Learning to Predict Trustworthiness of Classifiers with Steep Slope Loss

Anonymous Author(s)
Affiliation
Address
email

Abstract

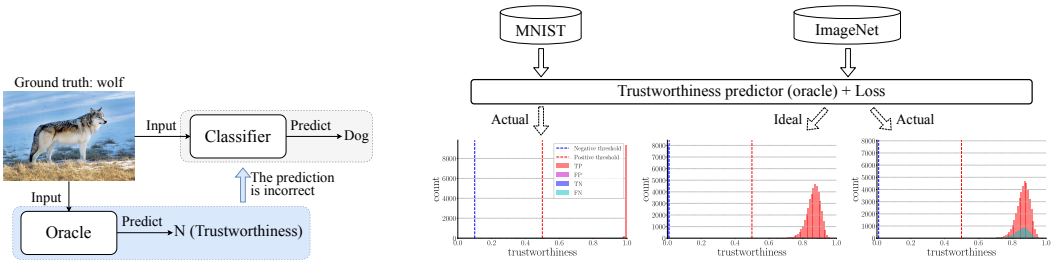
Understanding if a prediction yielded by a classifier is correct is critical to safely and effectively use AI techniques. Prior efforts have been proved to be reliable on small-scale datasets. In this work, we study the problem of predicting trustworthiness on a real-world large-scale dataset, i.e., ImageNet, where the task is more challenging due to high-dimensional features, diverse visual concepts, and large-scale samples. In such a theme, we observe that the trustworthiness predictors trained with prior-art loss functions, i.e., the cross entropy, focal loss, and true class probability confidence loss (TCP for short), are prone to view both correct predictions and incorrect predictions to be trustworthy. The reasons are two-fold. Firstly, correct predictions are generally dominant over incorrect predictions. Secondly, due to the data complexity, it is more challenging to differentiate the incorrect predictions from the correct predictions on real-world large-scale datasets. To improve the generalizability of trustworthiness predictors, we propose a novel loss function named the steep slope loss, where the features ride down two slides such that the ones w.r.t. trustworthy predictions are separated from the ones w.r.t. untrustworthy predictions. The proposed loss function is evaluated with two representative deep learning models, i.e., Vision Transformer (ViT) and ResNet, as trustworthiness predictors. Experimental results show that the proposed loss effectively improves the generalizability of trustworthiness predictors. The code and pre-trained trustworthiness predictors for reproducibility are available at `https://gitlab.com/anonymous_science/steep_slope.git`.

1 Introduction

Classification is a ubiquitous learning problem that categorizes objects according to input features, which is widely used in a range of applications, such as robotics [1], environment exploration [2], medical diagnosis [3], etc. The correctness of predictions yielded by classifiers determines the outcomes of decisions. In spite of the successful development of deep learning methods in recent decades, high-performance classifiers would still have a chance to make mistakes due to the complexity in the nature of real-world data [4, 5, 6, 7].

To access whether the prediction yielded by a classifier can be trusted or not, there are growing efforts towards learning to predict trustworthiness [8, 9]. These methods are evaluated on small-scale datasets, e.g., MNIST [10], where the data is relatively simple to separate such as the accuracies of most classifiers are considerably high ($> 99\%$). As such, the trustworthiness predictors that are trained on small-scale datasets tend to classify failure predictions as trustworthy predictions. To further understand the prowess and efficiency of predicting trustworthiness, we study this problem on the real-world large-scale datasets, e.g., ImageNet. This is a challenging theme for classification in

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) (b)

Figure 1: Conceptual illustrations related to predicting trustworthiness. (a) is the process of predicting trustworthiness. (b) is an illustration showing that the task is challenging on ImageNet. The state-of-the-art TCP confidence loss is used in this example. Ideally, we hope the trustworthiness confidence w.r.t. trustworthy predictions can be on the right-hand side of positive threshold 0.5 and the trustworthiness confidence w.r.t. untrustworthy predictions can be on the left-hand side of negative threshold 0.001 in the case of TCP. Note that we slightly adjust the negative threshold for illustrative purposes. TP, FP, TN, and FN stand for true positive, false positive, true negative, and false negative, respectively.

terms of class ambiguity, boundary complexity, and feature space dimensionality [11]. As a result, failure predictions are inevitable with a certain probability.

On the other hand, more and more pre-trained classifiers, e.g., ResNet [5], ViT [7], etc, become publicly available. Using the pre-trained classifiers would save considerable training time and computational resource. In contrast to the prior studies of predicting trustworthiness [8, 9], where the classifiers are relatively light-weight, how well the parameter-heavy classifiers perform on real-world images is still unclear. In this work, we study the problem of predicting trustworthiness with state-of-the-art classifiers.

A general illustration of predicting trustworthiness is shown in Fig. 1a. We term the entity that learns to predict trustworthiness of the observed classifier as the *oracle*, which has a counterpart named the confidence network in the work of true class probability (TCP) [9]. A higher confidence score from the classifier does not necessarily imply higher probability that the classifier is correct, as shown in [12, 13, 14, 8]. Instead of adopting the maximum confidence, TCP adopts the confidence w.r.t. the ground-truth class (i.e., TCP) as the indicator for predicting trustworthiness. As a result, it names its trustworthiness predictor as the confidence network. In contrast, we use a straightforward indicator, that is, the prediction is trustworthy if the classifier yields the correct label, otherwise untrustworthy.

Similar to the learning framework used in [9], we use powerful deep learning models as the oracle and train them with the widely-used loss functions, i.e., the cross entropy, focal loss [15], and TCP confidence loss [9], on the ImageNet training set [16]. We observe that the resulting oracles are prone to overfit the training samples, i.e., the true positive rate (TPR) is close to 100% while the true negative rate (TNR) is close to 0%. To improve the generalizability of oracles, we propose a novel loss function named the steep slope loss. The proposed steep slope loss consists of two slides that cross with each other and face in the opposite direction to separate the features w.r.t. trustworthy and untrustworthy predictions generated by the oracle. It is tractable to control the slopes by indicating the heights of slides. In this way, the proposed loss can effectively prevent the learning process from overfitting the training samples.

Predicting trustworthiness is different from conventional classification tasks, whilst it essentially is a binary classification task. Firstly, task-specific semantics are different between conventional classification task and predicting trustworthiness. The classes, i.e., being trustworthy or untrustworthy, in predicting trustworthiness are abstract concepts that could work on top of the classes in conventional classification tasks, e.g., human, cat, dog, etc. Secondly, the problem of domain generalization in predicting trustworthiness is more difficult than that of conventional classification tasks. For instance, the oracle may run into images of cat on the fly, when it learns with images of dog in the training phase. The dog images and the cat image should share much different visual appearance. In contrast, a real-world cat image is expected to shared similar visual appearance with a cartoon cat image.

The contribution of this work can be summarized as follows.

