Well-classified Examples are Underestimated in Classification with Deep Neural Networks

Guangxiang Zhao [†,‡], **Wenkai Yang** [♮], **Xuancheng Ren** [‡], **Lei Li** [‡], **Yunfang Wu** [‡], **Xu Sun** [‡,♯*]

[†] Institute for Artificial Intelligence, Peking University
[‡] MOE Key Laboratory of Computational Linguistics, School of EECS, Peking University
[♮] Center for Data Science, Peking University
[♯] Beijing Academy of Artificial Intelligence
{zhaoguangxiang, renxc, wuyf, xusun}@pku.edu.cn, {wkyang, lilei}@stu.pku.edu.cn

Abstract

The conventional wisdom behind learning deep classification models is to focus on bad-classified examples and ignore well-classified examples that are far from the decision boundary. For instance, when training with cross-entropy loss, examples with higher likelihoods (i.e., well-classified examples) contribute smaller gradients in back-propagation. However, we theoretically show that this common practice hinders representation learning, energy optimization, and margin growth. To counteract this deficiency, we propose to reward well-classified examples with additive bonuses to revive their contribution to the learning process. This counterexample theoretically addresses these three issues. We empirically support this claim by directly verifying the theoretical results or significant performance improvement with our counterexample on diverse tasks, including image classification, graph classification, and machine translation. Furthermore, this paper shows that we can deal with complex scenarios, such as imbalanced classification, OOD detection, and applications under adversarial attacks, because our idea can solve these three issues. Code is available at https://github.com/lancopku/well-classified-examples-are-underestimated.

1 Introduction

In common practice, classification with deep neural networks (DNNs) down-weights the contribution from well-classified examples. DNNs have achieved leading performance in mainstream classification tasks (He et al. 2016; Kipf and Welling 2016; Vaswani et al. 2017; Devlin et al. 2019). Usually, the training of DNNs relies on optimizing the designed metrics between the target and the prediction through back-propagation (Rumelhart, Hinton, and Williams 1986). Mean-Square Error (MSE) calculates a quadratic distance between the target and the probabilistic prediction of each example (Rumelhart, Hinton, and Williams 1986). Cross-Entropy (CE) loss measures the distance between the target distribution and the probability distribution (Baum and Wilczek 1988). CE loss is preferred as compared to MSE since CE loss encourages accurate predictions by bringing steep gradients to well-classified examples (Baum and Wilczek 1988; Goodfellow, Bengio, and Courville 2016).

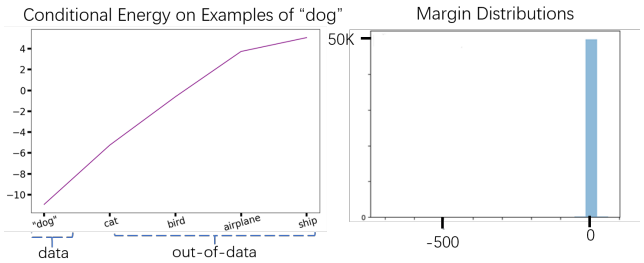

Figure 1: Illustration of energy and margins of CE loss on CIFAR-10. Left: averaged conditional energy $E(y|x)$ for examples of class "dog". Right: margin distributions of the trained classification models, there are some examples that have negative margins, most examples are around the decision boundary.

Therefore, CE loss shows better generalization ability due to the enlarged steepness (Solla, Levin, and Fleisher 1988). During training with CE loss, well-classified examples contribute less to the gradient as compared to bad-classified ones. The wisdom behinds the operation of overlooking well-classified examples is well-classified examples have relatively less information in the learning. The improved variants of CE loss still comply with such wisdom (Liu et al. 2016; Szegedy et al. 2016; Lin et al. 2017).

We doubt the above practice with the following three facts: (1) Recent studies in imbalanced learning indicate that down-weighting the learning of relatively well-classified data-rich classes severely impairs the representation learning (Kang et al. 2020; Zhou et al. 2020). These studies inspire us to reflect on whether this is also the case at the sample level, and we validate that down-weighting the learning of well-classified samples also decreases the performance (Table 3). (2) As to Energy-Based Models (EBM) (LeCun et al. 2006), a sharper energy surface is desired.[1] However, we find that energy surface trained with CE loss is not sharp, as plotted in Figure 1. The possible reason is that CE loss has insuffi-

*Corresponding Author

[1] Refer to the talk "The Future is Self-Supervised Learning" by Yann LeCun at ICLR 2020

cient power to push down the energy of the positive examples as long as it is lower than the energy of negative examples. Our validation in Figure 5 shows that up-weighting the well-classified examples returns us a sharper surface. (3) As to classification, it is acknowledged that building classification models with large margins lead to good generalization (Bartlett 1997; Jiang et al. 2019) and good robustness (Elsayed et al. 2018; Matyasko and Chau 2017; Wu and Yu 2019), but we find that the learning with CE loss leads to smaller margins (as plotted Figure 1). The reason may be the limited incentive to enlarge margins further since well-classified examples are less optimized. Our results in Figure 6 and 7 show up-weighting classified samples enlarges the margin and helps to improve the adversarial robustness.

Contributions: We systematically study the role of well-classified examples in classification learning with DNNs, and the results challenge the common practice. First, we theoretically identify issues of CE loss with back-propagation in terms of the learning of representations, the learning of energy functions, and the growth speed of margins. (Refer to § 2). Second, we propose Encouraging Loss (EL), which can be viewed as a counterexample to the common practice since it up-weights the learning of well-classified examples compared to CE loss. Besides, we theoretically demonstrate that by paying attention to well-classified examples, learning with EL revives the representation learning from the part of well-classified examples, reduces the energy on the data manifold, and enlarges the classification margin. (Refer to § 3). Third, we conduct extensive experiments to empirically verify the effectiveness of paying more attention to the well-classified instances for training canonical and state-of-the-art models. Results on image recognition and machine translation based on heterogeneous data types show that the counterexample can consistently improve learning performance. We also empirically verify that well-classified examples regarding representation learning, energy, and margins. Further analysis shows that enhancing the learning of well-classified examples improves models' capacity under complex scenarios, such as imbalanced classification, OOD detection, and application under adversarial attacks. (Refer to § 4).

