Ph.D. DISSERTATION

Countermeasures for Dataset Challenges in Deep Learning: Enhancing Robustness for Data Imbalance, Noisy Labels, and Stress-test

현실 데이터의 문제를 해결하기 위한 강건한 딥러닝 전략: 데이터 불균형, 레이블 노이즈, 스트레스 테스트

BY

SEULKI PARK

August 2023

DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Countermeasures for Dataset Challenges in Deep Learning: Enhancing Robustness for Data Imbalance, Noisy Labels, and Stress-test

현실 데이터의 문제를 해결하기 위한 강건한 딥러닝 전략: 데이터 불균형, 레이블 노이즈, 스트레스 테스트

지도교수 최 진 영

이 논문을 공학박사 학위논문으로 제출함

2023년 8월

서울대학교 대학원

전기·정보공학부

박 슬 기

박슬기의 공학박사 학위 논문을 인준함

2023년 8월

위 원 장:	고 형 석
부위원장:	최 진 영
위 원:	조 남 익
위 원:	곽 노 준
위 원:	윤 상 두

Abstract

Deep learning has shown remarkable success in solving a wide range of AI problems. However, when deployed in real-world scenarios, AI models are often challenged by issues such as noisy labels, imbalanced data, and robustness test. These challenges can have a significant impact on the performance and robustness of machine learning models.

This thesis proposes strategies for addressing these challenges and improving the robustness of deep learning models. Specifically, the thesis presents novel methods for handling noisy labels and imbalanced data. The proposed methods are evaluated on the most popular benchmark datasets, and the results show that they can significantly improve the performance and robustness of deep learning models.

Furthermore, the thesis introduces a new benchmark dataset, RoCOCO, to stress-test the robustness of multi-modal models. The dataset is designed to simulate real-world perturbations, providing a more realistic and challenging testbed for evaluating the robustness of AI models.

Overall, the research presented in this thesis contributes to the development of robust deep learning techniques that can better handle the challenges that arise when deploying machine learning models in real-world scenarios.

keywords: Imbalanced data, Long-tail distribution, Image classification, Oversampling, Augmentation, Noisy label, Robust AI, Multi-modal, Image-text Matching, Stress-test benchmark

student number: 2019-35916

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

In recent years, deep learning has made tremendous progress in various fields, including computer vision, natural language processing, and robotics. However, despite these advances, deep learning models are still prone to errors and failures when faced with challenging conditions, such as imbalanced data, noisy labels, and adversarial test datasets. These challenges are especially relevant in real-world scenarios, where deep learning models must operate in complex and dynamic environments that can introduce a wide range of data variations and uncertainties.

To address these challenges, researchers have been developing new techniques and approaches to improve the robustness and reliability of deep learning models. In this thesis, we investigate four different approaches that address different types of challenges that can arise in deep learning, with a particular focus on their applicability in real-world scenarios.

The first approach proposes a new loss function to address the problem of imbalanced data in visual classification tasks, which is a common issue in many real-world applications. The proposed method can significantly improve the performance of deep learning models under imbalanced data distributions, which can help address biases and inequities in real-world systems.

The second approach proposes a novel oversampling method to generate synthetic

minority samples by leveraging context-rich majority samples. This approach can be especially useful in real-world scenarios where collecting new data may be difficult or expensive, as it allows deep learning models to learn from existing data more effectively.

The third approach focuses on improving model robustness against noisy labels by introducing a post-training method called Influential Rank. This method can help mitigate the effects of noisy data in real-world scenarios, where data labeling errors are common and can have significant impacts on the performance of deep learning models.

Finally, the fourth approach proposes a new benchmark dataset to evaluate the robustness of image-text matching models against various types of challenges, including visual and textual noise, diverse visual and linguistic styles, and semantic shifts. This benchmark dataset can provide a more realistic and comprehensive evaluation of deep learning models' performance in real-world scenarios, where data variations and uncertainties are abundant.

Together, these four approaches contribute to the broader effort to build more reliable and robust deep learning systems that can operate effectively in the real world. The results of our investigations demonstrate that these approaches can significantly improve the performance of deep learning models under challenging conditions and offer valuable insights into the development of more robust deep learning algorithms. Consequently, by addressing the challenges that arise from datasets in real-world scenarios, we can develop more robust, effective, and trustworthy deep learning systems that can benefit a wide range of applications and domains.

The remainder of this thesis is organized as follows: Chapter 2 provides background and related work on deep learning with imbalanced data, noisy labels, and robustness benchmark. Then, Chapter 3 discusses the proposed new loss function for imbalanced data, Chapter 4 proposes oversampling method for imbalanced data, and Chapter 5 presents the Influential Rank method for noisy labels. Lastly, Chapter 6 in-

troduces the new benchmark dataset for image-text matching, and Chapter 7 concludes the thesis with a summary of the contributions, limitations, and future directions.

Chapter 2

RELATED WORK

The following section reviews previous work related to the research presented in this thesis. In particular, this section focuses on prior research related to the challenges that can arise from using datasets in real-world scenarios. These challenges can include class imbalance, noisy labels, and robustness test, and require novel methods to ensure the robustness and generalization of machine learning models.

2.1 Challenges from Imbalanced Data.

Another challenge in developing robust machine learning models is dealing with imbalanced datasets. Many real-world data exhibit skewed distributions [115, 72, 45, 120, 42], in which the number of samples per class differs greatly. This imbalance between classes can be problematic, since the model trained on such imbalanced data tends to overfit the dominant (majority) classes [78, 63, 12]. That is, while the overall performance appears to be satisfactory, the model performs poorly on minority classes. To overcome the class imbalance problem, extensive research has recently been conducted to improve the generalization performance.

The research on imbalanced learning can be divided into three approaches: reweighting approach, data-level approach, and meta-learning approach.

2.1.1 Re-weighting approach.

Cost-sensitive re-weighting methods assign different weights to samples to adjust their importance. Commonly used methods include re-weighting samples inversely proportional to the number of the class [74, 166] or the square root of class frequency [124]. Instead of heuristically using the number of classes, Cui et al. [35] proposed using the effective number of samples. While these methods can successfully assign more weights to the minority samples, they assign the same weights to all samples belonging to the same class, regardless of each importance.

To assign different weights to each sample according to its importance on the model, numerous methods were proposed for re-weighting samples based on their difficulties or losses [113, 40, 125]. That is, these methods down-weight well-classified samples and assign more weights to hard examples. These re-weighting methods might cause DNNs to be overfitted to the hard examples, since the high capacity of DNNs is sufficient to memorize the training data in the end [7]. In class imbalanced data, the hard examples are likely generated from the majority classes. As such, the minority samples are assigned smaller weights. Therefore, we need a more elaborate mean of re-weighting samples that can alleviate the overfitting to the majority samples. Meanwhile, Cao et al. [14] proposed label-distribution-aware margin loss to solve the overfitting to the minority classes by regularizing the margins.

2.1.2 Data-level approach.

Resampling methods aim to modify the training distributions to decrease the level of imbalance [82]. Resampling methods include undersampling and oversampling. Undersampling methods [184, 160] that discard the majority samples can lose valuable information, and undersampling is infeasible when the imbalance between classes is too high. The simplest form of oversampling is random oversampling (ROS) [160, 12], which oversamples all minority classes until class balance is achieved. This method is simple and can be easily used in any algorithm, but since the same sample is re-

peatedly drawn, it can lead to overfitting [142]. As a more advanced method, the synthetic minority over-sampling technique (SMOTE) [18], which oversamples minority samples by interpolating between existing minority samples and their nearest minority neighbors, was proposed. Following the success of SMOTE, several variants have been developed: Borderline-SMOTE [61], which oversamples the minority samples near class borders, and Safe-level-SMOTE [13], which defines safe regions not to oversample samples from different classes. These methods have been widely used by classical machine learning algorithms, but there are difficulties in using them for large-scale image datasets due to the high computational complexity of calculating the K-Nearest Neighbor for every sample. Generative adversarial minority oversampling (GAMO) [129] solves this issue by producing new minority samples by training a convex generator, inspired by the success of generative adversarial networks (GANs) [52] in image generation. However, training the generator incurs high additional training cost; moreover, GAMO can suffer from the infamous mode collapse of GANs [8]. To generate diverse minority data, recent works [87, 90] have proposed adversarial augmentations by adding small noise to the input images. To this end, Major-to-minor Translation (M2m) [87] transfers knowledge from majority classes using a pre-trained network, and Balancing Long-Tailed datasets (BLT) [90] uses a gradient-ascent image generator based on the confusion matrix.

Another recent line of research is oversampling in the feature space rather than in the input space: Deep Over-sampling (DOS) [6], Feature-space Augmentation (FSA)[27], and Meta Semantic Augmentation (MetaSAug) [109]. These methods aim to augment minority classes in the feature space by sampling from the in-class neighbors in the linear subspace [6], using learned features from pretrained networks [27], or using an implicit semantic data augmentation (ISDA) algorithm [167]. However, DOS [6] requires finding the nearest neighbors in the feature space, FSA [27] requires a pre-trained feature sub-network and a classifier for feature augmentation procedure. Lastly, MetaSAug [109] demands additional uniform validation samples that outnumber the

number of samples in the tail classes and hundreds and thousands of iterations for training. Consequently, these methods are less cost-efficient and technically more difficult to perform. On the other hand, our method oversamples diverse minority samples using a simple data augmentation technique and outperforms all previous methods while maintaining reasonable training costs.

2.1.3 Meta-learning approach.

Recently, the meta-learning-based approach [145, 118, 138] has emerged to enhance the performance of both approaches. Shu et al. [145] proposed a meta-learning process to learn a weighting function, while Liu et al. [118] proposed a re-sampling method by combining the advantage of ensemble learning and meta-learning. Furthermore, Ren et al. [138] proposed the meta-sampler and a balanced softmax that accommodates the shift of the distributions between the training data and test data. Although these methods can achieve satisfactory performance, these methods are somewhat difficult to implement in practice. For example, meta-weight-net [145] requires additional unbiased data for learning, and the meta-sampler in [138] is computationally expensive in practice. On the other hand, our proposed loss is simple to implement because it does not require a hyperparameter, a specially designed architecture, or additional learning for data re-sampling. Therefore, it is easy to use in collaboration with other methods.

2.1.4 Other long-tailed methods.

Recently, significant improvement has been achieved by two-stage algorithms: Deferred re-weighting (DRW) [14], classifier re-training (cRT), learnable weight scaling (LWS) [84], and the Mixup shifted label-aware smoothing model (MiSLAS) [192]. Two-stage algorithms decouple the learning process into representation learning and classifier learning. Meanwhile, a bilateral branch network (BBN) [193] uses an additional network branch for re-balancing, and RIDE [165] uses multiple branches called experts, each of which learns to specialize in different classes. PaCo [34] proposes

supervised contrastive learning with parametric class-wise centers for long-tailed classification.

