the trivial solution for learning translation tasks, the translated images' paddings are interpolated under reflection mode. The probing classifiers are trained for 10 epochs using the SGD optimizer and the training learning rate is scheduled with cosine annealing schedulers.

F More Experimental Results

This section provides more experimental results as follows.

F.1 The empirical evidence for probing confidence

Figure 1 shows the correlation between probing confidence and accuracy with VGG16 backbones trained on the CIFAR-10 and CINIC-10 datasets.

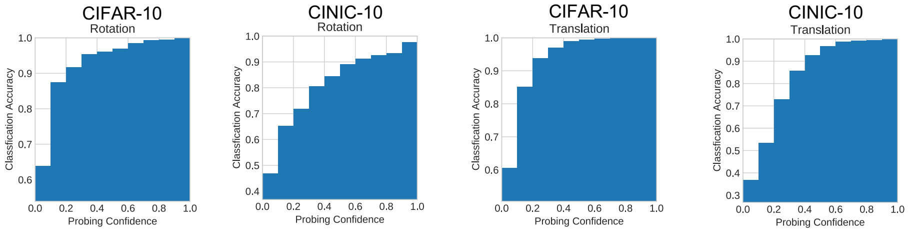

Fig. 1: Clear positive correlation between classification accuracy and probing confidence under the rotation and translation probing tasks with VGG16 models trained on CIFAR-10 and CINIC-10, respectively.

F.2 Q1: When does self-supervised probing adjust the original decision to be more (or less) confident?

This section provides the qualitative analysis of our proposed framework. As observed in the main paper, the images with clear and sharp objects tend to succeed in both object classification and probing tasks, while the images with objects intrinsically hard to detect tend to fail in the probing tasks even for object classification. Going a step further, it is natural to ask whether the classifier has

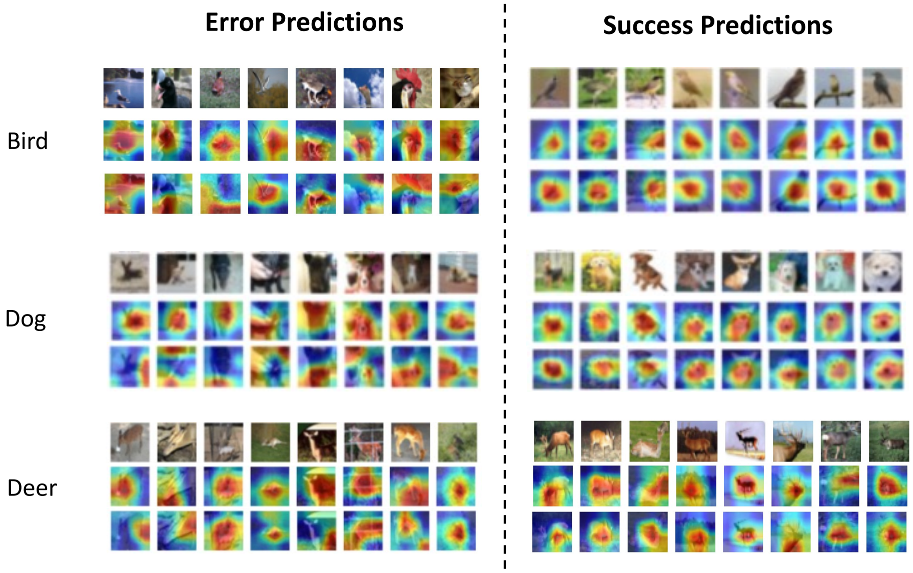

Fig. 2: Error and success predictions with activation maps in object classification and probing tasks. For each sample, the first row shows the original images, the second row shows the activation maps for object classification and the third row demonstrates the activation maps for the probing task. The success predictions tend to have consistent regions in the activation maps, which cover the objects of interest in the images.

been able to correctly localize the objects in the images such that it can perform well in both object classification and probing classification. To answer it, we use a gradient-based visualization algorithm, Grad-Cam [9][4], to show the activation maps for both object classification and probing task (rotation) classification. As demonstrated in Figure 2, we observe that, for successful predictions, the resultant activation maps obtained from two tasks are consistent with the highlighted regions enveloping the objects of interest. In contrast, the misclassification samples tend to exhibit the activation maps of different highlighted regions for object classification and probing tasks.

F.3 Q2: How do different combinations of probing tasks affect performance?

We have conducted ablation study on the combinations of probing tasks. Specifically, we have designed varying numbers of transformations in a probing task. For example, for rotation task, we have designed 2, 4, 6 transformations: $\{0°, 90°\}$, $\{0°, 90°, 180°, 270°\}$ and $\{0°, 60°, 120°, 180°, 240°, 300°\}$. For translation tasks,

[4] `https://github.com/jacobgil/pytorch-grad-cam`

Table 2: Results (AUROC %) for different transformations in a probing task.

	CIFAR-10	CINIC-10	STL-10
#rotations = 2	92.25	88.12	79.43
#rotations = 4	91.97	88.10	78.87
#rotations = 6	91.79	88.14	78.69
#translations = 3	91.11	88.28	79.22
#translations = 5	91.22	88.15	78.90
#translations = 9	91.26	88.11	78.80

we have designed 3, 5, 9 transformations: $\{(0,0),(0,-8),(0,8)\}$, $\{(0,0),(-8,0),(8,0),(0,-8),(0,8)\}$ and $\{(0,0),(-8,0),(8,0),(0,-8),(0,8),(8,8),(-8,-8),(8,-8),(-8,8)\}$. The results are reported based on the ResNet-18 models trained on different datasets under misclassification detection settings. We report the results with AUROC (%) in the Table 2.

F.4 Comparison with the baseline of using randomly initialized probing head.

Figure 3 displays the correlation between classification accuracy and probing confidence with randomly initialized head without any training. We obtained the probing confidence that is almost uniformly distributed and not correlated to the classification accuracy any more, which further verifies the nontrivial effectiveness of the SSL learned probing heads. We use normal distribution $\mathcal{N}(0,1)$ for the random head initialization.

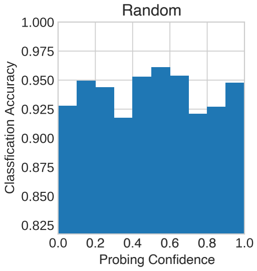

Fig. 3: The correlation between classification accuracy and probing confidence with randomly initialized probing head.

References

1. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on

artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)

2. Corbière, C., Thome, N., Bar-Hen, A., Cord, M., Pérez, P.: Addressing failure prediction by learning model confidence. arXiv preprint arXiv:1910.04851 (2019)
3. Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: Cinic-10 is not imagenet or cifar-10. arXiv preprint arXiv:1810.03505 (2018)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
5. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
8. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
9. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
11. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence **30**(11), 1958–1970 (2008)
12. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)