2 Exploring Theoretical Issues of CE Loss

2.1 Setup and Notations

Classification In classification tasks, we have input data $\boldsymbol{x} \in \mathbb{R}^D$ and a label $y \in \mathbb{Y}$ which belongs to one of all K classes, and use $\mathbb{Y} = \{1, 2, ..., K\}$ to denote the set of all class indices. We aim to learn a parametric function $f_{\boldsymbol{\theta}}(\cdot)[\mathbb{Y}]$ that predicts K-dim logits (before normalization) for the data $\boldsymbol{x}$, i.e., $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^K$. Let $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$ denotes the y-th value of the predicted K-dim logits $f_{\boldsymbol{\theta}}(\boldsymbol{x})[\mathbb{Y}]$, and $p_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$ is the normalized probability for the class y. We adopt the general $\mathrm{softmax}$ normalization which transforms the logit $f(\boldsymbol{x})[y]$ to the probability $p(\boldsymbol{x})[y]$ since it can generalize to both two-class and multi-class classification: $p_{\boldsymbol{\theta}}(\boldsymbol{x})[y] = \mathrm{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])) = \frac{exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])}{\sum_{y' \in \mathbb{Y}} exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'])}$.

CE loss Typically, the parametric classifier $f_{\boldsymbol{\theta}}(\boldsymbol{x})[\mathbb{Y}]$ is estimated using Maximum Likelihood Estimation (MLE), which is equivalent to minimizing the Cross-Entropy (CE) between the predicted probability $p(\boldsymbol{x})[\boldsymbol{y}]$ and the signal that whether the class $\boldsymbol{y}$ is the true label. When we get probabilities through the $\mathrm{softmax}$ normalization among all classes, we can simplify CE loss as minimizing negative log-likelihood (NLL) of $-\log p_\theta(\boldsymbol{x})[y]$, in which y is the true label class for the input $\boldsymbol{x}$. The NLL loss is:

$$\mathcal{L}_{NLL} = -\log p_\theta(y \mid \boldsymbol{x}) = -\log p_\theta(\boldsymbol{x})[y]. \tag{1}$$

In Eq. (1), we get the predicted probability from the model $\boldsymbol{\theta}$ and want to maximize the log probability of the target class y. We use the term **steepness of loss** to denote $\partial\mathcal{L}/\partial p$. For the NLL loss, the steepness of loss is $-\frac{1}{p}$, which means incorrect predicted examples with small p are learned with priority, i.e., embodying a sharper loss and a larger steepness of loss. In this section, we refer CE loss to NLL loss and discuss gradients regarding NLL loss and $\mathrm{softmax}$ normalization. However, our results can also easily generalize to BCE loss with sigmoid since the opposite class of BCE loss in case of the second class in NLL loss, and the derivative of the sigmoid is the same as $\mathrm{softmax}$. In the NLL loss, we can only consider gradients from the index of the label class y. For simplicity, we use p to denote the correctness of predictions.

Back-propagation

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}} &= \sum_{y' \in \mathbb{Y}} \frac{\partial\mathcal{L}}{\partial\sigma_{\boldsymbol{\theta}}(x)[y']} \frac{\partial\sigma(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'])}{\partial\boldsymbol{\theta}} \\ &= \sum_{y' \in \mathbb{Y}} \frac{\partial\mathcal{L}}{\partial\sigma_{\boldsymbol{\theta}}(x)[y']} \frac{\partial\sigma(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'])}{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']}{\partial\boldsymbol{\theta}}.\end{aligned} \tag{2}$$

In Eq. (2), gradients depend on the loss function $\mathcal{L}$, logits normalization function σ and the current model $f_{\boldsymbol{\theta}}$.

2.2 Limitation of CE loss in Three Aspects

Normalization function brings a gradient vanishing problem to CE loss and hinders the representation learning At the beginning of the introduction of back-propagation to train deep neural networks, the loss function for back-propagation is MSE with measures the L_2 distance between probabilities and labels (Rumelhart, Hinton, and Williams 1986). However, the steepness of MSE gets to zero when the prediction gets close to the target. Baum and Wilczek (1988) introduces CE loss to back-propagation and points out it addresses the above issue in MSE and makes predictions more accurate. Combining the derivative of NLL loss and the derivative of the normalization, gradients for the model parameters with the CE loss are:

$$\frac{\partial\mathcal{L}_{NLL}}{\partial\boldsymbol{\theta}} = (p-1)\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]}{\partial\boldsymbol{\theta}}. \tag{3}$$

The normalization function brings the gradient saturation back as the prediction becomes correct. Since DNNs are regarded as a pipeline for extracting features (Zeiler and Fergus 2014; Tenney, Das, and Pavlick 2019), and it is in line with our perception that well-classified examples share pipeline with other examples, gradient vanishing hinders the part of representation learning from well-classified examples.

CE loss has insufficient power in reducing the energy on the data manifold Energy-based models (EBM)

(LeCun et al. 2006) calculate the probability density of $p(x)$ for $x \in \mathbb{R}^D$ by: $p(x) = \frac{\exp(-E_{\boldsymbol{\theta}}(x))}{\int_x \exp(-E_{\boldsymbol{\theta}}(x))}$ Here, we re-interpret the classifier as a conditional EBM $p_\theta(y \mid \boldsymbol{x}) = \frac{\exp f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]}{\sum_{y' \in \mathbb{Y}} \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'])}$, in which the conditional energy is $E_\theta(y \mid \boldsymbol{x}) = -f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$. Thus, the CE loss in Eq. (1) can be written as:

$$\mathcal{L}_{NLL} = E_\theta(y \mid \boldsymbol{x}) + log[\exp(-E_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})) + \sum_{y' \neq y} \exp(-E_{\boldsymbol{\theta}}(y' \mid \boldsymbol{x}))]. \quad (4)$$

Minimizing the CE loss can push up the energy out of the data $E_\theta(y' \mid \boldsymbol{x}), y' \neq y$, but for the energy on the data manifold $E_\theta(y \mid \boldsymbol{x})$, although the nominator of the softmax function pulls that energy down, numerator which contains that term pushes the energy on the data up. Therefore, the learning with CE loss gets into a dilemma and has insufficient power to reduce the data's energy.