2.1.5 Data Augmentation and Mixup Methods.

It is known that one of the methods to reduce overfitting of a model is to use a lot of data, and thus various data augmentation techniques have been proposed. Basic image manipulations include horizontal and vertical flipping, cropping, rotation, change brightness, noise injection [128], and color space transformations [171]. Spatial-level augmentation methods have performed satisfactorily in the computer vision fields. Cutout [38] removes random regions whereas CutMix [178] fills the removed regions with patches from another training image. In addition, mixup methods [182, 161, 153] linearly interpolate two images in a training dataset. Since the data augmentation method is closely related to the oversampling methods, some recent long-tailed recognition methods have used the mixup method. Zhou et al. [193] use the mixup as a baseline method, and MiSLAS [192] uses mixup in its Stage-1 training. However, these methods apply mixup without any adjustments, and little work has been done to explore appropriate data augmentation techniques for a long-tailed dataset. Recently, for an imbalanced dataset, the Remix [26] assigned a label in favor of the minority classes when mixing two samples. Unlike these methods, our method samples images from different distributions, which takes into account the specificity of long-tailed data distribution.

2.2 Challenges from Noisy Labels.

A significant challenge in developing robust machine learning models is dealing with noisy labels. The current deep learning has made a huge breakthrough because of 'data'. Thus, many researchers in both academia and industry endeavor to obtain considerable data. However, real-world data inevitably contain some proportion of incor-

rectly labeled data, owing to perceptual ambiguity, or errors from human or machine annotations. These noisy labels negatively affect the generalization performance of a trained model since a deep neural network (DNN) can easily overfit to even noisy labels due to its high capacity [180]. Therefore, learning from noisy labels (LNL) has received much attention in recent years [67, 174, 147, 191, 112, 24, 76, 149, 47] due to the increasing need to handle noisy labels in practice.

Learning with noisy labels has two main research directions. One is to find and use only clean labels for training, and the other is to directly train a robust model on noisy labels.

2.2.1 Noise-cleaning Approach.

Most noise-cleaning approaches focus on finding small-loss examples before overfitting because DNNs learn easy samples first and gradually learn difficult samples [7]. To prevent overfitting of a neural network, some methods simultaneously train two neural networks and select small-loss examples [60, 177, 194, 148, 154], while others train a network guided by a teacher network [81, 190]. Meanwhile, O2U-Net [75] adjusts the learning rate to take the model from overfitting to underfitting cyclically and records the losses of each example during the iterations. DivideMix [102] and SELF [131] incorporate semi-supervised learning with the small-loss trick for better sample selection. Recently, UNICON [85] proposed uniform clean sample selection algorithm to tackle the class imbalance problem induced by prior sample selection methods.

2.2.2 Noise-tolerant Approach.

The noise-tolerant approach aims to train a robust model on a noisy-label dataset without removing the noise. Some methods design noise-robust losses [117, 123, 189, 174, 76], and others attempt to reweight losses [130, 136]. Despite their theoretical justification, these approaches require mathematical assumptions or prior knowledge, such

as known noise rates and class-conditional noise transition matrices, which make them challenging in practice. To tackle the difficulty in estimating transition matrix, Cheng et al. [24] recently proposed manifold-regularized transition matrix estimation method. Meanwhile, there are more recent efforts to add a noise adaptation layer or relabel data [51, 147, 21], but they do not perform well especially when numerous classes exist or noise rates are heavy.

2.3 Challenges from Robustness Test.

In addition to class imbalance and noisy labels, datasets can also be susceptible to perturbation. With the increasing adoption of deep learning models in various applications, ensuring their robustness has become a crucial issue. To evaluate the robustness of the models, various attempts have been made in computer vision [66, 69, 65], and natural language processing (NLP) [77, 3, 41] areas, respectively.

2.3.1 Unimodal Robustness Test.

After the initial finding [157] that deep learning (DL) models are vulnerable to imperceptible perturbations, robustness in deep learning methods has actively studied in both computer vision and natural language processing (NLP) areas. In computer vision, one research direction is data poisoning [10, 152, 68, 56, 22], which attacks the robustness of models during training by adding images with small perturbations. Meanwhile, adversarial attack studies [53, 94, 17, 31, 57] inject imperceptible noises to test images so that a model can make wrong predictions. For image retrieval task, Li et al. [99] showed that adding invisible noise to query image can make the model return incorrect images. Another line of research has proposed new ImageNet benchmarks for common robustness evaluation. For example, ImageNet-C [66] is applied with 75 common visual corruptions, and ImageNet-P [66] is implemented with common perturbations. Also, ImageNet-A [69] provides images belonging to ImageNet classes but

more difficult, and ImageNet-R [65] introduces examples with various renditions. In NLP, research on data poisoning [162] and adversarial attacks [44, 3, 80, 49, 98, 41, 11] has also been actively studied to fool the prediction of models. Adversarial examples are produced by character-level modifications [9], paraphrasing sentences [77], or substituting a word with a synonym [140, 107].

2.3.2 Multimodal Robustness Test.

As vision-language models have generated growing research interest, robustness work for cross-modal domain has been actively studied [133, 105, 16]. Especially, in visual question answering (VQA) task, diverse robustness-evaluation benchmark [186, 55, 143, 50, 144, 106] has been proposed. For example, VQA-Rephrasings [143] generated dataset by rephrasing questions to evaluate the robustness in the input question. Adversarial VQA [106] and AdVQA [144] collected adversarial examples in human-in-the-loop manner. However, to the best of our knowledge, this is the first work to propose robustness-evaluation benchmark in ITM task. We hope that our work can inspire the future research to create more diverse stress-test benchmarks in ITM area.

2.3.3 Image-Text Matching Methods.

Most image-text matching (ITM) methods [48, 46, 150, 73, 29, 168, 185] aim to learn joint visual-semantic embedding (VSE) such that paired image and text representation in the embedding space are close. Many VSE methods [97, 163, 39, 19] use region features extracted from Faster R-CNN [139] with bottom-up attention [5]. VSE∞ [19] also use grid features extracted from Faster R-CNN pre-trained on Visual Genome [91] and ImageNet [36] in [5], and Instagram pretrained ResNext-101 [173].

In recent years, large-scale pre-training models [23, 111, 187, 79, 88, 101, 100, 122, 25, 176, 1] have shown strong achievement in both zero-shot and fine-tuned performances. Most of these models adopt transformer architecture and can learn cross-modal representations benefiting from large-scale image-text pairs. For a more thor-

ough study, we refer the reader to a recent survey [15]. In this thesis, we re-evaluate the robustness of state-of-the-art ITM models.

2.3.4 Image-Text Matching Datasets.

Recently, new ITM benchmark datasets [132, 28] have been proposed by extending MS COCO. Crisscrossed Captions (CxC) [132] add semantic similarity between all pairs to improve limited associations in MS COCO. Thus, CxC has enabled scoring between intra- and intermodality pairs. Meanwhile, ECCV caption [28] provides abundant positive image-caption pairs to correct the false negatives in MS COCO. While the previous works provided improved benchmark datasets, our main difference is that we aim to test the vulnerability of the models.

2.4 Influence function.

The influence function was proposed to find the influential instance of a sample to a model, which has been studied for decades in robust statistics [59, 30]. Recently, attempts have been made to use influence function in deep neural networks [2, 89]. Koh & Liang [89] used influence functions to understand the effect of a training example on a test example.

Consider a classification problem with n training data $(x_1, y_1), \cdots, (x_n, y_n)$, where x_i is the i-th training point (e.g., an image) and y_i is its label. Let $f(x, \theta)$ denote a model parameterized by θ and $L(y, f(x, \theta))$ be the loss for a training point (x, y). Given the empirical risk $R(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i, \theta))$, the optimal parameter that minimizes the risk is $\hat{\theta} \stackrel{\text{def}}{=\!=} \text{argmin}_\theta R(\theta)$. The influence of a training point (x, y) can be efficiently approximated by the parameter change if the distribution of the training data at the point (x, y) is slightly modified. A new parameter when removing the training point (x, y) is derived as $\hat{\theta}_{x,\omega} \stackrel{\text{def}}{=\!=} \text{argmin}_\theta R(\theta) + \omega L(y, f(x, \theta))$, where we assign $\omega = -\frac{1}{n}$.

Then, the influence of (x, y) on the parameters of the trained model has been presented in [89], which is denoted by

$$\mathcal{I}_M(x; \hat{\theta}) = -H_{\hat{\theta}}^{-1} \nabla_\theta L(y, f(x, \hat{\theta})),$$

where $H_{\hat{\theta}} \overset{\text{def}}{=\!=} \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(y_i, f(x_i, \theta))$ is the Hessian and is positive definite by assumption. Meanwhile, [62] proposed stochastic gradient descent (SGD) influence that can infer the influential examples for models trained with SGD. However, this method is limited to optimization by SGD and requires to store the parameters of the model at every step, requiring huge memory consumption for DNNs. In this thesis, we utilize influence function to detect noisy labels and design a novel loss for imbalanced classification.

Chapter 3

Influence-Balanced Loss for Imbalanced Data

3.1 Overview

Despite the remarkable success of deep neural networks (DNNs) these days, many areas of computer vision suffer from highly imbalanced datasets. Many real-world data exhibit skewed distributions [115, 72, 45, 120, 42], in which the number of samples per class differs greatly. This imbalance between classes can be problematic, since the model trained on such imbalanced data tends to overfit the dominant (majority) classes [78, 63, 12]. That is, while the overall performance appears to be satisfactory, the model performs poorly on minority classes. To overcome the class imbalance problem, extensive research has recently been conducted to improve the generalization performance by reducing the overwhelming influence of the dominant class on the model.

The cost-sensitive re-weighting approach aims to assign class penalties to shift the decision boundary in a way that reduces the bias induced by the data imbalance. For this purpose, the most commonly adopted method is to re-weight samples inversely to the number of training samples in each class to assign more weights for the minority classes [74, 166, 35]. These methods have focused on only global-level class distribution and assign the same fixed weight to all samples belonging to the same class.

However, not all samples in a dataset play an equal role in determining the model parameters [30]. That is, some samples have greater influences on forming a decision boundary. Hence, each sample needs to be re-weighted differently according to its impact on the model.