- We study the problem of predicting trustworthiness of state-of-the-art classifiers on ImageNet. Specifically, we observe that a major challenge of this learning task is that the cross entropy, focal, and TCP losses are prone to overfit the training samples.
- We propose the steep slope loss function that improves overfitting. Comprehensive experimental results and analyses are provided to prove the efficacy of the proposed loss.
- To further explore the practical usefulness of the proposed loss, we train the oracle on the ImageNet training set and evaluate the learned oracle on two variants of the ImageNet validation set in the unseen domains, i.e., the stylized validation set and the adversarial validation set. We find that the learned oracle is able to consistently differentiate the trustworthy predictions from the untrustworthy predictions.

2 Preliminaries

In this section, we first recap how a deep learning method learns to classify images in the classification task. Then, we show how the task of predicting trustworthiness connects to the classification task.

Supervised Learning for Classification. In classification tasks, given a training sample, i.e., image $\boldsymbol{x} \in \mathbb{R}^m$ and corresponding ground-truth label $y \in \mathcal{Y} = \{1, \ldots, K\}$, we assume that samples are drawn i.i.d. from an underlying distribution. The goal of the learning task is to learn to find a classifier $f^{(cls)}(\cdot; \theta')$ with training samples for classification. θ' is the set of parameters of the classifier. Let $f_\theta^{(cls)}(\cdot) = f^{(cls)}(\cdot; \theta')$. The optimization problem is defined as

$$f_{\theta'}^{*(cls)} = \arg\min_{f_{\theta'}^{(cls)}} \hat{\mathcal{R}}(f_{\theta'}^{(cls)}, \ell^{(cls)}, D_{tr}), \tag{1}$$

where $f_{\theta'}^{*(cls)}$ is the learned classifier, $\hat{\mathcal{R}}$ is the empirical risk, $\ell^{(cls)}$ is a loss function for classification, and D_{tr} is the set of training samples.

Supervised Learning for Predicting Trustworthiness. In contrast to the learning task for classification, which is usually a multi-class single-label classification task [4, 5, 6, 7], learning to predicting trustworthiness is a binary classification problem, where the two classes are: positive (i.e., trustworthy) or negative (i.e., untrustworthy). Similar to [9], given a pair $(\boldsymbol{x}, y)$ and a classifier $f_{\theta'}^{(cls)}$, we define the ground-truth label o for predicting trustworthiness as

$$o = \begin{cases} 1, & \text{if } \arg\max f_{\theta'}^{(cls)}(\boldsymbol{x}) = y \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

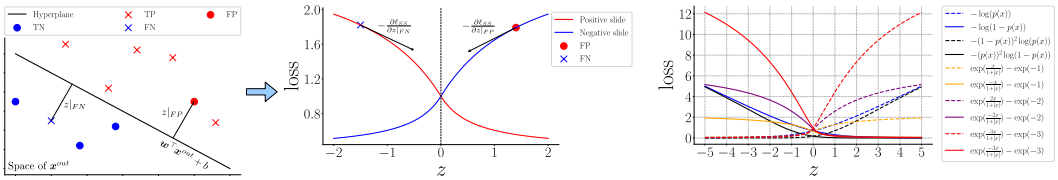
In other words, the classifier correctly predicts the image's label so the prediction is trustworthy in hindsight, otherwise the prediction is untrustworthy.

The learning task for predicting trustworthiness follows a similar learning framework in the classification task. Let $f_\theta(\cdot)$ be an oracle (i.e., trustworthiness predictor). A generic loss function $\ell : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ is a non-negative space. Given training samples $(\boldsymbol{x}, y) \in D_{tr}$, the optimization problem for predicting trustworthiness is defined as

$$f_\theta^* = \arg\min_{f_\theta} \frac{1}{|D_{tr}|} \sum_{i=1}^{|D_{tr}|} \ell(f_\theta(\boldsymbol{x}_i), o_i), \tag{3}$$

where $|D_{tr}|$ is the cardinality of D_{tr}.

Particularly, we consider two widely-used loss functions for classification and the loss function used for training trustworthiness predictors as baselines. They are the cross entropy [17], focal loss [15], and TCP confidence loss [9]. Let $p(o = 1|\theta, \boldsymbol{x}) = 1/(1 + \exp(-z))$ be the trustworthiness confidence, where $z \in \mathbb{R}$ is the descriminative feature produced by the oracle, i.e., $z = f_\theta(\boldsymbol{x})$. The



(a) (b)

Figure 2: Conceptual workflow of the proposed steep slope loss (a) and comparison between the proposed loss and the conventional losses (b), i.e., cross entropy an focal loss. The TCP confidence loss is a square error and varies with the classification confidence. Therefore, it is not plotted with the other loss functions.

three loss functions can be written into

$$\ell_{CE}(f_\theta(\boldsymbol{x}), o) = -o \cdot \log p(o = 1|\theta, \boldsymbol{x}) - (1 - o) \cdot \log(1 - p(o = 1|\theta, \boldsymbol{x})), \tag{4}$$

$$\begin{aligned}\ell_{Focal}(f_\theta(\boldsymbol{x}), o) = &-o \cdot (1 - p(o = 1|\theta, \boldsymbol{x}))^\gamma \log p(o = 1|\theta, \boldsymbol{x}) - \\ &(1 - o) \cdot (p(o = 1|\theta, \boldsymbol{x}))^\gamma \log(1 - p(o = 1|\theta, \boldsymbol{x})),\end{aligned} \tag{5}$$

$$\ell_{TCP}(f_\theta(\boldsymbol{x}), y) = (f_\theta(\boldsymbol{x}) - p(\hat{y} = y|\theta', \boldsymbol{x}))^2. \tag{6}$$

In the focal loss, γ is a hyperparameter. In the TCP confidence loss, $\hat{y}$ is the predicted label and $p(\hat{y} = y|\theta', \boldsymbol{x})$ is the classification probability w.r.t. the ground-truth class.

Consequently, the learned oracle would yield z to generate the trustworthiness confidence. In the cases of ℓ_{CE} and ℓ_{Focal}, the oracle considers a prediction is trustworthy if the corresponding trustworthiness confidence is greater than the positive threshold 0.5, i.e., $p(o = 1|\theta, \boldsymbol{x}) > 0.5$. The predictions whose trustworthiness confidences are equal to or lower than the negative threshold 0.5 are viewed to be untrustworthy. In the case of ℓ_{TCP}, the positive threshold is still 0.5, but the negative threshold correlates to the number of classes in the classification task. It is defined as $1/K$ [9].

3 Methodology

In this section, we first introduce an overall learning framework for predicting trustworthiness. Then, we narrow down to the proposed steep slope loss function. At last, we provide the generalization bound that is related to the proposed steep slope loss function.

3.1 Overall Design

[9] provides a good learning scheme for predicting trustworthiness. In short, it first trains a classifier with training samples. Then, the classifier is frozen and the confidence network (i.e., trustworthiness predictor) is trained/fine-tuned with the training samples. In general, we follows such a learning scheme.