CE loss is not effective in enlarging margins Previous studies prove that large minimum margin (Bartlett 1997; Bartlett, Foster, and Telgarsky 2017; Neyshabur, Bhojanapalli, and Srebro 2018) or large overall margins (Zhang and Zhou 2017; Jiang et al. 2019) on the training set indicate good generalization ability.. Though the margin $\gamma(\boldsymbol{x}, y) = f_{\boldsymbol{\theta}}(\boldsymbol{x})[y] - max_{y' \neq y} f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']$ is defined on the logits, since the softmax function generates probabilities against each other and has the exponential term that mimics the max operation, a larger likelihood is likely to lead to a larger margin. However, as we have deduced in Eq. (3), CE loss is not good at increasing the likelihood when it becomes larger. Hence it is likely to be limited in increasing the margin. Following previous work (Wang et al. 2018; Cao et al. 2019; Menon et al. 2021), we view the gap $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y] - f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']$ between the logit at the label position and the logit at any other position $y' \neq y$ as the approximated margin. Note that the NLL loss can then be written as $\mathcal{L}_{NLL} = log[1 + \sum_{y' \neq y} \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'] - f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])]$. We use A to denote $\exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'] - f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])$, the gradients of the NLL loss w.r.t. the parameter $\boldsymbol{\theta}$ is:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \boldsymbol{\theta}} = \frac{\sum_{y' \neq y} A(\frac{\partial (f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'] - f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])}{\partial \boldsymbol{\theta}})}{1 + \sum_{y' \neq y} A}. \quad (5)$$

The above formula interprets the training procedure of CE loss as increasing the logit for the label $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$, but decreasing the logits for other classes $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']$, thus it enlarges the gap, and the standard margin $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y] - \max_{y' \neq y} f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']$ is likely to be larger. However, when the prediction gets close to the target during training, A gets close to 0, but the numerator has a constant 1, so the incentive for further enlarging the margin gets close to 0. Therefore, CE loss is not effective in enlarging margins to a certain extent.

3 Gaining from Reviving the Learning of Well-classified Examples

In this section, we propose a counterexample **Encouraging Loss** (EL) which increases the relative importance of well-classified examples in optimization compared to that in CE loss. We first define EL and then demonstrate that it can mitigate issues of CE loss mentioned above.

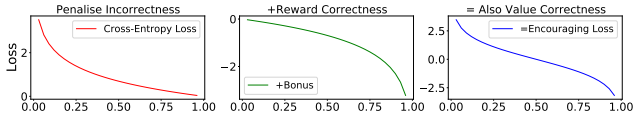

Figure 2: Illustration of the encouraging loss, which is a counterexample. CE loss: $-logp$; Bonus: $log(1-p)$; Encouraging loss: CE loss + bonus. Bonus strengthens the learning of well-classified examples.

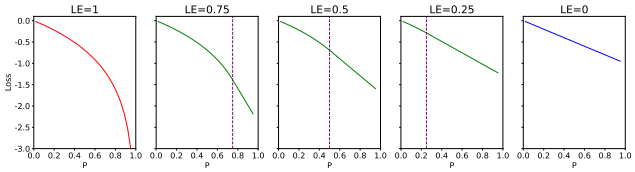

Figure 3: These are variations of bonuses. Left: normal bonus $log(1-p)$; Others: conservative bonus. LE is the position where **L**og curve **E**nds in conservative log bonus, loss curve continues after LE with the tangent at $p = LE$ of the log curve. These variations gradually increase steepness for well-classified examples from right to left. The original CE loss is the case $bonus = constant$.

3.1 The Counterexample: Encouraging Loss

As plotted in the Figure 2, encouraging loss is the addition of CE loss and an additional loss (we term it as a bonus) that makes the loss steeper again when p goes higher. The **normal bonus** is the mirror flip of the CE loss: $bonus = log(1-p)$, we clamp the value in the logarithm by a small epsilon (e.g. 1e-5) to avoid numerical instability. And the encouraging loss with normal bonus is:

$$\mathcal{L}_{EL} = -\log p_{\boldsymbol{\theta}}(\boldsymbol{x})[y]) + log(1 - p_{\boldsymbol{\theta}}(\boldsymbol{x})[y]) \quad (6)$$

We name it **encouraging loss** since it encourages the model to give more accurate predictions by rewarding these near-correct predictions with a bonus.

As long as the additional bonus is concave, its steepness for larger p is larger, indicating that the EL with that bonus pays more attention to the well-classified examples than the CE loss. To study the relative importance of the learning for well-classified examples compared to other examples, we can adjust the relative steepness of the additional bonus. We design many types of **conservative bonus** that approximate the normal bonus but are more conservative and show them in Figure 3. These variants end the log curve early in the high-likelihood region and replace the log curve with the tangent of the endpoint. The relative importance of optimization for well-classified examples in encouraging loss with these bonuses is larger than CE and gradually increases from the right to the left. The bonus can also be designed to be more aggressive than the normal bonus.

3.2 The counterexample can solve the identified issues in CE loss

Taking the normal bonus as a typical example, we analyze the benefits of encouraging loss.