Recently, numerous studies have been conducted in which each sample is considered to design sample-wise loss functions [40, 113, 125]. Specifically, these methods down-weight well-classified samples and assign more weights to *hard examples*, which yield high errors. This re-weighting might lead to the complete training when the high capacity of DNNs is sufficient to finally memorize the whole training data [181, 7]. This implies that DNN is overfitted to hard samples, which are located at the overlapping region between the majority and minority classes. In the imbalanced data, most hard samples are majority samples that enforce the decision boundary to be complex and shift to the minority region.

To address the aforementioned problem, in this thesis, we propose a loss-sensitive method to down-weight samples that cause overfitting of a DNN trained with highly imbalanced data. To this end, we derive a formula that measures how much each sample influences the complex and biased decision boundary. To derive the formula, we utilize the influence function [30], which has been widely used in robust statistics. Using the derived formula, we design a novel loss function, called influence-balanced (IB) loss, that adaptively assigns different weights to samples according to their influence on a decision boundary. Specifically, we re-weight the loss proportionally to the inverse of the influence of each sample. Our method is divided into two phases: standard training and fine-tuning for influence balancing. During the fine-tuning phase, the proposed IB loss alleviates the influence of the samples that cause overfitting of the decision boundary.

Through extensive experiments on multiple benchmark data sets, we demonstrate the validity of our method, and show that the proposed method outperforms the state-of-the-art cost-sensitive re-weighting methods. Furthermore, since our IB loss is not

restricted to a specific task, model, or training method, it can be easily utilized in combination with other recent data-level algorithms and hybrid methods for class-imbalance problems.

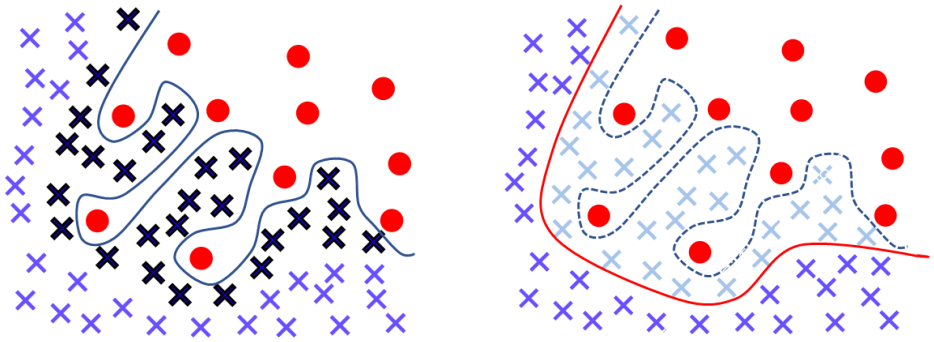
The main contributions are as follows: (1) We discover that the existing loss-based loss methods can lead a decision boundary of DNNs to eventually overfit to the majority classes. (2) We design a novel influence-balanced loss function to re-weight samples more effectively in such a way that the overfitting of the decision boundary can be alleviated. (3) We demonstrate that simply substituting our proposed loss for the standard cross-entropy loss significantly improves the generalization performance on highly imbalanced data.

3.2 Influence-balanced Loss

To address the imbalanced data learning problem, our idea is to re-weight samples by their influences on a decision boundary to create a more generalized decision boundary. First, we present the key idea of our proposed method in Section 3.2.1. For the background, we briefly review the influence function in Section 3.2.2 and then derive the IB loss in Sections 3.2.3, 3.2.4, and 3.2.5. Finally, the training scheme is presented in Section 3.2.6.

3.2.1 Key Idea of Proposed Method

In this section, we explain how the re-weighting of samples according to their influence can help to form a well-generalized decision boundary on class imbalance data. It is well known that the high capacity of DNNs is sufficient to finally memorize the entire training data [181, 7]. This implies that DNN can be overfitted to samples that are located at the overlapping region between the majority and minority classes, as illustrated in Figure 3.1 (a). In the imbalanced data, many majority samples invade among sparse minority samples and become dominant in the overlapping area, thereby



(a) Original decision boundary.

(b) Proposed method.

Figure 3.1: **Illustration of the key concept of our approach.** The red and blue marks belong to the minority and majority classes, respectively, in binary classification. (a) The black border line represents an initial decision boundary formed on an imbalanced dataset. The black $\times$ samples have greater influence on the decision boundary than do the blue $\times$ samples, since the decision boundary would substantially change without the black $\times$ samples. (b) Our proposed method aims to down-weight the samples (light blue $\times$ samples) that have a large influence on the overfitted decision boundary (dotted line) to create a smoother decision boundary (the red line).

enforcing the decision boundary to be complex and shift to the minority region.

Furthermore, the black $\times$ samples in Figure 3.1 (a) have a stronger influence on forming the decision boundary, as they support the decision boundary, which substantially changes when the samples are removed. Thus, it can be said that the dominant samples with high influence are likely to create a complex and biased decision boundary. As illustrated in Figure 3.1 (b), by down-weighting the highly influential samples, the decision boundary can be smoothed via fine-tuning. To this end, we derive an influence-balanced (IB) loss by employing the influence function [30], which measures the training sample's influence on the model.

3.2.2 Influence Function

The influence function [30] allows us to estimate the change in the model parameters when a sample is removed, without actually removing the data and retraining the model. Let $f(x, w)$ denote a model parameterized by w with n training data $(x_1, y_1), \cdots, (x_n, y_n)$, where x_i is the i-th training sample, and y_i is its label. Given the empirical risk $R(w) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i, w))$, the optimal parameter after initial training is defined by $w^* \stackrel{\text{def}}{=\!=} \text{argmin}_w R(w)$.

During the fine-tuning phase, to address the imbalance issue, we re-weight loss proportionally to the inverse of the influence of a sample. The influence of a point (x, y) can be approximated by the parameter change if the distribution of the training data at that point is slightly modified. A new parameter when removing the training point (x, y) is derived as $w_{x,\varepsilon} \stackrel{\text{def}}{=\!=} \text{argmin}_w R(w) + \varepsilon L(y, f(x, w))$. Then, under the assumption that $\nabla_w R(w) \approx 0$ for w in the vicinity of w^*, we can utilize the influence function in [2, 89] to re-weight the sample-wise loss during the fine-tuning phase. The influence function is given by

$$\mathcal{I}(x; w) = -H^{-1} \nabla_w L(y, f(x, w)), \tag{3.1}$$

where $H \stackrel{\text{def}}{=\!=} \frac{1}{n}\sum_{i=1}^{n} \nabla_w^2 L(y_i, f(x_i, w))$ is the Hessian and is positive definite based by assumption that L is strictly convex in a local convex basin around the optimal point w^*.

3.2.3 Influence-balanced weighting factor

From $\mathcal{I}(x; w)$, we derive the IB loss. Since $\mathcal{I}(x; w)$ is a vector that requires heavy computation of the inverse Hessian, it is nearly impossible to directly use this. Therefore, we solve this problem by modifying $\mathcal{I}(x; w)$ to a simple but effective influence-balanced weighting factor. First, since we need the relative influence of the training samples, not the absolute values, we can simply ignore the inverse Hessian in $\mathcal{I}(x; w)$.

This is because the inverse of hessian is commonly multiplied by all the training samples. Then, we design the IB weighting factor as follows:

$$\mathcal{IB}(x; w) = ||\nabla_w L(y, f(x, w))||_1 \tag{3.2}$$

Equation 3.2 turns out to be the magnitude of the gradient vector. Anand et al. [4] revealed that the net error gradient vector is dominated by the major classes in the class imbalance problem. Hence, re-weighting samples by the magnitude of the gradient vector can successfully down-weight samples from dominant classes. In the Experiments section, we justify the choice of the L1 norm. In the following section, we demonstrate how the IB weighting factor can be used with the actual loss.

3.2.4 Influence-Balanced Loss

When using the softmax cross-entropy loss, Equation (3.2) can be further simplified. The cross-entropy loss is denoted by $L(y, f(x, w)) = -\sum_k^K y_k \log f_k$, where y_k is a ground truth, and f_k is the k-th output of the model $f(x, w)$, with K total classes. Since we are interested in the overfitting on the decision boundary of the model, we focus on the change in the last fully connected (FC) layer of a deep neural network. Let $h = [h_1, \cdots, h_L]^T$ be a hidden feature vector, an input to the FC layer, and $f(x, w) = [f_1, \cdots, f_K]^T$ be the output denoted by $f_k := \sigma(w_k^T h)$, where σ is the softmax function. The weight matrix of the FC layer is denoted by $w = [w_1, \cdots, w_K]^T \in R^{K \times f}$.

Then, the gradient of the loss w.r.t. w_{kl} is computed as

$$\frac{\partial}{\partial w_{kl}} L(y, f(x, w)) = (f_k - y_k) h_l. \tag{3.3}$$

The same results are obtained for the cross-entropy loss with a sigmoid function or a mean squared error (MSE) loss for regression. Then, IB weighting factor in (3.2) is given by

$$\begin{aligned}\mathcal{IB}(x;w) &= \sum_{k}^{K}\sum_{l}^{L}|(f_k - y_k)h_l| \\ &= \sum_{k}^{K}|(f_k - y_k)|\sum_{l}^{L}|h_l| \\ &= ||f(x,w) - y||_1 \cdot ||h||_1,\end{aligned} \tag{3.4}$$

of which inverse can be used for the re-weighting factor to down-weight an influential sample in fine-tuning to adjust the decision boundary that enhance the imbalanced data learning. Finally, the influence-balanced loss is given by

$$L_{IB}(y, f(x,w) = \frac{L(y, f(x,w))}{||f(x,w) - y||_1 \cdot ||h||_1}. \tag{3.5}$$

The proposed influence-balanced term constrains the decision boundary to not overfit to influential majority samples (see Figure 3.1(b)).

3.2.5 Influence-Balanced Class-wise Re-weighting

Moreover, we add a class-wise re-weighting term λ_k to the IB-loss in (3.5) as

$$L_{IB}(w) = \frac{1}{m}\sum_{(x,y)\in D_m}\lambda_k\frac{L(y, f(x,w))}{||f(x,w) - y||_1 \cdot ||h||_1}, \tag{3.6}$$

where $\lambda_k = \alpha n_k^{-1} / \sum_{k'=1}^{K} n_{k'}^{-1}$. Here, n_k is the number of samples in the k-th class in the training dataset, and normalization is performed to make λ_k have a similar scale for every class. α is introduced as a hyper-parameter for an adjustment.

The class-wise re-weighting yields the following two effects. First, λ_k mitigates the bias of the decision boundary arising from the overall imbalanced distribution through the slow-down of the majority loss minimization. Second, λ_k further controls the sample-wise re-weighting depending on the class to which a highly influential sample belongs. That is, if the sample belongs to a majority class, λ_k further down-weights the sample because the decision boundary is likely to be overfitted by the majority sample. Meanwhile, if the sample belongs to a minority class, λ_k becomes smaller than

Algorithm 1 Influence-Balanced Training

INPUT: training dataset $D = (X, Y)$.