Instead of training classifiers from scratch, we use publicly available pre-trained classifiers, i.e., ViT [7] and ResNet [5]. On the other hand, we use the backbones of the pre-trained classifiers as the backbones of the oracles for general purposes. In this sense, the set of the oracle's parameters can be split into two parts, one is related to the backbone and the other one is related to the head, i.e., $\theta = \{\theta_{backbone}, \theta_{head}\}$. $\theta_{backbone}$ is used to generated the intermediate feature $\boldsymbol{x}^{out}$ and $\theta_{head} = \{\boldsymbol{w}, b\}$ are usually the weights of a linear function to generate the discriminative feature $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$. With the classifier, the oracle, a pre-defined loss, and the training samples, now we can optimize the problem (3) to find the optimal parameters for the oracle.

3.2 Steep Slope Loss

The conceptual workflow of the proposed steep slope loss is shown in Fig. 2a. The core idea is that we exploit the graph characteristics of the exponential function and the softsign function to establish two slides such that the features z w.r.t. the positive class ride down the positive slide to the right

bottom and the features z w.r.t. the negative class ride down the negative slide to the left bottom. z is defined as the signed distance to the hyperplane (i.e., oracle head). Given an image $\boldsymbol{x}$, $z = f_\theta(\boldsymbol{x})$ can be broken down into

$$z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out} + b}{\|\boldsymbol{w}\|}, \quad \boldsymbol{x}^{out} = f_{\theta_{backbone}}(\boldsymbol{x}). \tag{7}$$

The signed distance to the hyperplance has a geometric interpretation: its sign indicates in which half-space $\boldsymbol{x}^{out}$ is and its absolute value indicates how far $\boldsymbol{x}^{in}$ is away from the hyperplane.

It is desired that the signed distance of $\boldsymbol{x}^{out}$ with ground-truth label $o = 1$ tends towards $+\infty$ as much as possible and the signed distance of $\boldsymbol{x}^{out}$ with ground-truth label $o = 0$ tends towards $-\infty$ as much as possible. To this end, we define the steep slope (SS) loss function as follows

$$\ell_{SS}(f_\theta(\boldsymbol{x}), o) = o \cdot \underbrace{\left(\exp\left(\frac{\alpha^+ z}{1+|z|}\right) - \exp(-\alpha^+)\right)}_{\text{Positive slide}} + (1-o) \cdot \underbrace{\left(\exp\left(\frac{-\alpha^- z}{1+|z|}\right) - \exp(-\alpha^-)\right)}_{\text{Negative slide}} \tag{8}$$

where $\alpha^+, \alpha^- \in \mathbb{R}^+$ control the slope of the positive slide and the negative slide, respectively. If z w.r.t. the positive class is on the left-hand side of $z = 0$, minimizing the loss would push the point on the hill down to the bottom, i.e., the long tail region indicating the well-classified region. Similarly, z w.r.t. the negative class would undergo a similar process. $\exp(-\alpha^+)$ and $\exp(-\alpha^-)$ are vertical shifts for the positive and negative slides such that ℓ_{ss} has a minimum value 0. Note that the proposed steep slope loss is in the range $[0, \text{maximum}\{\exp(\alpha^+) - \exp(-\alpha^+), \exp(\alpha^-) - \exp(-\alpha^-)\}]$, whereas the cross entropy and the focal loss are in the range $[0, +\infty)$. The proposed steep slope loss can work with the output of the linear function as well. Because the signed distance and the output of the linear function have a proportional relationship with each other, i.e., $\frac{\boldsymbol{w}^\top \boldsymbol{x}^{out} + b}{\|\boldsymbol{w}\|} \propto \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$.

Essentially, as shown in Fig. 2b, the cross entropy, focal loss, and steep slope loss work in a similar manner to encourage z w.r.t. the positive class to move the right-hand side of the positive threshold 0 and encourage z w.r.t. the negative class to move the left-hand side of the negative threshold 0, which is analogous to the sliding motion. The proposed steep slope loss is more tractable to control the steepness of slopes than the cross entropy and focal loss. This leads to an effective way learning to yield discriminative feature for predicting trustworthiness.

3.3 Generalization Bound

With the proposed steep slope loss, we are interested in the generalization bound of trustworthiness predictors. For simplicity, we simplify a trustworthiness predictor as $f \in \mathcal{F}$, where $\mathcal{F}$ is a finite hypothesis set. The risk of predicting trustworthiness is defined as $\mathcal{R}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell_{SS}(f(\boldsymbol{x}), o)]$, where P is the underlying joint distribution of $(\boldsymbol{x}, o)$. As P is inaccessible, a common practice is to use empirical risk minimization (ERM) to approximate the risk [18], i.e., $\hat{\mathcal{R}}_D(f) = \frac{1}{|D|}\sum_{i=1}^{|D|} \ell_{SS}(f(\boldsymbol{x}_i), o_i)$.

The following theorem provides an insight into whether an oracle yields a consistent separability over unseen sample from all domains, given a separability computed with seen samples on a seen domain.

Theorem 3.1. *Denote maximum*$\{\exp(\alpha^+) - \exp(-\alpha^+), \exp(\alpha^-) - \exp(-\alpha^-)\}$ *as* ℓ_{SS}^{max}. $\ell_{SS} \in [0, \ell_{SS}^{max}]$. *Assume* $\mathcal{F}$ *is a finite hypothesis set, for any* $\delta > 0$*, with probability at least* $1 - \delta$*, the following inequality holds for all* $f \in \mathcal{F}$*:*

$$|\mathcal{R}(f) - \hat{\mathcal{R}}_D(f)| \leq \ell_{SS}^{max}\sqrt{\frac{\log|\mathcal{F}| + \log\frac{2}{\delta}}{2|D|}}$$

The proof sketch is similar to the generalization bound provided in [19] and the detailed proof is provided in the appendix A.1.

A desired characteristic of the proposed steep slope loss is that it is in a certain range determined by α^+ and α^-, as discussed in Section 3.2. This leads to the generalization bound shown in Theorem 3.1. The theorem implies that given a generic classifier, as the number of training samples increases, the empirical risk would be close to the true risk with a certain probability.

4 Related Work

Loss Function. The loss function is key to search for optimal parameters and has been extensively studied in a range of contexts of learning tasks [20, 21, 15, 9]. Specifically, the cross entropy loss may be the most widely-used loss for classification [20]. However, the cross entropy loss is not optimal for all cases. Lin et al. propose the focal loss to down-weight the loss w.r.t. well-classified examples and focus on minimizing the loss w.r.t. misclassified examples by reshaping the cross entropy loss function [15]. Liu et al. introduce the large margin softmax loss that is based on cosine similarity between the weight vectors and the feature vectors [21]. Nevertheless, it requires more hyperparameters and entangles with the linear function in the last layer of the network. Similar to the idea of the focal loss, the proposed steep loss is flexible to reshape the function graphs to improve the class imbalance. TCP [9] confidence loss is used to train the confidence network for predicting trustworthiness. We adopt TCP as a baseline in this work.