Encouraging loss enhances the representation learning from the part of well-classified examples The steepness of the encouraging loss is $-\frac{1}{p} - \frac{1}{1-p}$, so the gradients are:

$$\frac{\partial \mathcal{L}_{EL}}{\partial \boldsymbol{\theta}} = -1 \cdot \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]}{\partial \boldsymbol{\theta}}. \tag{7}$$

In contrast to gradients of NLL/CE loss in Eq. (3), here gradients are independent of the likelihood p (this is a bit like RELU in DNNs (Glorot, Bordes, and Bengio 2011)). As to the EL with conservative bonus, since the bonus is concave, gradients are also larger for well-classified examples. For these conservative variants, gradients for probability before LE (<1) are the same as that when LE=1, but the gradients after LE (<1) are $\left(-\frac{1}{p} - \frac{1}{1-LE}\right) * [p\,(1-p)]$ times smaller, but they do not scale linearly as CE loss does.

Encouraging loss makes the model have smaller obstacles for reducing the energy on the data The conditional energy form of the EL is:

$$\mathcal{L}_{EL} = E_\theta(y \mid \boldsymbol{x}) - log[\sum_{y' \neq y} \exp(-E_{\boldsymbol{\theta}}(y' \mid \boldsymbol{x}))]. \tag{8}$$

The difference between it and the conditional energy form of the CE loss in Eq. (4) lies in the second term. Notice that training with EL does not need to push up the energy on the data to minimize the second term, so there is more incentive to lower the energy on the data. Although conservative bonus $< log1$ and $> log(1-p)$ do not remove the obstacle in the second term, the obstacle will be smaller than CE loss.

Encouraging loss makes margins grow faster The margin perspective of gradients w.r.t EL is:

$$\frac{\partial \mathcal{L}_{EL}}{\partial \boldsymbol{\theta}} = \frac{\sum_{y' \neq y} A(\frac{\partial (f_{\boldsymbol{\theta}}(\boldsymbol{x})[y'] - f_{\boldsymbol{\theta}}(\boldsymbol{x})[y])}{\partial \boldsymbol{\theta}})}{\sum_{y' \neq y} A}. \tag{9}$$

Now the growth speed for the margin is $\frac{1+\sum_{y' \neq y} A}{\sum_{y' \neq y} A}$ times faster than that during the training with the CE loss. When the model is getting better, the exponent of negative gap A gets close to 0, the ratio becomes large and helps further increase the margin. The EL with a conservative bonus has a smaller ratio, but the ratio is still large than 1.

4 Practical Effect of Encouraging the Learning of Well-classified Examples

This section analyzes the practical effect of encouraging learning well-classified examples by applying the counterexample to various classification tasks and settings.

4.1 Experiment Setup

In here we briefly clarify the experiment setup, please refer to the Appendix[2] and the code for more details. For reliability, each result is the mean result of 5 different runs with error bars. Especially, on graph datasets, each run contains 50 different train, valid, test splits of the data (proportion is 0.8, 0.1, 0.1 respectively) since a recent study indicates that different dataset splits largely affect the test performance (Shchur et al. 2019). For other tasks, we use their official data splits.

Image Recognition It is a typical application of multi-class classification. In these tasks, we need to predict the category an image belongs to. We adopt four tasks MNIST (LeCun and Cortes 2010), CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), and ImageNet (Russakovsky et al. 2015), the descriptions of the dataset are in Appendix. For training, we borrow code from repositories with good reproduced accuracy and keep all their default settings unchanged. Specifically, we train the CNN model from Liu et al. (2016) on MNIST, train ResNet-50 (He et al. 2016) and EfficientNet-B0 (Tan and Le 2019) on CIFAR-10 and CIFAR-100 using the code by Narumiruna, train the ResNet-50 on ImageNet with the example code from PyTorch, train the EfficientNet-B0 on ImageNet using the code from timm (Wightman 2019). We choose ResNet-50 and EfficientNet-B0 because they are canonical and SoTA parameter-efficient models, respectively. For evaluation, we report the best top-1 accuracy on the test set following common practice.

Graph classification Typical applications of graph classification are to binary classify the functionality of the graph-structured biological data, we do the experiments on PROTEINS (1113 graphs of protein structures) (Dobson and Doig 2003; Borgwardt et al. 2005) and NCI1 (4110 graphs of chemical compounds) (Wale, Watson, and Karypis 2008). The model is the node feature learning model GCN (Kipf and Welling 2016) with the pooling method SAGPooling (Lee, Lee, and Kang 2019). We report test accuracy on the early stopping model with the best valid accuracy.

Machine translation In this task, we need to sequentially select a word class from the vocabulary, containing tens of thousands of word classes. We perform experiments on IWSLT De-En ($160K$ training sentence pairs) and IWSLT Fr-En ($233K$ training sentence pairs). The base model is Transformer (Vaswani et al. 2017), and the evaluation metric is BLEU which calculates how many N-grams (N-gram is a contiguous sequence of N classification predictions) both exist in the predicted sequence and the generated sequence (Papineni et al. 2002). We adopt most settings from fairseq, including training and evaluation. The only modification is that we tune the best hyper-parameters for the default loss (CE loss with label smoothing) and then use them for training the model with encouraging loss for fair comparisons.

Imbalanced classification We perform experiments on a large-scale natural imbalanced classification dataset–iNaturalist 2018, which has $8,142$ classes, $438K$ training samples, and $24K$ valid samples. The setting for training and evaluation is the same as Kang et al. (2020), including separately evaluating results on subsets of "many-shot", "medium-shot", and "few-shot".

[2]Please refer to https://arxiv.org/abs/2110.06537 for Appendix.