OUTPUT: influence-balanced model $f(x, w)$.

Phase 1: Normal training

for $t = 1, \ldots, T_1$

sample mini-batch D_m from D.

$L(w) \leftarrow \frac{1}{m} \sum_{(x,y) \in D_m} L(y, f(x, w))$.

update $w^t = w^{t-1} - \eta \nabla L(w)$.

end for

Phase 2: Fine-tuning for influence balancing

for $t = T_1 + 1, \ldots, T$

sample mini-batch D_m from D

$L_{IB}(w) \leftarrow \frac{1}{m} \sum_{(x,y) \in D_m} \lambda_k \frac{L(y, f(x,w))}{||f(x,w) - y||_1 \cdot ||h||_1}$

update $w^t = w^{t-1} - \eta \nabla L(w)$

end for

that of a majority sample and does not down-weight the loss much, because the large influence of the minority sample is natural due to the data scarcity.

3.2.6 Influence-balanced Training Scheme

The influence-balanced training process comprises two phases: normal training and fine-tuning for balance. We refer to T_1 as the transition time from normal training to fine-tuning. During the normal training phase, the network is trained following any training scheme for the first T_1 epochs. Meanwhile, during the fine-tuning phase, the influence-balanced loss is applied to mitigate the overfitting of the decision boundary arising from the influential (noisy) majority samples. Since our IB loss during the fine-tuning phase alleviates the overfitting, it is advantageous to set T_1 as the epoch when the model has begun to converge to the local (global) minimum. Generally, it

is recommended to set T_1 as half of the total training scheme. We present the performance change according to the number of training epochs during normal training in the Experiments section. As evident, our training does not require an additional training scheme or a specifically designed architecture. Thus, it can be utilized easily in any tasks suffering from imbalanced data. The pseudo-code of the training procedure is presented in Algorithm 1.

3.3 Experiments

3.3.1 Experimental Settings

Datasets. We verified the effectiveness of our method on three commonly used benchmark datasets: CIFAR-10, CIFAR-100 [92], Tiny ImageNet [95], and iNaturalist 2018 [72]. The CIFAR-10 and CIFAR-100 datasets consist of 50,000 training images and 10,000 test images with 10 and 100 classes, respectively. Meanwhile, Tiny ImageNet contains 200 classes for training, in which each class has 500 images. Its test set contains 10,000 images. Since CIFAR and Tiny ImageNet are evenly distributed, we have made these datasets imbalanced according to [35, 12], respectively. Primarily, we investigate two common types of imbalance: (i) long-tailed imbalance [35] and (ii) step imbalance [12]. In long-tailed imbalance, the number of training samples for each class decreases exponentially from the largest majority class to the smallest minority class. To construct long-tailed imbalanced datasets, the number of selected samples in the k-th class was set to $n_k\mu^k(\mu \in (0, 1))$, where n_k is the original number of the k-th class. Meanwhile, in step imbalance, the classes are divided into two groups: the majority class group and minority class group. Every class within a group contains the same number of samples, and the class in the majority class group has many more samples than that in the minority class group. For evaluation, we used the original test set. The imbalance ratio ρ is defined by $\rho = \frac{\max_k\{n_k\}}{\min_k\{n_k\}}$. Thus, the imbalance ratio represents the degree of imbalance in the dataset. We evaluated the performance of our method under

various imbalance ratios from 10 to 200.

The iNaturalist 2018 dataset is a large-scale real-world dataset containing 437,513 training images and 24,426 test images with 8,142 classes. iNaturalist 2018 exhibits long-tailed imbalance, whose imbalance ratio is 500. We used the official training and test splits in our experiments.

Baselines. We compared our algorithm with the following cost-sensitive loss methods: (1) Our baseline model, which is trained on the standard cross-entropy loss. Comparing our model with this baseline enables us to clearly understand how much our training scheme has improved the performance; (2) focal loss [113], which increases the relative loss for hard samples and down-weights well-classified samples; (3) CB loss [35], which re-weights the loss inversely proportional to the effective number of samples; (4) LDAM loss [14], which regularizes the minority classes to have larger margins.

Since our IB loss can be easily combined with other methods, we employee two further variants. First, IB + CB uses the effective number in CB loss, instead of using λ_k in IB. Second, IB + focal uses focal loss during the fine-tuning phase, instead of using the cross-entropy loss. We demonstrate that combination with other methods can further improve the performance.

Implementation Details. We used PyTorch [135] to implement and train all the models in the thesis, and we used ResNet architecture [64] for all datasets. For CIFAR datasets, we used randomly initialized ResNet-32. The networks were trained for 200 epochs with stochastic gradient descent (SGD) (momentum = 0.9). Following the training strategy in [35, 14], the initial learning rate was set to 0.1 and then decayed by 0.01 at 160 epochs and again at 180 epochs. Furthermore, we used a linear warm-up of the learning rate [54] in the first five epochs. Since our method uses a two-phase training schedule, we trained for the first 100 epochs with the standard cross-entropy loss, then fine-tuned the networks using the IB loss for the next 100 epochs. We trained the models for CIFAR on a single NVIDIA GTX 1080Ti with a batch size of 128. For

Tiny ImageNet, we employed ResNet-18 and used the stochastic gradient descent with a momentum of 0.9, and weight decay of $2\mathrm{e}{-4}$ for training. The networks were initially trained for 50 epochs, and then fine-tuned for the subsequent 50 epochs with IB loss. The learning rate at the start was set to 0.1 and was dropped by a factor of 0.1 after 50 and 90 epochs. For iNaturalist 2018, we trained ResNet-50 with four GTX 1080Ti GPUs. The networks were initially trained for 50 epochs and then fine-tuned for the subsequent 150 epochs with IB loss. The learning rate at the start was set to 0.01 and was decreased by a factor of 0.1 after 30 and 180 epochs.

As a simple but important implementation trick, we added $\epsilon = 0.001$ to $\mathcal{IB}(x; w)$ to prevent numerical instability in inversion when the influence approaches zero. We discuss the influence of the hyperparameter (ϵ) in the following section.

3.3.2 Analysis

To validate the proposed method, we conducted extensive experiments.

Is influence meaningful for re-weighting?

First, to confirm whether influence can act as a meaningful clue of re-weighting for class imbalance learning, we compared the influences between a balanced dataset and an imbalanced dataset. For an imbalanced CIFAR-10, we used the long-tailed version of CIFAR-10 with the imbalance ratio $\rho = 100$, in which the largest class, 'plane' (i.e., class index 0), contains 5,000 samples, while the smallest class, 'truck' (i.e., class index 9), contains only 50 samples. We trained ResNet-32 with a standard cross-entropy loss for 200 epochs, as described in Implementation Details, on both the balanced (original) and imbalanced CIFAR-10. We plotted the influences of both classes in Figure 3.2. We scaled the influences to between 0 and 1 for each dataset. Since the minority class contains only 50 samples, we selected the highest 50 samples for comparison. As illustrated in Figure 3.2, there was little difference in the distributions of the influences between the classes in the balanced dataset. However, in the imbalanced

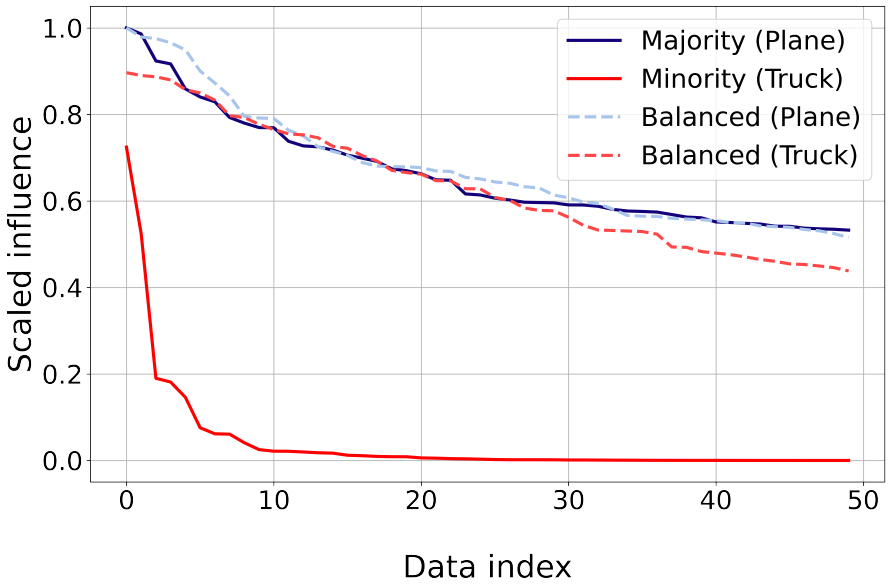


Figure 3.2: **Comparison of Influences between balanced and imbalanced dataset.** We plotted the influences of samples on ResNet-32 trained on the original CIFAR-10 and the imbalanced version of CIFAR-10. The solid and dashed lines represent the influences of the imbalanced data and balanced data, respectively. While there is little difference in the balanced dataset, it can be seen that the influence of the dominant class is much greater than that of the minor class in the imbalance dataset.

dataset, the minority samples had significantly less influence on the model than did the majority samples. This result corroborates that majority samples greatly contribute to forming a decision boundary, and re-weighting their influences can improve the generalization of the model.

Magnitude of Influence.

In Section 3.2.3, we used L_1 norm to compute the magnitude of the influences. We investigated performance variations depending on three vector norms to compute the

magnitude of the gradient vector $\nabla_w L(y, f(x, w))$: L_1, L_2, L_∞. As indicated in Table 3.1, L_1 norm, which provides a distinctive change of influence around the equilibrium point, exhibits the best classification accuracy on CIFAR-10 with multiple imbalance ratios.

Table 3.1: Comparison of norms. Using L_1 norm yields the best performance.

	CIFAR-10		**CIFAR-100**	
Imbalance (ρ)	100	20	100	20
L_1	**78.41**	**85.80**	**40.85**	**52.85**
L_2	75.67	84.35	36.41	50.95
L_∞	77.23	84.30	37.48	50.99

Timing for starting fine-tuning for balancing.

Our training scheme is divided into two phases: normal training and fine-tuning for balancing. This must determine the transition time between normal training and fine-tuning for balancing. Hence, we investigated the results on how much the transition time affects the performance and determined the best transition time. For this, we experimented on the long-tailed version of CIFAR-10 with imbalance ratios of $\rho = 10$ and 100. In Figure 3.3, the X-axis represents the number of training epochs T_1 for the normal training phase. We varied the transition time, T_1, from 0 to 120 while the total number of training epochs was fixed at 200. The solid line represents the classification accuracy earned by the models for each training schedule. To analyze the relationship between the convergence of the normal training phase and the transition timing, we plotted the standard cross-entropy loss without adopting the IB loss for the whole training epochs (dashed lines).