Trustworthiness Prediction. Understanding trustworthiness of a classifier has been studied in the past decade [22, 23, 8, 9, 24]. Hendricks et al. aim to detect the misclassified examples according to maximum class probability [23]. This method depends on the ranking of confidence scores and is proved to be unreliable [8]. Monte Carlo dropout (MCDropout) intends to understand trustworthiness through the lens of uncertainty estimation [22]. Still, MCDropout is difficult to distinguish the trustworthy prediction from the incorrect but overconfident predictions [9]. Jiang et al. propose a confidence measure named trust score, which is based on the distance between the classifier and a modified nearest-neighbor classifier on the test examples [8]. The shortcomings of the method are the poor scalability and computationally expensive. Recently, Moon et al. does not implement a standalone trustworthiness predictor but use the classifier's weights to compute the correctness ranking [24]. The correctness ranking is taken into account in the loss function such that the classification accuracy is improved. This method relies on the pairs of training samples for computing the ranking. Moreover, it is unclear if the classifier improved by the correctness ranking outperforms the pre-trained classifier on ImageNet. In contrast, the learning scheme to train the trustworthiness predictor (i.e., confidence network) in [9] is standard and does not affect the classifiers. Its efficacy has been verified on small-scale datasets. In this work, we follows the learning scheme used in [9].

Classifier. Deep learning models achieve remarkable success in the classification tasks [4, 5, 6, 7, 25]. To comprehensively evaluate the proposed loss, we use two representative types of deep learning models, i.e., ViT (transformer based network) [25] and ResNet (convolutional network) [5], as the classifiers in this work.

5 Experiment & Analysis

In this section, we first introduce the experimental set-up. Then, we report the performances of baselines and the proposed loss on ImageNet. At last, we present comprehensive analyses to help better understand the efficacy of the proposed loss.

Experimental Set-Up. As stated before, we use ViT B/16 [7] and ResNet-50 [5] as the classifiers. Also, the backbones of the two models are used as the oracles' backbones. We denote the combination of oracles and classifiers as $\langle$O, C$\rangle$. There are four combinations in total, i.e., $\langle$ViT, ViT$\rangle$, $\langle$ViT, RSN$\rangle$, $\langle$RSN, ViT$\rangle$, and $\langle$RSN, RSN$\rangle$, where RSN stands for ResNet. In this work, we adopt three baselines, i.e., the cross entropy [20], focal loss [15], and TCP confidence loss [9] for comparison purposes.

We evaluate the baselines and the proposed loss on ImageNet [16], which consists of 1.2 million labeled training images and 50000 labeled validation images. It has 1000 visual concepts. Similar to the learning scheme used in [9], the oracle is trained with training samples and evaluated on the validation set. During the training process of the oracle, the classifier works in the evaluation mode so training the oracle would not affect the parameters of the classifier. Moreover, we conduct the analyses about how well the learned oracle generalizes to the images in the unseen domains. To this end, we apply the widely-used style transfer method [26] and the functional adversarial attack method [27] to generate two variants of the validation set, i.e., stylized validation set and adversarial validation set. Then, the oracle trained with regular training samples would be evaluated with the samples that are in the two unseen domains.

As the oracle's backbone is initialized by the pre-trained classifier's backbone. The training process of the oracles is equivalent to the process of fine-tuning the initialized oracles. Training the oracles with

Table 1: Performance on the ImageNet validation set. The oracles are trained with the ImageNet training samples. The classifier is used in the evaluation mode in the experiment. Acc is the classification accuracy and is helpful to understand the proportion of correct predictions.

⟨O, C⟩	Loss	Acc↑	FPR-95%-TPR↓	AUPR-Error↑	AUPR-Success↑	AUC↑	TPR↑	TNR↑
⟨ViT, ViT ⟩	CE	83.90	92.83	**15.08**	84.99	52.78	**100.00**	0.01
	Focal [15]	83.90	92.68	14.69	85.46	53.47	99.06	1.61
	TCP [9]	83.90	88.07	12.86	87.80	60.45	99.72	0.00
	SS	83.90	**80.89**	10.31	**92.90**	**73.31**	88.44	**35.64**
⟨ViT, RSN⟩	CE	68.72	93.10	30.93	69.10	51.26	**99.85**	0.21
	Focal [15]	68.72	94.51	**31.47**	68.57	50.40	93.25	7.33
	TCP [9]	68.72	84.16	23.15	79.85	67.09	94.33	0.00
	SS	68.72	**78.09**	20.94	**85.30**	**74.20**	67.96	**67.80**
⟨RSN, ViT⟩	CE	83.90	92.58	14.59	85.57	53.78	**100.00**	0.00
	Focal [15]	83.90	94.92	**14.87**	85.26	52.49	**100.00**	0.00
	TCP [9]	83.90	91.63	14.17	86.06	55.37	**100.00**	0.00
	SS	83.90	**88.63**	11.75	**89.87**	**64.11**	95.41	**10.48**
⟨RSN, RSN⟩	CE	68.72	94.83	29.52	70.67	52.19	**100.00**	0.00
	Focal [15]	68.72	95.42	**30.21**	69.88	51.16	99.91	0.03
	TCP [9]	68.72	88.54	24.62	77.42	62.53	99.17	0.00
	SS	68.72	**85.60**	22.50	**81.92**	**68.08**	80.44	**40.85**

Table 2: Performances on the stylized ImageNet validation set and the adversarial ImageNet validation set. ⟨ViT, ViT⟩is used in the experiment and the domains of the two validation sets are different from the one of the training set that is used for training the oracle.

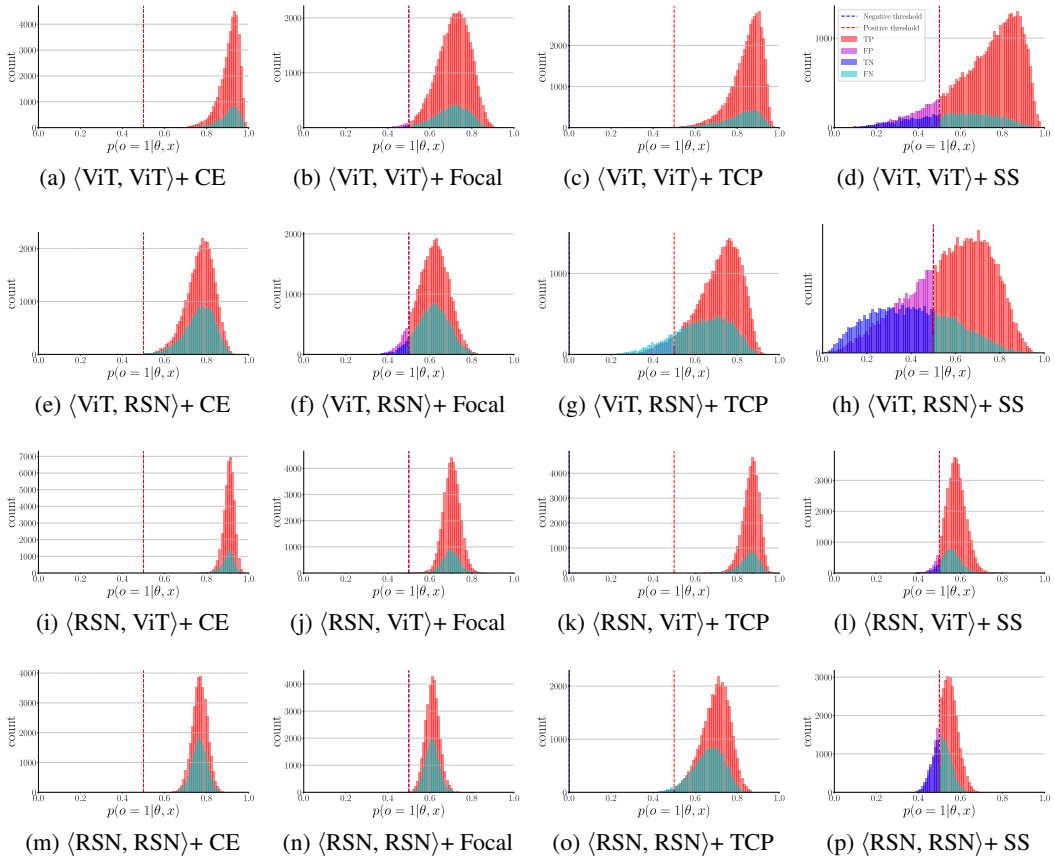
Dataset	Loss	Acc↑	FPR-95%-TPR↓	AUPR-Error↑	AUPR-Success↑	AUC↑	TPR↑	TNR↑
Stylized [26]	CE	15.94	95.52	84.18	15.86	49.07	**99.99**	0.02
	Focal [15]	15.94	95.96	**85.90**	14.30	46.01	99.71	0.25
	TCP [9]	15.94	93.42	80.17	21.25	57.29	99.27	0.00
	SS	15.94	**89.38**	75.08	**34.39**	**67.68**	44.42	**81.22**
Adversarial [27]	CE	6.14	94.35	**93.70**	6.32	51.28	**99.97**	0.06
	Focal [15]	6.15	93.67	93.48	6.56	52.39	99.06	1.43
	TCP [9]	6.11	93.94	92.77	7.55	55.81	99.71	0.00
	SS	6.16	**90.07**	90.09	**13.07**	**65.36**	87.10	**24.33**