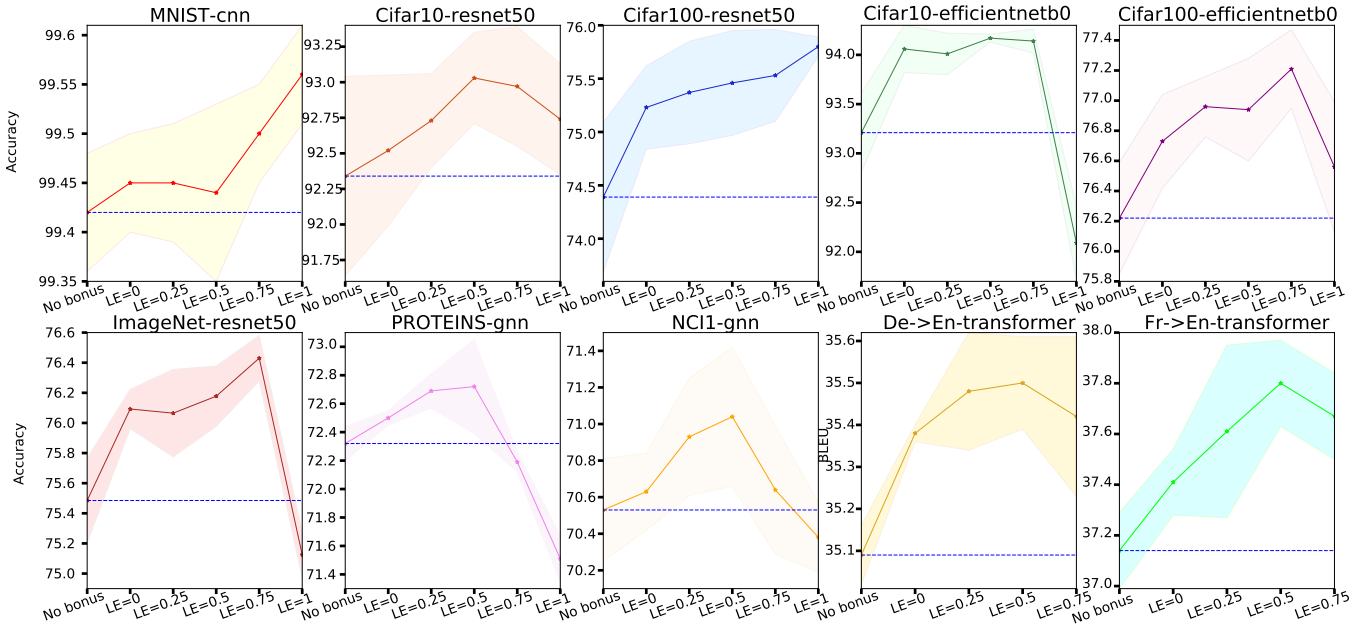

Figure 4: These figures plot performance under various settings. The colored areas denote the standard deviation of the 5 runs. We can see that enhancing the learning of well-classified examples can improve performance.

Setting	MNIST	C10-r50	C10-eb0
CE	99.42±0.06	92.34±0.70	93.21±0.40
EL	99.56±0.05	92.97±0.42	94.24±0.17
Setting	C100-r50	C100-eb0	Img-r50
CE	74.39±0.70	76.22±0.37	75.49±0.28
EL	75.80±0.09	77.21±0.26	76.43±0.15
Setting	Img-eb0	Proteins	NCI1
CE	77.80±0.15	72.32±0.12	70.53±0.28
EL	78.28±0.13	72.76±0.18	71.04±0.38
Setting	De-En	Fr-En	
CE	35.09±0.07	37.10±0.06	
EL	35.50±0.11	37.73±0.17	

Table 1: We summarize the results of CE and EL on various tasks. Encouraging the Learning of well-classified examples brings consistent improvements. We abbreviated the names, e.g., Img refers to ImageNet.

4.2 Improving Classification Performance

Image recognition of multi-class classification on the pixel data We plot the results in Figure 4 and summarize the improvements in Table 1. For EfficientNet-B0 on Imagenet, we directly run the baseline and encouraging loss with a conservative bonus (LE=0.75) to save energy since EfficientNet is not time-efficient. From all results, we point out that strengthening the learning of well-classified examples with an additional bonus can indeed help improve the accuracy. Take ImageNet as an example, by strengthening the learning of well-classified examples, the accuracy of the canonical model ResNet-50 is improved by 0.94 points, and the accuracy of the SoTA parameter efficient model EfficientNet-B0 can also be improved from 77.8 to 78.28. We want to point out that the improvement by EL is actually remarkable compared with other methods. For example: On ImageNet (1.28M images), we improved the valid accuracy of EfficientNet-b0 from 77.8 to 78.3. Noisy Student (Xie et al. 2020) is the SoTA training method for EfficientNet-b0, which enhances the performance from 77.3 to 78.1 while it relies on training on a much larger dataset, google's JFT-300M. Mixup (Zhang et al. 2018) improves the ResNet50 by 0.2 with the same number of epochs (90 epochs) and improves by 1.5 with more than 2× epochs, while we can improve the ResNet-50 by 0.9 without training extra epochs.

Binary graph classification on graph-structured data We can see from the Figure 4 that strengthening the learning of well-classified examples with a conservative bonus can bring an increase in the accuracy of 0.44 on Proteins and accuracy of 0.51 on NCI1.

Setting	Cifar10-resnet50	Cifar100-resnet50
CE	92.34±0.70	74.39±0.70
CE (2xLR)	91.53±0.19	73.57±0.66
EL	92.74±0.38	75.80±0.09
EL (0.5xLR)	93.69±0.25	76.37±0.29

Table 2: Training with EL (LE=1) can benefit from reducing the amount of gradients since the overall gradients of EL are larger than CE in theory.

Machine translation of sequential multi-class classification on the text data As we can see from the last two sub-figures of Figure 4, on machine translation, rewarding correct predictions can bring BLEU score improvement of 0.41 on De-En translation and 0.63 on Fr-En translation. We show in the Appendix that our improvements are additive label smoothing (Szegedy et al. 2016) and on par with it.