From Figure 3.3, it can be observed that the proposed method demonstrates ro-

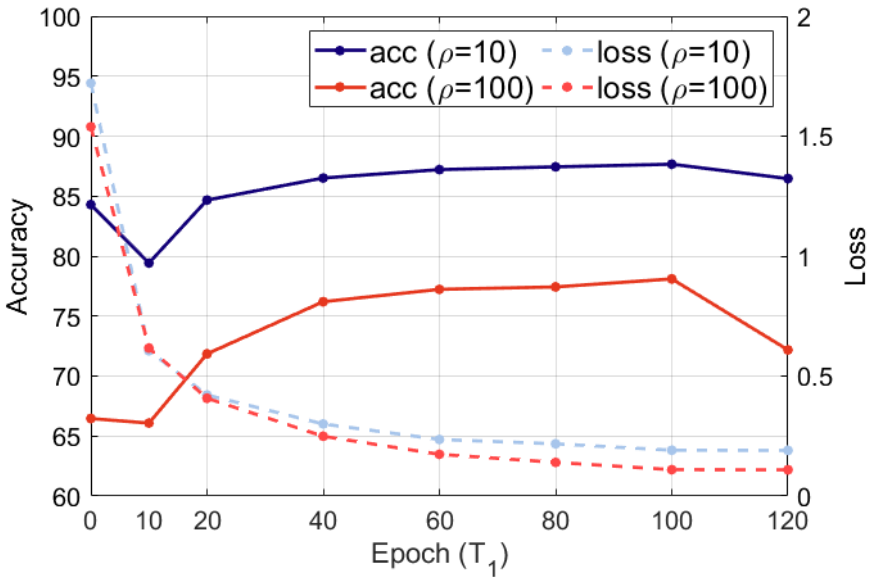


Figure 3.3: **Influence-balanced training scheme.** We varied the *training epochs for the normal training*, T_1, to determine the best transition time from the normal training to the influence-balance fine-tuning. We achieved the best performance when setting the transition time to the point when the training loss converges.

bust performance regardless of the choice of transition time T_1. Yet, the transition to fine-tuning after the 100th epoch yields the best performance when the training loss has converged. Since the influence function is derived from the loss minimization context [89], it is reasonable to begin the fine-tuning phase after the learning converges.

Effects of ϵ.

As mentioned in Implementation Details, for all datasets, we added the hyperparameter ($\epsilon = 0.001$) to $\mathcal{IB}(x; w)$ to prevent numerical instability. To analyze the effects of the hyperparameter, we conducted experiments with the following denominators for the IB loss (3.5): (a) $\mathcal{IB}(x; w) + 1\text{e}{-8}$, (b) $\mathcal{IB}(x; w) + 1\text{e}{-3}$, (c) $\mathcal{IB}(x; w) + 1\text{e}{-2}$,

and (d) $1e{-}3$. We iterated experiments three times with different random seeds on the long-tailed CIFAR-10 ($\rho = 100$). As presented in Table 3.2, setting ϵ to $1e{-}3$ yields the best performance. Thus, we set ϵ as $1e{-}3$ in all the experiments. However, when we did not use the IB weighting factor, the accuracy greatly decreased.

Table 3.2: Effects of ϵ.

Epsiilon	(a) IB+1e-8	(b) IB+1e-3	(c) IB+1e-2	(d) 1e-3
Accuracy	76.03 ± 0.97	78.17 ± 0.57	77.55 ± 0.55	64.91 ± 1.40

3.3.3 Comparison of Class-Wise Accuracy.

In this section, to validate that the performance improvement has actually resulted from the minority classes, not from the majority classes, we report the class-wise accuracy on both the long-tailed and the step-imbalanced CIFAR-10. We compare the proposed method with the state-of-the-art cost-sensitive loss methods. Since previous studies do not report the class-wise accuracy on the imbalanced CIFAR-10, we implemented the baseline methods [113, 35, 14]. For the implementation of LDAM [14], we used their official implementation code to reproduce the results.

The overall results are reported in Table 3.3. As presented in Table 3.3, existing methods exhibit severe performance degradation in the minority classes. That is, the reported improvements from the existing methods were attributed to the majority classes, not the minority classes. In contrast, the proposed IB loss exhibited a significant improvement in all the minority classes.

It is noteworthy that the performance improvement was not significant, especially on the step-imbalanced CIFAR-10 with the focal loss [113] method. We argue that this demonstrates that most hard examples are majority samples in highly imbalanced data and that those samples enforce the decision boundary to be overfitted. In contrast, our proposed influence-balanced re-weighing can mitigate the influences of the

Table 3.3: Class-wise classification accuracy (%) of ResNet-32 on imbalanced CIFAR-10 dataset. The number of test samples for each class is the same as 1000. The best results are marked in bold.

	Imbalanced CIFAR-10									
Class	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Long-Tailed ($\rho = 50$)										
#Training samples	5000	3237	2096	1357	878	568	368	238	154	100
Baseline (CE)	**97.4**	**98.0**	**84.0**	**80.3**	78.8	68.4	76.1	64.5	57.0	52.0
Focal [113]	91.6	95.1	73.1	59.2	67.8	67.2	84.2	77.3	**83.9**	61.8
CB [35]	92.9	96.3	79.2	75.1	82.4	69.9	75.0	69.1	73.6	66.8
LDAM [14]	96.9	98.5	82.9	74.7	82.8	69.0	78.5	69.9	65.3	66.0
LDAM-DRW [14]	94.8	97.8	82.6	72.3	85.3	73.0	82.0	76.7	75.8	72.4
IB	92.2	96.2	81.3	66.6	**85.7**	**76.4**	81.7	75.9	79.9	**81.1**
IB + CB	93.8	97.2	78.1	64.8	84.8	74.2	**86.4**	**79.7**	79.5	76.9
IB + Focal	90.9	96.1	81.7	69.0	82.0	75.7	85.2	77.5	80.2	76.8
Step-Imbalance ($\rho = 50$)										
#Training samples	5000	5000	5000	5000	5000	100	100	100	100	100
Baseline (CE)	95.9	**99.2**	**91.5**	**91.9**	95.5	24.8	40.2	46.7	52.7	55.1
Focal [113]	96.3	93.9	91.2	90.5	**95.7**	20.0	46.7	48.8	56.1	57.6
CB [35]	87.4	96.3	76.8	77.0	85.7	34.6	61.5	56.5	68.7	63.8
LDAM [14]	**96.4**	98.5	91.1	90.2	94.6	28.3	50.3	57.0	56.2	64.4
LDAM-DRW [14]	94.5	97.2	88.0	84.5	94.3	50.4	69.9	71.4	74.6	76.0
IB	94.0	97.7	86.7	83.2	93.8	56.9	71.0	**75.1**	76.5	81.7
IB + CB	91.8	95.7	86.6	79.4	93.6	62.8	77.2	72.3	74.2	**87.3**
IB + Focal	91.2	96.4	83.3	77.1	92.0	**64.8**	**78.0**	74.4	**83.5**	83.1

majority samples that cause overfitting. As a result, it can achieve robust and superior performance for the minority classes with a very small number of samples.

Although using the influence-balanced loss alone can achieve significant enhancement for the classification of the minority classes, it is beneficial to combine it with other methods. For example, the results indicate that applying the influence-balanced

Table 3.4: Classification accuracy (%) of ResNet-32 on imbalanced CIFAR-10 and CIFAR-100 datasets. "†" indicates that the results are copied from the original paper, and "‡" means that the results are from the experiments in CB [35]. The best results are marked in bold.

	Imbalanced CIFAR-10					**Imbalanced CIFAR-100**				
Imbalance (ρ)	200	100	50	20	10	200	100	50	20	10
Long-Tailed										
Baseline (CE)	66.28	70.87	78.22	82.43	86.49	33.54	38.05	43.71	51.21	56.96
‡Focal [113]	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
†CB [35]	68.89	74.57	79.27	84.36	87.49	36.23	39.60	45.32	52.59	57.99
†LDAM [14]	-	73.35	-	-	86.96	-	39.6	-	-	56.91
†LDAM-DRW [14]	-	77.03	-	-	88.16	-	42.04	-	-	57.99
IB	73.96	78.26	**81.70**	**85.8**	**88.25**	37.31	**42.14**	46.22	52.63	57.13
IB + CB	73.69	78.04	81.54	85.42	88.09	37.06	41.31	46.16	52.74	56.78
IB + Focal	**75.05**	**79.76**	81.51	85.31	88.04	**38.23**	42.06	**47.49**	**53.28**	**58.20**
Step-Imbalance										
Baseline (CE)	56.97	64.81	69.35	79.71	84.16	38.29	39.27	41.65	48.55	54.13
†LDAM [14]	-	66.58	-	-	85.00	-	39.58	-	-	56.27
†LDAM-DRW [14]	-	76.92	-	-	87.81	-	45.36	-	-	59.46
IB	72.15	76.53	81.66	85.41	87.72	39.66	**45.39**	**48.93**	53.57	57.96
IB + CB	69.96	75.97	82.09	85.27	**88.01**	39.69	45.27	48.80	53.42	57.86
IB + Focal	**74.12**	**77.97**	**82.38**	**85.68**	87.90	**40.39**	44.96	48.92	**54.53**	**59.54**

loss with the focal loss can encourage the network to learn 'good' hard samples, while down-weighting the influential ones that induce overfitting.

3.3.4 Comparison with State-of-the-Art

Experimental results on CIFAR.

The overall classification accuracy is provided in Table 3.4. The model performance is reported on the unbiased test set as the same as the other methods. The results indicate that adopting the proposed influence-balanced loss significantly improves the generalization performance and outperforms the recent cost-sensitive loss methods. On multiple benchmark datasets, using IB loss alone could achieve the best performance. This suggests that it is effective for the robustness of the model to balance the influence of samples responsible for overfitting the decision boundary. When combined with other methods [35, 113], we could further improve the accuracy on multiple datasets. This indicates that our proposed method of down-weighting influential samples that induce overfitting can benefit other methods as well.

Table 3.5: Class. accuracy (%) of ResNet-18 on Tiny ImageNet.