all the loss functions uses the same hyperparameters, such as learning rate, weight decay, momentum, batch size, etc. The details for the training process and the implementation are provided in appendix A.2.

We use $\alpha^+ = 1$ and $\alpha^- = 3$ for the oracle that is based on ViT's backbone, while we use $\alpha^+ = 2$ and $\alpha^- = 5$ for the oracle that is based on ResNet's backbone.

Following [9], we use FPR-95%-TPR, AUPR-Error, AUPR-Success, and AUROC as the metrics. FPR-95%-TPR is the false positive rate (FPR) when the true positive rate (TPR) is equal to 95%. AUPR is the area under the precision-recall curve. Specifically, AUPR-Success considers the correct prediction as the positive class, whereas AUPR-Error considers the incorrect prediction as the positive class. AUROC is the area under the receiver operating characteristic curve, which is the plot of TPR versus FPR. Moreover, we use the TPR and the ture negative rate (TNR) as the additional metrics because they are useful to assess overfitting, e.g., TPR=100% and TNR=0% imply that the trustworthiness predictor is prone to view all the incorrect predictions to be trustworthy due to overfitting.

Performance on ImageNet. As reported in Table 1, we have two observations. Firstly, using the cross entropy, focal loss, and TCP confidence loss to train oracles lead to overfitting the imbalanced training samples, i.e., the dominance of trustworthy predictions. Specifically, TPR is higher than 99% while TNR is less than 1%. Secondly, the performance of predicting trustworthiness is contingent on both the oracle and the classifier. When a high-performance model (i.e., ViT) is used as the oracle and a relatively low-performance model (i.e., ResNet) is used as the classifier, the cross entropy and focal



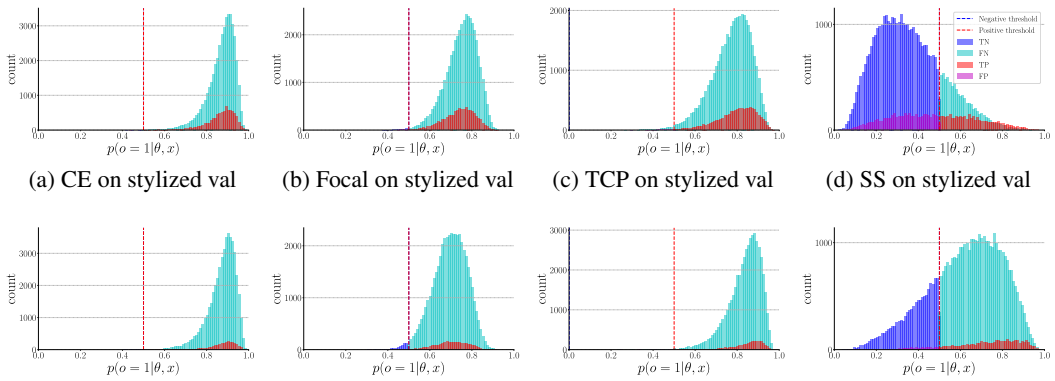
(a) ⟨ViT, ViT⟩+ CE (b) ⟨ViT, ViT⟩+ Focal (c) ⟨ViT, ViT⟩+ TCP (d) ⟨ViT, ViT⟩+ SS

(e) ⟨ViT, RSN⟩+ CE (f) ⟨ViT, RSN⟩+ Focal (g) ⟨ViT, RSN⟩+ TCP (h) ⟨ViT, RSN⟩+ SS

(i) ⟨RSN, ViT⟩+ CE (j) ⟨RSN, ViT⟩+ Focal (k) ⟨RSN, ViT⟩+ TCP (l) ⟨RSN, ViT⟩+ SS

(m) ⟨RSN, RSN⟩+ CE (n) ⟨RSN, RSN⟩+ Focal (o) ⟨RSN, RSN⟩+ TCP (p) ⟨RSN, RSN⟩+ SS

Figure 3: Histograms of trustworthiness confidences w.r.t. all the loss functions on the ImageNet validation set. The oracles that are used to generate the confidences are the ones used in Table 1.

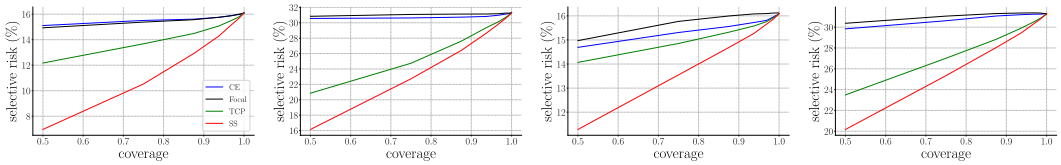


(a) CE on stylized val (b) Focal on stylized val (c) TCP on stylized val (d) SS on stylized val

(e) CE on adversarial val (f) Focal on adversarial val (g) TCP on adversarial val (h) SS on adversarial val

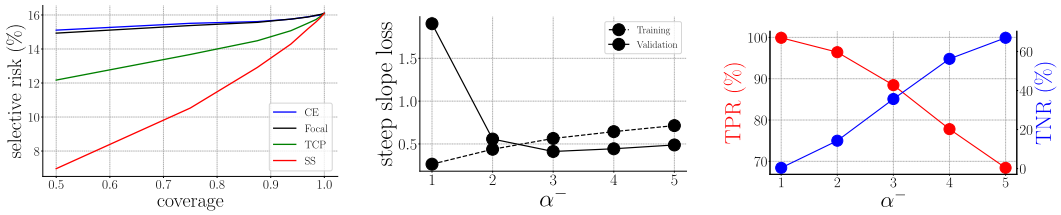
Figure 4: Histograms of trustworthiness confidences w.r.t. all the loss functions on the stylized ImageNet validation set (stylized val) and the adversarial ImageNet validation set (adversarial val). ⟨ViT, ViT⟩is used in the experiment and the domains of the two validation sets are different from the one of the training set that is used for training the oracle. The histograms correspond to the oracles used in Table 2.

loss with ⟨ViT, ResNet⟩achieve higher TNRs than the loss functions with the other combinations. In contrast, the two loss functions with ⟨ResNet, ViT⟩lead to the lowest TNRs (i.e., 0%), compared to the cases with the other combinations.