Discussion about the hyper-parameter LE and the conservative bonus Results from Figure 4 indicate that in many conservative settings (LE$\leq$0.5) where encouraging loss already pays more attention to well-classified examples than CE loss, EL with conservative bonuses consistently improves the performance. However, encouraging loss with a steeper bonus can not further improve the accuracy of deep classification models in some settings.

a. Why does a steep bonus like LE=1 does not bring improvements in some scenarios? The reason is that CE loss implicitly decreases the learning rate along with the training as the gradient norm of CE loss decays during training, and existing methods for optimization which adapt to CE loss should be modified. In our experiments, we choose to adopt the best setting in baselines for EL, which may not be the most suitable for encouraging loss. To verify this, we first show in Table 2 that when we enlarge the learning rate for CE loss, accuracy also drops. Then, we find that re-normalizing the gradients of encouraging loss by decreasing the global learning rate can help learn better from well-classified examples. For example, we can continue improving the accuracy gap between CE loss and encouraging loss to 1.35 on Cifar10-resnet50 and 1.98 on Cifar100-resnet50, respectively.

b. Every performance peak seems to occur between LE=0.5 and LE=0.75. These two settings strengthen the training for well-classified examples while not much changing the overall gradient norm. For example, in our preliminary experiments, we observe that on Transformer translation, LE=1 makes the norm of gradients $> 5\times$ larger and brings 1-2 BLEU drop, but the norm is only $< 1.7\times$ larger than CE for LE=0.75. Thus, when we directly use the original training hyper-parameters for CE loss for a fair comparison, LE=0.5 and LE=0.75 are good default choices.

c. How to select the additional hyper-parameter LE? First, practitioners can choose LE=0.5 as it works consistently better than CE loss on all tasks with existing mainstream systems we tried. They can select higher LE for models that are stable to train (e.g., ResNet). Second, Table 2 shows that we can benefit from modifying the original system to use higher LE (e.g., LE=1), which yields better results.

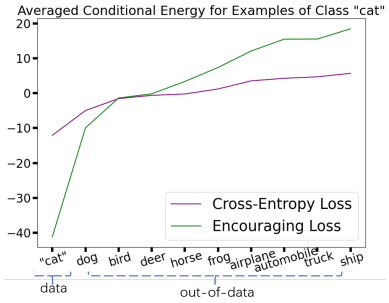

Figure 5: Conditional energy $E(y \mid x)$ averaged over the examples(X_{y*}) of the target class (here it is 'cat'), the energy around the data ($y = y*$) become sharper with the help of EL.

Easy to apply Besides all the promising results shown above, our method has another advantage in that it can be widely applicable as a direct plug-in module without changing all the original hyper-parameters.

4.3 Addressing the Issues of CE Loss in Practice

In this subsection, we demonstrate that strengthening the learning of well-classified examples addresses the issues that CE loss faces as we discussed in §2.2. Due to the space limit, we compare results between CE loss and EL with a normal bonus for training ResNet-50 on CIFAR-10. Please refer to the Appendix for more results in other settings.

Representation Learning Besides the overall accuracy improvement illustrated in §4.2, we also perform experiments to evaluate the learned representations by two losses. Specifically, we first use CE loss and EL to train the whole model, and then only train the re-initialized output layer with CE loss but fix the representations. The procedure is similar to Kang et al. (2020) that decouples the training into the representation learning phase and classification phase. We observe that representation learning with EL achieves an accuracy of 92.98 ± 0.01, but representation learning with CE loss only gets an accuracy of 91.69 ± 0.04.

Energy optimization We plot the conditional energy $E(y \mid x)$ averaged over the examples of class "cat" in Figure 5. We can see that the energy around the data becomes sharper with the help of encouraging loss since it pushes down the energy on the data more than CE loss.

Growth of margins We can see from Figure 6 that margins of encouraging loss are several times larger than margins of CE loss. These results demonstrate that learning well-classified examples with the additional bonus greatly improves the classification models by enlarging margins, which further makes the model has great generalization and robustness as we will discuss in the following.

4.4 Coping with Complex Application Scenarios

This section shows that enhancing the learning of well-classified examples by EL can cope well with three complex application scenarios since this idea mitigates the three

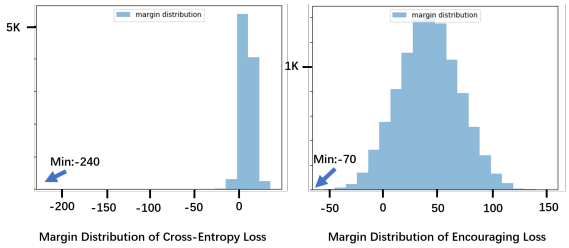

Figure 6: Left: Margins of CE loss distributed among -240 to 50, centered by 15. Right: Margins of EL distributed among -70 to 170, centered by 50. The learning of well-classified examples help double the margins in classification.

Method	iNaturalList2018			
	Overall	Many	Medium	Few
CE loss	64.3	74.1	65.9	59.8
+ Conservative Bonus (LE=0.5)	65.3	74.3	66.6	61.2
+ Normal Bonus	65.8	74.4	66.6	62.4
+ Aggressive Bonus	66.3 (+2.0)	**75.1** (+1.0)	67.4(+1.6)	62.6(+2.8)
Decoupling Reps&Cls (CRT)	64.9	71.4	65.9	61.9
+ Normal Bonus	66.8(+1.9)	71.7(+0.3)	67.6(+1.7)	64.6(+2.7)
Deferred Re-weighting	68.1	71.0	68.3	67.1
+ Normal Bonus	**70.3**(+2.2)	69.0(-2.0)	**70.1**(+1.8)	**70.9**(+3.8)

Table 3: Comparison between CE loss and encouraging loss (add a bonus to CE loss) on imbalanced classification dataset iNaturalist 2018, the average standard deviation for results on iNaturalList 2018 is 0.4. Additional bonuses that revive the learning of well-classified examples bring improvements both on CE loss and its advanced methods.

issues.

Imbalanced classification In imbalanced classification, we improve the classification performance of rare classes with the help of representation learning from samples of data-rich classes (Kang et al. 2020). Because enhancing the learning of easy samples also enhances their representation learning, we believe this property benefits imbalanced learning. We conduct validation experiments on the iNaturalist 2018 dataset, and the results are in Table 3. We find that encouraging the learning of well-classified samples makes models' performance outperform that trained with CE loss, both for the case with conservative bonus or the case with aggressive bonus (only to reward the highly well-classified samples of $p > 0.5$ with the normal bonus). We can also combine our idea with other advanced methods (Cao et al. 2019; Kang et al. 2020). For the additive experiment with "Decoupling classifier and representation learning" (Kang et al. 2020), in the representation learning phase, we not only remain the learning of data-rich classes as they do but also revive the learning of well-classified examples. In the classifier learning phase, we keep all the settings unchanged. Our method improves them by 1.9 points, **which empirically verify that the traditional re-weighting at the sample level (CE loss down-weights the importance of well-classified samples) is also harmful to the representation learning**.