	Long-Tailed		***Step-Imbalance***	
Imbalance (ρ)	100	10	100	10
Baseline (CE)	38.52	36.62	36.74	51.11
Focal [113]	38.95	54.02	38.24	41.77
CB [35]	41.37	54.82	37.35	54.3
LDAM* [14]	37.47	52.78	39.37	52.57
IB	**42.65**	**57.22**	**41.13**	**54.83**

Experimental results on Tiny ImageNet.

We evaluated our method on Tiny ImageNet in Table 3.5. While we performed the experiments for the other baselines, the results of LDAM were copied from their original thesis. As presented in Table, IB loss outperforms other baselines on Tiny ImageNet as well.

Table 3.6: Class. accuracy (%) of ResNet-50 on iNaturalist 2018.

	iNaturalist 2018	
Method	top1	top5
Baseline (CE)	57.30	79.48
Focal [113]	58.03	78.65
CB [35]	61.12	81.03
LDAM [14]	64.58	83.52
IB	**65.39**	**84.98**

Experimental results on iNaturalist 2018.

We evaluated our method on the large-scale real-world image data, iNaturalist 2018. We compared our method with the state-of-the-art loss-based methods. Table 3.6 reveals that simply balancing the influence of loss could achieve considerable improvement.

3.4 Summary

In this chapter, we propose a novel influence-balanced loss to solve the overfitting of the majority classes in a class imbalance problem. A model trained on imbalanced class data is susceptible to overfitting due to the high capacity of DNN and the scarcity of samples in certain classes. Therefore, as learning progresses, existing methods are likely to produce undesirable results, such as assigning higher weights to samples from majority classes. Unlike the existing methods, IB loss can robustly assign weights because it directly focuses on a sample's influence on the model. We conducted experiments to demonstrate that our method can improve generalization performance under a class imbalance setting. In addition, our method is easy to be implemented and integrated into existing methods.

Chapter 4

Context-rich Minority Oversampling for Imbalanced Data

4.1 Overview

Real-world data are likely to be inherently imbalanced [119, 115, 43, 72], where the number of samples per class differs greatly. If models are trained on an imbalanced dataset, they can be easily biased toward majority classes and tend to have a poor generalization ability on recognizing minority classes (*i.e.,* overfitting).

A simple and straightforward method to overcome the class imbalance problem is to repeatedly oversample the minority classes [18, 160]. However, these naive oversampling can intensify the overfitting problem, since the repeatedly selected samples have less diversity but almost similar image contexts [142]. For example, consider a minority class of 'snow goose,' in which the geese always stand upon grass in the training images. If samples are drawn from these limited training samples [160] or even if new samples are produced by interpolating within the class [18], only **context-limited** images will be created as in Figure 4.1. Our goal is to solve the aforementioned problem by introducing a simple **context-rich** oversampling method.

We pay attention to the characteristics of long-tailed distributions; that is, majority class samples are data-rich and information-rich. Unlike the existing re-sampling methods that ignore (*i.e.*, undersample) majority samples, our method uses the affluent

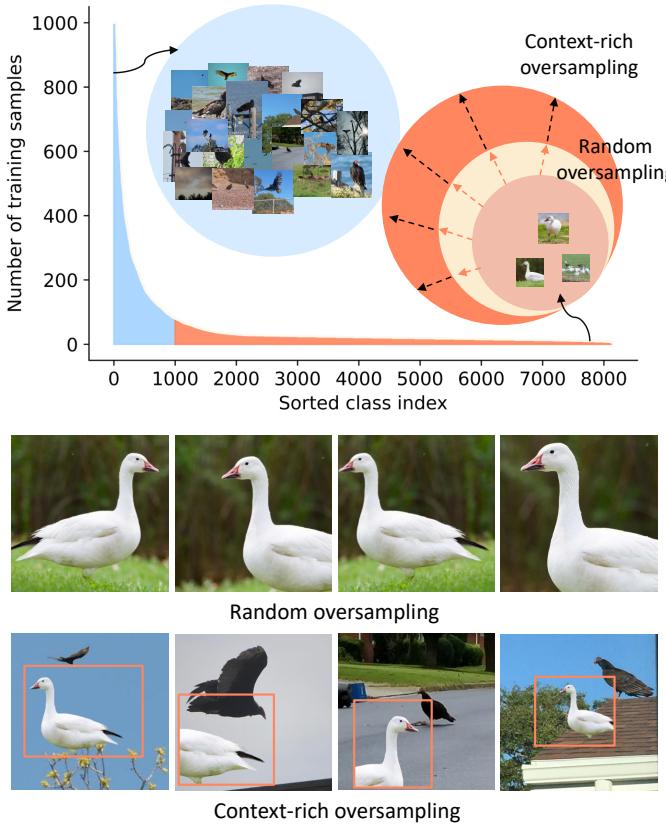


Figure 4.1: **Concept of context-rich minority oversampling**. In the real-world long-tailed dataset iNaturalist 2018 [72], the number of samples from the head class and the tail class is extremely different (Upper). Simple random oversampling method repeatedly produces context-limited images from minority classes. We propose a novel context-rich oversampling method to generate diversified minority images. To this end, we oversample the tail-class images with various sizes. Then, these patches are pasted onto the head-class images to have various backgrounds. Our key idea is to bring rich contexts from majority samples into minority samples.

information of the majority samples to generate new minority samples. Specifically, our idea is to leverage the rich major-class images as the background for the newly created minor-class images. Figure 4.1 illustrates the concept of our proposed context-

rich oversampling strategy. Given an original image from a minority class, the object is cropped in various sizes and pasted onto the various images from majority classes. Then, we can create images with more diverse contexts (*e.g.*, 'snow goose' images with the sky, road, roof, crows, etc). Since this is an interpolation of the majority and minority class samples, it generates diversified data around the decision boundary, and as a result, it improves the generalization performance for minority classes.

To this end, we adopt an image-mixing data augmentation method, CutMix [178]. As our key idea is to transfer rich contexts from majority to minority samples, we apply a simple and effective data sampling method to generate new minority-centric images with majority's contexts. However, naive use of CutMix may exacerbate the overfitting problem in favor of the majority classes because it may generate more majority-centric samples than minority samples. We solve this problem by sampling the background images and the foreground patches from different distributions to achieve the desired minority oversampling.

Our key contributions can be summarized as follows: (1) We propose a novel context-rich minority oversampling method that generates various samples by leveraging the rich context of the majority classes as background images. (2) Our method requires little additional computational cost and can be easily integrated into many end-to-end deep learning algorithms for long-tailed recognition. (3) We demonstrate that significant performance improvements and state-of-the-art performance can be achieved by applying the proposed oversampling to existing commonly used loss functions without any architectural changes or complex algorithms. (4) We empirically prove the effectiveness of the proposed oversampling method through extensive experiments and ablation studies. We believe that our study offers a useful and universal minority oversampling method for research into long-tailed classification.

4.2 Context-rich Minority Oversampling

4.2.1 Algorithm

We propose a new oversampling method called Context-rich Minority Oversampling (CMO). CMO utilizes the contexts of the majority samples to diversify the limited context of the minority samples. As shown in the Figure 4.1, the background images are sampled from majority classes and combined with foreground images of minority classes. Let $x \in \mathbb{R}^{W \times H \times C}$ and y denote a training image and its label, respectively. We aim to generate a new sample $(\tilde{x}, \tilde{y})$ by combining two training samples (x^b, y^b) and (x^f, y^f). Here, the image x^b is used as a background image, and the image x^f provides the foreground patch to be pasted onto (x^b, y^b).

For the image combining method, we chose CutMix [178] data augmentation due to its simplicity and effectiveness. Following CutMix [178] settings, the image and label pairs are augmented as

$$\begin{aligned} \tilde{x} &= \mathbf{M} \odot x^b + (\mathbf{1} - \mathbf{M}) \odot x^f \\ \tilde{y} &= \lambda y^b + (1 - \lambda) y^f, \end{aligned} \tag{4.1}$$

where $(\mathbf{1} - \mathbf{M}) \in \{0, 1\}^{W \times H}$ denotes a binary mask indicating where to select the patch and paste it onto a background image. $\mathbf{1}$ means a binary mask filled with ones, and $\odot$ is element-wise multiplication. The combination ratio $\lambda \in \mathbb{R}$ between two images is sampled from the beta distribution $Beta(\alpha, \alpha)$. To sample the mask and its coordinates, we apply the original CutMix [178] setting.

Since CutMix was originally designed for data augmentation on a class-balanced dataset, Eq. 4.1 does not represent the majority or minority class of samples. To change the method to CMO , we include sampling data distributions for foreground (x_f, y_f) and background samples (x^b, y^b). In our design, the background samples (x^b, y^b) should be biased to the majority classes. Therefore, we sample the background samples from the original data distribution P. Meanwhile, the foreground samples (x^f, y^f)

are sampled from minor-class-weighted distribution Q to be biased to the minority classes. In short, CMO consists of data sampling from two distributions, $(x^b, y^b) \sim P$ and $(x^f, y^f) \sim Q$, and combining the images using Eq. 4.1. The pseudo-code of the training procedure is presented in Algorithm 2.

Algorithm 2 Context-rich Minority Oversampling (CMO)

INPUT: Dataset $\mathcal{D}_{i=1}^N$, model parameters θ, P, Q, any loss function $L(\cdot)$.

Randomly initialize θ.
Sample weighted dataset $\tilde{\mathcal{D}}_{i=1}^N \sim Q$.
for epoch $= 1, \ldots, T$
  for batch $i = 1, \ldots, B$
    Draw a mini-batch (x_i^b, y_i^b) from $\mathcal{D}_{i=1}^N$
    Draw a mini-batch (x_i^f, y_i^f) from $\tilde{\mathcal{D}}_{i=1}^N$
    $\lambda \sim Beta(\alpha, \alpha)$
    $\tilde{x}_i = \mathbf{M} \odot x_i^b + (\mathbf{1} - \mathbf{M}) \odot x_i^f$
    $\tilde{y}_i = \lambda y_i^b + (1 - \lambda) y_i^f$
    $\theta \leftarrow \theta - \eta \nabla L((\tilde{x}_i, \tilde{y}_i); \theta)$
  end for
end for

4.2.2 Minor-class-weighted Distribution Q

To sample the foreground image from minority classes, we design the minor-class-weighted distribution Q by utilizing the re-weighting methods. The re-weighting approach, dating back to the classical importance sampling method [83], provided a way to assign appropriate weights to samples. Commonly used sampling strategies include ones that assign a weight inversely proportional to the class frequency [74, 166], to the smoothed class frequency [127, 124], or to the effective number [35].