(a) ⟨ViT, ViT⟩ (b) ⟨ViT, RSN⟩ (c) ⟨RSN, ViT⟩ (d) ⟨RSN, RSN⟩

Figure 5: Curves of risk vs. coverage. Selective risk represents the percentage of errors in the remaining validation set for a given coverage percentage. The curves correspond to the oracles used in Table 1.



(a) (b) (c)

Figure 6: Analyses based on ⟨ViT, ViT⟩. (a) are the curves of risk vs. coverage. Selective risk represents the percentage of errors in the remaining validation set for a given coverage percentage. (b) are the loss curves against various α^-. (c) are TPR and TNR against various α^-.

Despite the combinations of oracles and classifiers, using the proposed steep slope loss to train oracles can achieve significantly higher TNRs than using the other loss functions, while it achieves desirable performance on metrics FPR-95%-TPR, AUPR-Success, and AUC. This verifies that the proposed loss is effective to improve the generalizability for predicting trustworthiness. Note that the scores of AUPR-Error and TPR yielded by the proposed loss are lower than that of the other loss functions. AUPR-Error aims to inspect how easy to detect failures and depends on the negated trustworthiness confidences w.r.t. incorrect predictions [9]. The AUPR-Error correlates to TPR and TNR. When TPR is close to 100% and TNR is close to 0%, it indicates the oracle is prone to view all the predictions to be trustworthy. In other words, almost all the trustworthiness confidences are on the right-hand side of $p(o = 1|\theta, \boldsymbol{x}) = 0.5$. Consequently, when taking the incorrect prediction as the positive class, the negated confidences are smaller than -0.5. On the other hand, the oracle trained with the proposed loss intends to yield the ones w.r.t. incorrect predictions that are smaller than 0.5. In general, the negated confidences w.r.t. incorrect predictions are greater than the ones yielded by the other loss functions. In summary, a high TPR score and a low TNR score leads to a high AUPR-Error.

To intuitively understand the effects of all the loss functions, we plot the histograms of trustworthiness confidences w.r.t. TP, FP, TN, and FN in Fig. 3. They confirm that the oracles trained with the cross entropy, focal loss, and TCP confidence loss are prone to predict overconfident trustworthiness for incorrect predictions, while the oracles trained with the proposed loss can properly predict trustworthiness for incorrect predictions.

Selective Risk Analysis. Risk-coverage curve is an important technique for analyzing trustworthiness through the lens of rejection mechanism in the classification task [28, 9]. In the context of predicting trustworthiness, selective risk is the empirical loss that takes into account the decisions, i.e., to trust the prediction or not to trust the prediction. Correspondingly, coverage is the probability mass of the non-rejected region. As can see in Fig. 5, the proposed loss leads to significantly lower risks, compared to the other loss functions. All the combinations of oracles and classifiers consistently exhibit similar pattern.

Performance on Small-Scale Datasets. We also provide comparative experimental results on small-scale datasets, i.e., MNIST [10] and CIFAR-10 [29]. The experiment details and the performance are report in appendix A.3. The proposed loss outperforms TCP [9] on metrics FPR-95%-TPR and AUC on MNIST, while it outperforms TCP on metrics FPR-95%-TPR, AUPR-Error, AUPR-Success,

and AUC on CIFAR-10. This results demonstrate that the proposed loss is able to adapt to relatively simple data.

Generalization to the Data in Unseen Domains. In practice, the oracle may run into the data in the domains that are different from the ones of training samples. Thus, it is interesting to find out how well the learned oracles generalize to the unseen domain data. To this end, we apply a style transfer method [26] and the functional adversarial attack method [27] to generate the stylized ImageNet validation set and the adversarial ImageNet validation set. The oracles are trained with the ImageNet training set (i.e., the ones used in Table 1) and evaluated on the two variants of the validation set. $\langle$ViT, ViT$\rangle$is used in the experiment.

The experimental results are reported in Table 2. As the images in the two sets have not been seen by both the oracles and the classifiers, the classification accuracies are much lower than the ones in Table 1 and the adversarial validation set is more challenging than the stylized validation set. As a result, the difficulty affects the scores of FPR-95%-TPR, AUPR-Error, AUPR-Success, and AUC. The oracles trained with the three baseline loss functions are still prone to recognize the incorrect prediction to be trustworthy. The proposed loss consistently improves the performance on metrics FPR-95%-TPR, AUPR-Sucess, AUC, and TNR. Note that the adversarial perturbations are computed on the fly [27]. Instead of truncating the sensitive pixel values and saving into the images files, we follow the experimental settings in [27] to evaluate the oracles on the fly. Therefore, as shown in Table 2, the classification accuracies w.r.t. various loss function are slightly different but are stably around 6.14%.

Ablation Study. In contrast to the cross entropy, focal loss, and TCP confidence loss, the proposed loss has more hyperparameters to be determined, that is, α^+ and α^-. As we know that the proportion of correct predictions is usually larger than that of incorrect predictions, we would prioritize α^- over α^+ such that the oracle is able to recognize a certain amount of incorrect predictions. In other words, we first search for α^- by freezing α^+. After searching for a desirable α^-, we freeze α^- and search for α^+.

Fig. 6 shows how the proposed loss, TPR, and TNR vary with various α^-. In this analysis, the combination $\langle$ViT, ViT $\rangle$is used and $\alpha^+ = 1$. We can see that $\alpha^- = 3$ achieves the optimal trade-off between TPR and TNR. We follow a similar search strategy to determine $\alpha^+ = 2$ and $\alpha^- = 5$ for training the oracle that is based on ResNet.

Effects of using $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$. Using the signed distance as z, i.e., $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out} + b}{\|\boldsymbol{w}\|}$, has a geometric interpretation as shown in Fig. 2a. However, the main-stream models [5, 6, 7] use $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$. Therefore, we provide the corresponding results in appendix A.4, which are generated by the proposed loss taking the output of the linear function as input. In comparison with the results of using $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out} + b}{\|\boldsymbol{w}\|}$, using $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$ yields comparable scores of FPR-95%-TPR, AUPR-Error, AUPR-Success, and AUC. Also, TPR and TNR are moderately different between $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out} + b}{\|\boldsymbol{w}\|}$ and $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$, when α^+ and α^- are fixed. This implies that TPR and TNR are sensitive to $\|\boldsymbol{w}\|$.