OOD detection The task of OOD detection is to detect whether a new sample belongs to the distribution of training data. We can improve the performance of OOD detection

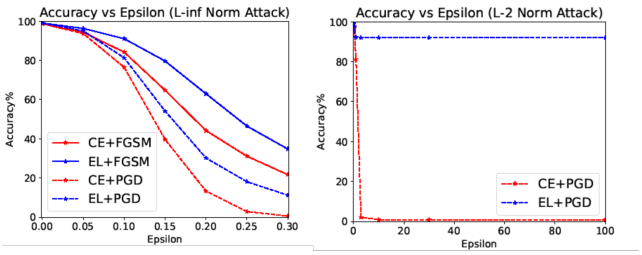

Figure 7: Accuracy of CE loss and encouraging loss under adversarial attacks of FGSM and PGD.

Setting	MNIST vs. F. MNIST		Img vs .iNautralist2018	
Metric	AUROC↑	FPR95↓	AUROC↑	FPR95↓
CE	95.68	20.92	76.54	75.40
EL	98.04	8.69	78.41	71.86

Table 4: EL significantly improves OOD detection performance. We use F. MNIST to denote Fashion MNIST. Higher AUROC and lower FPR95 are better.

since EL sharpens the energy surface around the data. For OOD detection, we use an indicator to detect whether samples are in the distribution (In-D) of the training set or out of the distribution (OOD). We use the minimum conditional energy $min_{y' \in \mathbb{Y}} E_{\boldsymbol{\theta}}(y' \mid \boldsymbol{x})$ as the indicator. We show in the Appendix that our indicator is better than the maximum probability and free energy (Liu et al. 2020). We conduct an experiment on a small-scale OOD detection task (In-D: MNIST, OOD: Fashion-MNIST) and a large-scale task (In-D: ImageNet, OOD: iNautralist2018). Samples in Fashion-MNIST are clothes, while samples in MNIST are digital numbers; samples in ImageNet are instances of objects, while samples in iNautralist2018 are instances of species (refer to Appendix for detailed settings). We can see from Table 4 that EL leads to better performance than CE loss. For example, we reduce the metric FPR95 from 8.10% to 0.95% on MNIST vs. Fashion MNIST. These results confirm that strengthening the learning of well-classified examples leads to discriminative energy distribution.

Robustness to adversarial attacks We have shown in theory and practice that paying more attention to the classified samples can significantly increase margins. This property is likely to increase the robustness of adversarial examples because the predictions of disturbed samples are likely to remain far away from the decision boundary. To verify this assumption, on MNIST, we use the FGSM (Goodfellow, Shlens, and Szegedy 2015), PGD (Madry et al. 2018) with L_∞ bound, and PGD with L_2 bound to construct adversarial samples to attack the models trained with CE loss and with EL, respectively. The results in Figure 7 validate our assumption that the model trained with EL is more robust on predictions when receiving a specific range of adversarial disturbances. We show in the Appendix that our idea can also improve robustness under other powerful attack methods and is additive with large margin variants of CE loss to further enhance robustness and margins.

Name	Origin Loss	+Mirror Bonus
CE loss + softmax	99.42 ± 0.06	99.56 ± 0.05
MSE + softmax	99.56 ± 0.05	99.66 ± 0.04
MSE + sigmoid	99.63 ± 0.04	99.69 ± 0.02

Table 5: Well-classified examples also help improve MSE, which is better than CE loss on MNIST.

4.5 Well-classified Examples Are also Underestimated in MSE

It is well-known to the community that CE loss outperforms MSE for training deep neural networks, and the superiority has also been proven to come from the steeper gradient of CE (Solla, Levin, and Fleisher 1988). However, Table 5 shows that MSE beat CE loss on MNIST. The possible reason is that CE loss suffers from a much higher sensitivity to bad-classified examples. Nevertheless, we find that by adding a mirror bonus $-(y*p)^2 - ((1-y)*(1-p))^2$ to MSE $(y-p)^2$ to encourage the learning of well-classified examples, the performance improves. This indicates that MSE also has the problem of underestimating well-classified examples.

4.6 The Idea is Additive with Improvements on CE Loss

We have shown in the paper that our idea improves MSE, CE loss, and variations of CE loss in imbalanced classification. We also discuss previous improvements in CE loss in the Appendix, and demonstrate our idea is additive with including focal loss (Lin et al. 2017), label smoothing (Szegedy et al. 2016), adversarial training (Goodfellow, Shlens, and Szegedy 2015), and large margin softmax (Liu et al. 2016).

5 Limitations

To facilitate future research, we analyze the difficulties and the possible solutions in this new area. First, EL with the normal bonus can not improve the accuracy of deep classification models in some settings. One possible reason is that CE loss reduces the overall gradient norm, and existing optimization which adapts to CE loss should be modified. We show in Table 2 that reducing the learning rate when employing encouraging loss is a possible solution. Second, models trained with EL tend to give more confident predictions, and the Expected Calibration Error (ECE) (Guo et al. 2017) is slightly increased. Nevertheless, we can mitigate the issue by combining EL with label smoothing (Szegedy et al. 2016) to ECE. We show in the Appendix that the combination of EL and label smoothing gets lower ECE than CE or CE with label smoothing.

6 Related work

Several studies are relevant to ours in different aspects.

Representation learning There are several studies that direly or indirectly mitigate the shortage of representation learning from well-classified examples. Relying on various augmentation techniques, contrastive learning is one of the methods which directly mitigates the issue and has made outstanding progress in learning representations (He et al. 2020; Chen et al. 2020; Grill et al. 2020). Kang et al. (2020) mitigate the representation learning issue caused by class-level re-weighting, which indirectly down-weight well-classified examples, by turning it off in representation learning. However, they do not improve CE loss in representation learning.