Let n_k be the number of samples in the k-th class, then for the C classes, the total number of samples is $N = \sum_{k=1}^{C} n_k$. Then, the generalized sampling probability for

the k-th class can be defined by

$$q(r,k) = \frac{1/n_k^r}{\sum_{k'=1}^{C} 1/n_{k'}^r}, \tag{4.2}$$

where the k-th class has a sampling weight inversely proportional to n_k^r. As r increases, the weight of the minor class becomes increasingly larger than that of the major class. By adjusting the value of r, we can examine diverse sampling strategies. Setting $r = 1$ uses the inverse class frequency [74, 166] while setting $r = 1/2$ uses the smoothed inverse class frequency, as in [127, 124]. We can also use the effective number [35] instead of n_k^r, which is defined as

$$E(k) = \frac{(1-\beta^{n_k})}{(1-\beta)}, \tag{4.3}$$

where $\beta = (N-1)/N$. Since CMO is a new approach for long-tailed classification, it is hard to predict the performance of each sampling strategy for CMO . Therefore, we evaluate the different sampling strategies on the long-tailed CIFAR-100 [92] and select the best strategy $q(1,k)$ for the minor-class-weighted distribution Q. The experimental results are displayed in Table 4.11 of the experimental section.

4.2.3 Regularization Effect of CMO

A recent study [192] has reported that models trained on long-tailed datasets are more over-confident than the models trained on balanced data. In addition, the study reveals that the long-tailed classification accuracy can be improved by solving the over-confidence issue. Moreover, CMO can be interpreted as a way to mitigate over-confidence in long-tailed classification. Inherited from CutMix, CMO uses a soft-target label $\tilde{y}$, as in Eq. 4.1. The soft-target label penalizes over-confident outputs, similarly to the label smoothing regularization [156]. Therefore, we argue that CMO contributes not only to minority sample generation but also to mitigating the over-confidence, which both enable an impressive performance improvement in diverse long-tail settings. We will demonstrate the effectiveness of CMO through various experiments in the experimental section.

4.3 Experiments

We present experiments on and analyses of CMO in this section. We first describe our experimental settings and implementation details in Section 3.3.1. Next, we present the effectiveness of CMO using three long-tailed classification benchmarks: CIFAR-100-LT, ImageNet-LT, and iNaturalist. CMO consistently boosts the performance of these baselines with state-of-the-art accuracy (Section 4.3.2). In Section 4.3.3 we present in-depth analyses of CMO to study its inherent characteristics.

4.3.1 Experimental Settings

Datasets. We validate CMO on the most commonly used long-tailed recognition benchmark datasets: CIFAR-100-LT[14], ImageNet-LT [121], and iNaturalist 2018 [72] (see Table 4.1). CIFAR-100-LT and ImageNet-LT are artificially made imbalanced from their balanced versions (CIFAR-100 [92] and ImageNet-2012 [141]). The iNaturalist 2018 dataset is a large-scale real-world dataset that exhibits long-tailed imbalance. We used the official training and test splits in our experiments.

Table 4.1: **Summary of datasets**. The imbalance ratio ρ is defined by $\rho = \max_k\{n_k\}/\min_k\{n_k\}$, where n_k is the number of samples in the k-th class.

Dataset	# of classes	# of training	Imbalance ratio
CIFAR-100-LT	100	50K	{10, 50, 100}
ImageNet-LT	1,000	115.8K	256
iNaturalist 2018	8,142	437.5K	500

Evaluation Metrics. Performances is mainly reported as the overall top-1 accuracy. Following [121], we also report the accuracy of three disjoint subsets: Many-shot classes (classes that contain more than 100 training samples), medium-shot classes (classes that contain 20 to 100 samples), and few-shot classes (classes that contain under 20 samples).

Comparison methods. We compare CMO with the minority oversampling methods, the state-of-the-art long-tail recognition methods, and their combinations.

- **Minority oversampling.** (1) No oversampling (vanilla); (2) Random oversampling (ROS) [160], that oversamples minority samples to balance the classes in the training data; (3) Remix [26], which oversamples minority classes by assigning higher weights to the minority labels when using Mixup [182]; (4) Feature space augmentation (FSA) [27].

- **Re-weighting.** (5) label-distribution-aware margin (LDAM) loss [14], which regularizes the minority classes to increase margins to the decision boundary; (6) influence-balanced (IB) loss [134], which re-weights samples by their influences; (7) Balanced Softmax [138], an unbiased extension of Softmax; (8) LADE [70], which disentangles the source label distribution from the model prediction.

- **Other state-of-the-art methods.** (9) Deferred re-weighting (DRW) [14] and (10) Decouple [84] are two-stage algorithms that re-balance the classifiers during fine-tuning; (11) BBN [193] and (12) RIDE [165] use additional network branches to handle class imbalance; (13) Causal Norm [159], which disentangles causal effects and adjusts the effects in training; (14) MiSLAS [192], a two-stage algorithm, enhances classifier learning and calibration with label-aware smoothing (LAS) in stage-2.

Implementation. We use PyTorch [135] for all experiments. For the CIFAR datasets, we use ResNet-32 [64]. The networks are trained for 200 epochs following the training strategy in [14]. For ImageNet-LT, we use ResNet-50 as the backbone network. The network is trained for 100 epochs using an initial learning rate of 0.1. The learning rate is decayed at the 60th and 80th epochs by 0.1. For iNaturalist 2018, we use ResNet-$\{50, 101, 152\}$ and Wide ResNet-50 [179]. We train the networks for 200 epochs using an initial learning rate of 0.1, and decay the learning rate at epochs 75 and 160 by 0.1.

All experiments are trained with stochastic gradient descent (SGD) with a momentum of 0.9.

4.3.2 Long-tailed classification benchmarks

CIFAR-100-LT

We conduct experiments on CIFAR-100-LT using different imbalance ratios: 10, 50, 100. We apply CMO to various methods to verify its effectiveness on different algorithms: vanilla cross-entropy loss, class-reweighting loss (LDAM [14]), a two-stage algorithm (DRW [14]), and multi-branch architecture (RIDE [165]).

In addition, for more detailed results, we report the precision for the main baselines, 'CE + CMO', 'CE-DRW + CMO', and 'BS + CMO' in Table 4.2.

Table 4.2: Recall and Precision for CIFAR-100-LT (IB=100)

CIFAR-100 (IF=100)	Recall	Precision
CE + CMO	43.9	48.3
CE-DRW + CMO	47.0	46.4
BS + CMO	49.8	51.7

Comparison with state-of-the-art methods. The overall classification accuracies are displayed in Table 4.3. It is surprising that CMO with basic cross-entropy (CE) loss shows comparable performance to that of complex long-tail recognition methods. Moreover, applying CMO to the state-of-the-art model (*i.e.*, RIDE) further boosts the performance markedly, especially when the imbalance ratios are high as 50 and 100.

Comparison with oversampling methods. We further compare the performance improvement of CMO with that of other oversampling techniques when combined with long-tailed recognition methods (see Table 4.4). The results reveal that CMO consistently improves the performance of all long-tailed recognition methods. On the other hand, simply balancing the class distribution with ROS [160] severely degrades per-

Table 4.3: **State-of-the-art comparison on CIFAR-100-LT dataset.** Classification accuracy (%) for ResNet-32 architecture on CIFAR-100-LT with different imbalance ratios. $*$ and † are from the original paper and [70], respectively.

Imbalance ratio	100	50	10
Cross Entropy (CE)	38.6	44.0	56.4
CE-DRW	41.1	45.6	57.9
LDAM-DRW [14]	41.7	47.9	57.3
BBN [193]†	42.6	47.1	59.2
Causal Norm [159]†	44.1	50.3	59.6
IB Loss [134]*	45.0	48.9	58.0
Balanced Softmax (BS) [138]†	45.1	49.9	61.6
LADE [70]†	45.4	50.5	61.7
Remix [26]	45.8	49.5	59.2
RIDE (3 experts) [165]	48.6	51.4	59.8
MiSLAS [192]*	47.0	52.3	**63.2**
CE + CMO	43.9	48.3	59.5
CE-DRW + CMO	47.0	50.9	61.7
LDAM-DRW + CMO	47.2	51.7	58.4
BS + CMO	50.2	51.4	62.3
RIDE (3 experts) + CMO	**50.0**	**53.0**	60.2

formance. We speculate that this is because the naive balancing of the sampling distribution across classes hinders the model from learning generalized features for major classes and induces the model to memorize the minor class samples. Remix [26] improves the performance of some methods but degrades the performance when combined with RIDE [165]. This indicates that the simple labeling policy of Remix may not be effective when the model complexity becomes large, as in RIDE.

Table 4.4: **Comparison against baselines on CIFAR-100-LT (Imbalance ratio = 100).** Classification accuracy (%) of ResNet-32.

	Vanilla	+ROS [160]	+Remix [26]	+CMO
CE	38.6	32.3	40.0	**43.9**
	(+0.0)	(-5.3)	(+1.4)	**(+5.3)**
CE-DRW [14]	41.1	35.9	45.8	**47.0**
	(+0.0)	(-5.2)	(+4.7)	**(+5.9)**
LDAM-DRW [14]	41.7	32.6	45.3	**47.2**
	(+0.0)	(-9.1)	(+3.6)	**(+5.5)**
RIDE [165]	48.6	22.6	44.0	**50.0**
	(+0.0)	(-26.0)	(-4.6)	**(+1.4)**

ImageNet-LT

Comparison with state-of-the-art methods. The results of our method and other long-tailed recognition methods are displayed in Table 4.5. Applying CMO to the basic training with CE loss improves the performance by a significant margin, outperforming most of the recent baselines. The greater performance improvement on ImageNet-LT compared to CIFAR-100 indicates that our method benefits from the richer context information available in the major classes of ImageNet-LT. In addition, a consistent performance improvement by using CMO when combined with DRW or BS bolsters the efficacy of CMO , which can be easily integrated into modern state-of-the-art long-tailed recognition methods. It is noteworthy that as {CE-DRW + CMO } and {BS + CMO } especially achieve a much higher few-shot class accuracy than did the other methods, our method is useful for achieving consistent performance across classes. Lastly, applying CMO to RIDE further boosts performance, outperforming the results of RIDE with four experts.

Comparison with oversampling methods. In Table 4.6, we compare performance

Table 4.5: **State-of-the-art comparison on ImageNet-LT.** Classification accuracy (%) of ResNet-50 with state-of-the-art methods trained for 90 or 100 epochs. "$*$" and "†" denote the results are from the original papers, and [84], respectively. The best results are marked in bold.