Separability between Distributions of Correct Predictions and Incorrect Predictions. As observed in Fig. 3, the confidences w.r.t. correct predictions and incorrect predictions follow Gaussian-like distributions. It is interesting to access the separability between the distributions of correct predictions and incorrect predictions from a probabilistic perspective. There are two common tools to achieve the goal, i.e., Kullback–Leibler (KL) divergence [30] and Bhattacharyya distance [31]. Given the distribution of correct predictions $\mathcal{N}_1(\mu_1, \sigma_1^2)$ and the distribution of correct predictions $\mathcal{N}_2(\mu_2, \sigma_2^2)$, we use the average KL divergence $d_{KL}(\mathcal{N}_1, \mathcal{N}_2)$ and Bhattacharyya distance $d_B(\mathcal{N}_1, \mathcal{N}_2)$ to measure the separability. More details and the quantitative results are reported in appendix A.5. The proposed loss leads to larger separability than the other three loss functions. This implies that the proposed loss is more effective to differentiate incorrect predictions from correct predictions, or not vice versa.

6 Conclusion

In this work, we study how to learn to predict trustworthiness on large-scale data. We observe that the oracle, i.e., trustworthiness predictor, trained with the cross entropy, focal loss, and TCP confidence

loss lean towards viewing incorrect predictions to be trustworthy due to overfitting. To improve the generalizability of the oracles, we propose the steep slope loss that encourages the features w.r.t. correct predictions and incorrect predictions to be separated from each other. We evaluate the proposed loss on ImageNet through the lens of the trustworthiness metrics, selective classification metric, and separability of distributions, respectively. Experimental results show that the proposed loss is effective in training the oracle for predicting trustworthiness.

References

[1] Daniel Leidner, Christoph Borst, Alexander Dietrich, Michael Beetz, and Alin Albu-Schäffer. Classifying compliant manipulation tasks for automated planning in robotics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1769–1776. IEEE, 2015.

[2] Antoine Caté, Lorenzo Perozzi, Erwan Gloaguen, and Martin Blouin. Machine learning as a tool for geologists. *The Leading Edge*, 36(3):215–219, 2017.

[3] José Antonio Sanz, Mikel Galar, Aranzazu Jurio, Antonio Brugos, Miguel Pagola, and Humberto Bustince. Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. *Applied Soft Computing*, 20:103–111, 2014.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, 2019.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[8] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[9] Charles Corbière, Nicolas THOME, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] Mitra Basu and Tin Kam Ho. *Data complexity in pattern recognition*. Springer Science & Business Media, 2006.

[12] Foster J. Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[14] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[17] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[18] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[19] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[20] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 507–516. JMLR.org, 2016.

[22] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

[24] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7034–7044. PMLR, 13–18 Jul 2020.

[25] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[26] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[27] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[28] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

[30] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[31] Anil Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, pages 401–406, 1946.

[32] Erick Cantu-Paz. Feature subset selection, class separability, and genetic algorithms. In *Genetic and evolutionary computation conference*, pages 959–970. Springer, 2004.

[33] Yan Luo, Yongkang Wong, Mohan Kankanhalli, and Qi Zhao. $\mathcal{G}$ -softmax: Improving intraclass compactness and interclass separability of features. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2):685–699, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[TODO]**
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[TODO]**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? **[TODO]**
   (b) Did you include complete proofs of all theoretical results? **[TODO]**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[TODO]**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[TODO]**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? **[TODO]**
   (b) Did you mention the license of the assets? **[TODO]**
   (c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[TODO]**
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[TODO]**
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[TODO]**

A Appendix

A.1 Generalization Bound

Theorem. *Denote maximum*$\{\exp(\alpha^+) - \exp(-\alpha^+), \exp(\alpha^-) - \exp(-\alpha^-)\}$ *as* ℓ_{SS}^{max}. $\ell_{SS} \in [0, \ell_{SS}^{max}]$. *Assume* $\mathcal{F}$ *is a finite hypothesis set, for any* $\delta > 0$*, with probability at least* $1 - \delta$*, the following inequality holds for all* $f \in \mathcal{F}$*:*

$$|\mathcal{R}(f) - \hat{\mathcal{R}}_D(f)| \leq \ell_{SS}^{max} \sqrt{\frac{\log |\mathcal{F}| + \log \frac{2}{\delta}}{2|D|}}$$

Proof. The proof sketch is similar to the generalization bound provided in [19]. By the union bound, given an error ξ, we have

$$p[\sup_{f \in \mathcal{F}} |R(f) - \hat{R}(f)| > \xi] \leq \sum_{f \in \mathcal{F}} p[|R(f) - \hat{R}(f)| > \xi].$$

By Hoeffding's bound, we have

$$\sum_{f \in \mathcal{F}} p[|\mathcal{R}(f) - \hat{\mathcal{R}}(f)| > \xi] \leq 2|\mathcal{F}| \exp\left(-\frac{2|D|\xi^2}{(\ell_{SS}^{max})^2}\right).$$

Due to the probability definition, $2|\mathcal{F}| \exp(-\frac{2|D|\xi^2}{(\ell_{SS}^{max})^2}) = \delta$. Considering ξ is a function of other variables, we can rearrange it as $\xi = \ell_{SS}^{max} \sqrt{\frac{\log |\mathcal{F}| + \log \frac{2}{\delta}}{2|D|}}$. Since we know $p[|\mathcal{R}(f) - \hat{\mathcal{R}}(f)| > \xi]$ is with probability at most δ, it can be inferred that $p[|\mathcal{R}(f) - \hat{\mathcal{R}}(f)| <= \xi]$ is at least $1 - \delta$. □

A.2 Experimental Set-Up

The ViT (ViT Base/16) used in this work is implemented in the ASYML project[1], which is based on PyTorch. The pre-trained weights are the same as the original pre-trained weights[2]. On the other hand, the pre-trained ResNet (ResNet-50) is provided in PyTorch[3].

We use the training scheme implemented by ASYML and tune the hyperparameters such that the oracles trained with the cross entropy and focal loss produce the best performance among multiple trials. Then, we fix the set of hyperparameters for the TCP confidence loss and the proposed loss. Specifically, the stochastic gradient descent (SGD) optimization method is used with initial learning rate 1e-5, weight decay 0, and momentum 0.05 for optimizing the learning problem. The 1cycle learning rate policy applies at each learning step. The batch size is fixed to 40 as ViT would make the full use of four 12 GB GPUs with 40 images. To stimulate a challenging and practically useful environment, we train the oracle in only one epoch, rather than multiple epochs. All the three baseline loss functions and the proposed loss use the same hyperparameters and undergo the same experimental protocol for training the oracle. The code is implemented in Python 3.8 with PyTorch 1.7.1 and is tested under Ubuntu 1804 with four Nvidia GTX 1080 ti graphics cards in a standalone machine.