Using energy-based models to interpret the classifier Recent studies re-interpret the classifier as EBM (Grathwohl et al. 2020) or conditional EBM (Xie et al. 2016; Du and Mordatch 2019) for generative models. However, our focus is to separately investigate the energy on the data and the data optimized by CE loss in classification models.

Enlarging the margin A typical example of utilizing the idea of enlarging the minimum margin is the hinge loss (Suykens and Vandewalle 1999) that is proposed to help learn a considerate margin in SVMs (Cortes and Vapnik 1995) by focusing only on the data points close to the decision boundary. Recently, the idea of enlarging margins has been introduced to the case of CE loss (Liu et al. 2016; Wang et al. 2018; Li et al. 2018; Menon et al. 2021; Elsayed et al. 2018). Since these methods are based on logits adjustment before $\mathrm{softmax}$, they can be combined with our method by substituting the original CE loss with EL loss after $\mathrm{softmax}$. We show in the Appendix that our idea can be combined with their large margin idea.

7 Conclusion

In this paper, we theoretically and empirically show that well-classified examples are very helpful for the further improvement of deep classification models. To illustrate this finding, we first directly analyze the failure of common practice, which weakens the learning of these examples, and then verify the positive effect of our proposed counterexamples, which value the learning of well-classified examples.

Ethical statement

In this work, we do not introduce new datasets but use the existing widely-used datasets from their public release. As to societal harm, we do not facilitate social bias, but we can improve the performance of minority classes. Since algorithms based on our idea have a similar calculation time with CE loss, we do not facilitate cost more energy. Thus, our work does not have potential ethical considerations.

Acknowledgments

This work is partly supported by National Key R&D Program of China (2020AAA0105200), Natural Science Foundation of China (NSFC) No. 62176002, and Beijing Academy of Artificial Intelligence (BAAI). We thank all the anonymous reviewers for their constructive comments and Zhiyuan Zhang, and Sishuo Chen for helpful discussion in preparing the manuscript.

References

Bartlett, P. 1997. For Valid Generalization the Size of the Weights is More Important than the Size of the Network. In

Mozer, M. C.; Jordan, M.; and Petsche, T., eds., *Advances in Neural Information Processing Systems*, volume 9. MIT Press.

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30, 6240–6249. Curran Associates, Inc.

Baum, E.; and Wilczek, F. 1988. Supervised Learning of Probability Distributions by Neural Networks. In Anderson, D., ed., *Neural Information Processing Systems*. American Institute of Physics.

Borgwardt, K. M.; Ong, C. S.; Schönauer, S.; Vishwanathan, S.; Smola, A. J.; and Kriegel, H.-P. 2005. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20(3): 273–297.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dobson, P. D.; and Doig, A. J. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4): 771–783.

Du, Y.; and Mordatch, I. 2019. Implicit Generation and Modeling with Energy Based Models. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; and Bengio, S. 2018. Large Margin Deep Networks for Classification. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep Sparse Rectifier Neural Networks. In Gordon, G.; Dunson, D.; and Dudík, M., eds., *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, 315–323. Fort Lauderdale, FL, USA: PMLR.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572.

Grathwohl, W.; Wang, K.-C.; Jacobsen, J.-H.; Duvenaud, D.; Norouzi, M.; and Swersky, K. 2020. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; Piot, B.; kavukcuoglu, k.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 21271–21284. Curran Associates, Inc.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1321–1330. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jiang, Y.; Krishnan, D.; Mobahi, H.; and Bengio, S. 2019. Predicting the Generalization Gap in Deep Networks with Margin Distributions. In *International Conference on Learning Representations*.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2020. Decoupling Representation and Classifier for Long-Tailed Recognition. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; and Huang, F. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

LeCun, Y.; and Cortes, C. 2010. MNIST handwritten digit database.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*, 3734–3743. PMLR.

Li, Y.; Gao, F.; Ou, Z.; and Sun, J. 2018. Angular softmax loss for end-to-end speaker verification. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 190–194. IEEE.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision*, 2999–3007.

Liu, W.; Wang, X.; Owens, J.; and Li, Y. 2020. Energy-based Out-of-distribution Detection. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 21464–21475. Curran Associates, Inc.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, 7.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Matyasko, A.; and Chau, L.-P. 2017. Margin maximization for robust classification using deep learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 300–307.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2021. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*.

Neyshabur, B.; Bhojanapalli, S.; and Srebro, N. 2018. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *International Conference on Learning Representations*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.*, 115(3): 211–252.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2019. Pitfalls of Graph Neural Network Evaluation. arXiv:1811.05868.

Solla, S. A.; Levin, E.; and Fleisher, M. 1988. Accelerated Learning in Layered Neural Networks. *Complex Syst.*, 2(6): 625–639.

Suykens, J. A.; and Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3): 293–300.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114. PMLR.

Tenney, I.; Das, D.; and Pavlick, E. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593–4601. Florence, Italy: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30, 5998–6008. Curran Associates, Inc.

Wale, N.; Watson, I. A.; and Karypis, G. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3): 347–375.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, 25(7): 926–930.

Wightman, R. 2019. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models.

Wu, K.; and Yu, Y. 2019. Understanding Adversarial Robustness: The Trade-off between Minimum and Average Margin. arXiv:1907.11780.

Xie, J.; Lu, Y.; Zhu, S.-C.; and Wu, Y. 2016. A Theory of Generative ConvNet. In Balcan, M. F.; and Weinberger, K. Q., eds., *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 2635–2644. New York, New York, USA: PMLR.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, T.; and Zhou, Z.-H. 2017. Multi-Class Optimal Margin Distribution Machine. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 4063–4071. PMLR.

Zhou, B.; Cui, Q.; Wei, X.-S.; and Chen, Z.-M. 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9719–9728.