A.3 Performance on Small-Scale Datasets

The resulting results are reported in Table A1. The experiment is based on the official implementation[4] of [9]. The implementation provides the pre-trained models on MNIST and CIFAR-10, but it does not provide the configurations for training the uncertainty networks. So we could not reproduce comparable performance reported in [9] from scratch. Instead, we fine-tune the pre-trained models with the proposed steep slope loss. For comparison purposes, we fine-tune the pre-trained with the TCP confidence loss (i.e., *TCP*†), where the experimental settings of the fine-tuning process are the same as the ones with the proposed steep slope loss. The proposed loss use $\alpha^+ = 10$ and $\alpha^- = 6$ on MNIST, and $\alpha^+ = 1$ and $\alpha^- = 1$ on CIFAR-10.

[1] `https://github.com/asyml/vision-transformer-pytorch`

[2] `https://github.com/google-research/vision_transformer`

[3] `https://pytorch.org/vision/stable/models.html`

[4] `https://github.com/valeoai/ConfidNet`

Table A1: Performance on MNIST and CIFAR-10.

Dataset	Loss	Acc↑	FPR-95%-TPR↓	AUPR-Error↑	AUPR-Success↑	AUC↑	TPR↑	TNR↑
MNIST	MCP [23]	99.10	5.56	35.05	**99.99**	98.63	99.89	**8.89**
	MCDropout [22]	99.10	5.26	38.50	**99.99**	98.65	-	-
	TrustScore [8]	99.10	10.00	35.88	99.98	98.20	-	-
	TCP [9]	99.10	3.33	**45.89**	**99.99**	98.82	99.71	0.00
	TCP†	99.10	3.33	45.88	**99.99**	98.82	99.72	0.00
	SS	99.10	**2.22**	40.86	**99.99**	**98.83**	**100.00**	0.00
CIFAR-10	MCP [23]	92.19	47.50	45.36	99.19	91.53	99.64	6.66
	MCDropout [22]	92.19	49.02	46.40	**99.27**	92.08	-	-
	TrustScore [8]	92.19	55.70	38.10	98.76	88.47	-	-
	TCP [9]	92.19	44.94	49.94	99.24	92.12	**99.77**	0.00
	TCP†	92.19	45.07	49.89	99.24	92.12	97.88	0.00
	SS	92.19	**44.69**	**50.28**	99.26	**92.22**	98.46	**28.04**

Table A2: Performance comparison between $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$ and $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$.

⟨O, C⟩+ Loss	z	Acc↑	FPR-95%-TPR↓	AUPR-Error↑	AUPR-Success↑	AUC↑	TPR↑	TNR↑
⟨ViT, ViT ⟩+ SS	$\frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$	83.90	80.89	10.31	92.90	73.31	88.44	35.64
	$\boldsymbol{w}^\top \boldsymbol{x}^{out}+b$	83.90	80.42	10.36	92.78	72.93	92.40	27.01
⟨ViT, RSN⟩+ SS	$\frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$	68.72	78.09	20.94	85.30	74.20	67.96	67.80
	$\boldsymbol{w}^\top \boldsymbol{x}^{out}+b$	68.72	76.65	20.75	85.82	75.00	73.54	62.66
⟨RSN, ViT⟩+ SS	$\frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$	83.90	88.63	11.75	89.87	64.11	95.41	10.48
	$\boldsymbol{w}^\top \boldsymbol{x}^{out}+b$	83.90	88.39	11.65	90.06	64.56	97.95	5.45
⟨RSN, RSN⟩+ SS	$\frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$	68.72	85.60	22.50	81.92	68.08	80.44	40.85
	$\boldsymbol{w}^\top \boldsymbol{x}^{out}+b$	68.72	85.59	22.42	82.02	68.33	80.03	41.82

A.4 Linear Function vs. Signed Distance

Although the signed distance z, i.e., $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$, leads to a geometric interpretation as shown in Fig. 2a, the main-stream models [5, 6, 7] use $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$. Therefore, we provide the corresponding comparative results in Table A2, which are generated by the proposed loss taking the output of the linear function as input. In this analysis, the combination ⟨ViT, ViT ⟩is used and $\alpha^+ = 1, \alpha^- = 3$.

As shown in Table A2, both $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$ and $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$ yield similar performance on metrics FPR-95%-TPR, AUPR-Error, AUPR-Success, and AUC. On the other hand, TPR and TNR are moderately different between $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}^{out}+b}{\|\boldsymbol{w}\|}$ and $z = \boldsymbol{w}^\top \boldsymbol{x}^{out} + b$, when α^+ and α^- are fixed. This implies that TPR and TNR are sensitive to $\|\boldsymbol{w}\|$.

A.5 Separability between Distributions of Correct Predictions and Incorrect Predictions

We access the separability between the distributions of correct predictions and incorrect predictions from a probabilistic perspective. There are two common tools to achieve the goal, i.e., Kullback–Leibler (KL) divergence [30] and Bhattacharyya distance [31]. KL divergence is used to measure the difference between two distributions [32, 33], while Bhattacharyya distance is used to measure the similarity of two probability distributions. Given the distribution of correct predictions $\mathcal{N}_1(\mu_1, \sigma_1^2)$ and the distribution of correct predictions $\mathcal{N}_2(\mu_2, \sigma_2^2)$, we use the averaged KL divergence, i.e., $\bar{d}_{KL}(\mathcal{N}_1, \mathcal{N}_2) = (d_{KL}(\mathcal{N}_1, \mathcal{N}_2) + d_{KL}(\mathcal{N}_2, \mathcal{N}_1))/2$, where $d_{KL}(\mathcal{N}_1, \mathcal{N}_2) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$ is not symmetrical. On the other hand, Bhattacharyya distance is defined as $d_B(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{4} \ln \left(\frac{1}{4} \left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + 2 \right) \right) + \frac{1}{4} \left(\frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \right)$. A larger $\bar{d}_{KL}$ or d_B indicates that the two distributions are further away from each other.

Table A3: Separability of distributions w.r.t. correct predictions and incorrect predictions.

$\langle$**O, C**$\rangle$	**Loss**	$\bar{d}_{KL} \uparrow$	$d_B \uparrow$
$\langle$ViT, ViT $\rangle$	CE	0.5139	0.0017
	Focal [15]	0.5212	0.0026
	TCP [9]	0.7473	0.0297
	SS	1.2684	0.0947
$\langle$ViT, RSN$\rangle$	CE	0.5131	0.0016
	Focal [15]	0.5017	0.0002
	TCP [9]	0.9602	0.0559
	SS	1.3525	0.1065
$\langle$RSN, ViT$\rangle$	CE	0.5267	0.0033
	Focal [15]	0.5121	0.0015
	TCP [9]	0.5528	0.0066
	SS	0.7706	0.0337
$\langle$RSN, RSN$\rangle$	CE	0.5077	0.0010
	Focal [15]	0.5046	0.0006
	TCP [9]	0.7132	0.0265
	SS	0.9539	0.0565

The separabilities are reported in Table A3. We can see that the proposed loss leads to larger separability than the other three loss functions. This implies that the proposed loss is more effective to differentiate incorrect predictions from correct predictions, or not vice versa.