	All	Many	Med	Few
Cross Entropy (CE)†	41.6	64.0	33.8	5.8
Decouple-cRT [84]†	47.3	58.8	44.0	26.1
Decouple-LWS [84]†	47.7	57.1	45.2	29.3
Remix [26]	48.6	60.4	46.9	30.7
LDAM-DRW [14]	49.8	60.4	46.9	30.7
CE-DRW	50.1	61.7	47.3	28.8
Balanced Softmax (BS) [138]	51.0	60.9	48.8	32.1
Causal Norm [159]*	51.8	62.7	48.8	31.6
RIDE (3 experts) [165]*	54.9	66.2	51.7	34.9
RIDE (4 experts) [165]*	55.4	66.2	52.3	36.5
CE + CMO	49.1	**67.0**	42.3	20.5
CE-DRW + CMO	51.4	60.8	48.6	35.5
LDAM-DRW + CMO	51.1	62.0	47.4	30.8
BS + CMO	52.3	62.0	49.1	**36.7**
RIDE (3 experts) + CMO	**56.2**	66.4	**53.9**	35.6

Table 4.6: **Comparison against baselines on ImageNet-LT.** Classification accuracy (%) of ResNet-50.

	Vanilla	+Remix [26]	+CMO
CE	41.6	41.7	**49.1**
	(+0.0)	(+0.1)	**(+7.5)**
CE-DRW [14]	50.1	48.6	**51.4**
	(+0.0)	(-1.5)	**(+1.3)**
Balanced Softmax [138]	51.0	49.2	**52.3**
	(+0.0)	(-1.8)	**(+1.3)**

improvement using other oversampling techniques. While CMO consistently improves performance for all methods, Remix [26] fails to improve the performance of the long-tailed recognition methods and barely improves the model trained with cross-entropy loss. This implies that the labeling strategy of Remix is not sufficient to compensate for the adverse effect of using the same original distribution as the two sampling distributions of the mixup method, especially when the imbalance ratio rises severly to 256, as with ImageNet-LT. In contrast, CMO generates more minority samples by using different distributions when selecting two images and produces much better classification accuracy on all tasks.

Results on longer training epochs. Recently, PaCo [34] performed impressively by using supervised contrastive learning. Since contrastive learning requires diverse augmentation strategies and longer training times, PaCo trained networks for 400 epochs using RandAugment [33]. Since CMO should also improve using longer training epochs, we evaluate CMO using the same setting from PaCo (*i.e.*, 400 epochs & RandAug). Table 4.7 reveals that {BS + CMO } achieves a new state-of-the-art performance. It is noteworthy that applying CMO significantly surpasses the two baselines, especially in the few-shot classes. On top of its simplicity and much lower computational cost, the results demonstrate the effectiveness of the proposed method.

Table 4.7: **Results on longer training epochs with RandAugment [33].** Classification accuracy (%) of ResNet-50 on ImageNet-LT. "$*$" denotes the results from [34].

	All	Many	Med	Few
BS*	55.0	66.7	52.9	33.0
PaCo [34]*	57.0	65.0	**55.7**	38.2
BS + CMO	**58.0**	**67.0**	55.0	**44.2**

Table 4.8: **State-of-the-art comparison on iNaturalist2018.** Classification accuracy (%) of ResNet-50 on iNaturalist2018. "$*$" and "†" indicate the results from the original paper and [193], respectively. RIDE [165] was trained for 100 epochs.

	All	Many	Med	Few
Cross Entropy (CE)	61.0	73.9	63.5	55.5
IB Loss [134]*	65.4	-	-	-
FSA [27]*	65.9	-	-	-
LDAM-DRW [14]†	66.1	-	-	-
Decouple-cRT [84]*	68.2	73.2	68.8	66.1
Decouple-LWS [84]*	69.5	71.0	69.8	68.8
BBN [193]*	69.6	-	-	-
Balanced Softmax [138]	70.0	70.0	70.2	69.9
LADE [70]*	70.0	-	-	-
Remix [26]*	70.5	-	-	-
MiSLAS [192]*	71.6	73.2	72.4	70.4
RIDE (3 experts) [165]*	72.2	70.2	72.2	72.7
RIDE (4 experts) [165]*	72.6	70.9	72.4	**73.1**
CE + CMO	68.9	**76.9**	69.3	66.6
CE-DRW + CMO	70.9	68.2	70.2	72.2
LDAM-DRW + CMO	69.1	75.3	69.5	67.3
BS + CMO	70.9	68.8	70.0	72.3
CE-DRW + CMO + LAS [192]	71.8	69.6	72.1	71.9
RIDE (3 experts) + CMO	**72.8**	68.7	**72.6**	**73.1**

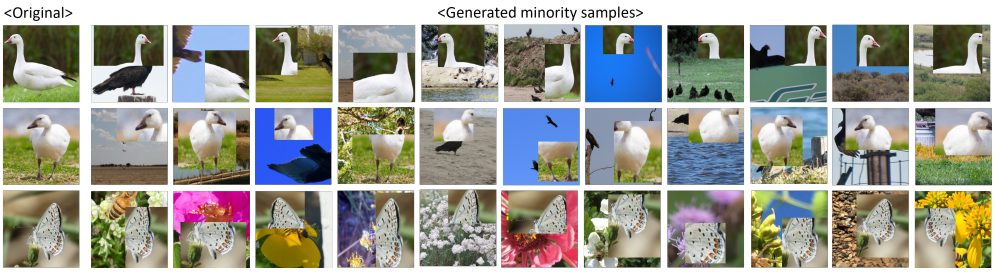


Figure 4.2: **A display of the minority images generated by CMO** (minority classes: the snow goose and the Acmon blue (butterfly)). We randomly choose generated images for each original image. Our method is able to generate context-rich minority samples that have diverse contexts. For example, while the original 'snow goose' class contains only images of a 'snow goose' on grass, the generated images have various contexts such as the sky, the sea, the sand, and a flock of crows. These generated images enable the model to learn a robust representation of minority classes.

iNaturalist 2018

Comparison with state-of-the-art methods. Table 4.8 presents the classification results. On the naturally-skewed dataset, applying CMO to the simple training scheme of CE-DRW surpasses most of the state-of-the-arts. On iNaturalist 2018, as in ImageNet-LT, CMO dramatically improves the performance of the cross-entropy loss (CE) by **7.9%p** (61.0% increased to 68.9%). This is because the sample generation by CMO fully utilizes the abundant context of training data. Again, it achieves a remarkable performance improvement in the few-shot classes. It is moreover noteworthy that when we apply the same stage-2 strategy, LAS, from [192], it further boosts performance. Lastly, applying CMO to RIDE achieves a new state-of-the-art performance.

Results on large models. We investigate the performance of CMO and other oversampling methods using the large deep networks of Wide ResNet-50 [179], ResNet-101, and ResNet-152 [64]. We compare CMO with the feature space augmentation method

(FSA) [27]. While both methods improve the results from vanilla training with cross-entropy loss, our method provides superior performance to that of FSA. This indicates that using the context-rich information from majority classes in the input space is simple but effective in improving the overall performance.

Table 4.9: **Results on large architectures.** Classification accuracy (%) of large backbone networks on iNaturalist 2018. The results are copied from [27].

Method	ResNet-50	Wide ResNet-50	ResNet-101	ResNet-152
CE	61.0	-	65.2	66.2
FSA [27]	65.9	-	68.4	69.1
CMO	**70.9**	**71.9**	**72.4**	**72.6**

Display of the generated images. We visualize the generated images for the minority classes in Figure 4.2. From the rarest minority classes, we randomly choose generated images for each original image. CMO produces diverse minority samples that have various contexts. For example, while the 'snow goose' class contains only images of geese on grass, the generated images have various contexts, such as the sky or sea. Likewise, the butterflies in the third row are newly created as diverse images that have various contexts, containing bees and flowers of various colors and shapes. We argue that various combinations of context and minority samples encourage the model to learn a robust representation of the minority classes.

4.3.3 Analysis

Is the distribution for augmenting images important? To justify the need for different distributions of background and foreground images, we compare CutMix and CMO . As can be seen from Table 4.10, CMO outperforms CutMix on long-tailed classification by a large margin. In particular, there is a remarkable performance improvement in the medium and few-shot classes. The performance gap is due to the absence of a minor-class-weighted distribution in CutMix augmentation. Although CutMix can

generate informative mixed samples, its effect is limited when used with long-tailed distributions. Thus, we claim that the use of a minor-class-weighted distribution is a key-point in data augmentation in the long-tailed settings; this highlights the contribution and originality of CMO .

Table 4.10: **Comparison with CutMix** using cross-entropy loss.

	All	Many	Med	Few
CIFAR-100-LT				
CutMix	35.6	71.0	37.9	4.9
CMO	43.9	70.4	42.5	14.4
ImageNet-LT				
CutMix	45.5	68.6	38.1	8.1
CMO	49.1	67.0	42.3	20.5

How to choose the appropriate probability distribution Q. We evaluate different sampling strategies in Section 4.2.2 on CIFAR-100 with the imbalance ratio 100, The results are reported in Table 4.11. $q(1, k)$ displays the most balanced performance. This result is consistent with the common practice of balancing the dataset by assigning weights inversely proportional to the class frequency. While $q(2, k)$, which imposes a higher probability on the minority class than does $q(1, k)$, performs acceptably in the few-shot classes, the overall performance slightly deteriorates. We assume this is because we cannot sample more diverse images when imposing too high probabilities on the few-shot classes. Based on this result, we set Q as $q(1, k)$ in our all experiments.

Why should we oversample only for the foreground samples? One may wonder why oversampling only for the foreground samples is better than oversampling both patches and background samples or oversampling only the backgrounds. To verify our design choice, we evaluate two variants of CMO . The first variant, CMO $_{back}$, samples

Table 4.11: **Impact of different Q sampling distributions.** Results on CIFAR-100-LT (imbalance ratio=100) according to different Q sampling probabilities.

	All	Many	Med	Few
$q(1/2, k)$	42.6	**71.6**	42.1	9.5
$q(1, k)$	**43.9**	70.4	**42.5**	**14.4**
$q(2, k)$	40.1	67.2	36.7	12.3
$E(k)$ [35]	39.5	70.4	38.0	4.7

background images from a minor-class-weighted distribution and patches from the original distribution, which is exactly the opposite design of CMO , i.e., $(x^b, y^b) \sim Q, (x^f, y^f) \sim P$. The second variant, CMO $_{minor}$, samples both the background and the patches from a minor-class-weighted distribution, *i.e.*, $(x^b, y^b), (x^f, y^f) \sim Q$. We report the results of applying these variants of the CMO method to the model trained with CE loss and LDAM loss [14] in Table 4.12.

Table 4.12: **Ablation study.** Results from variants of CMO with ResNet-32 on imbalanced CIFAR-100; imbalance ratio of 100.

	All	Many	Med	Few
Cross Entropy (CE)	38.6	65.3	37.6	8.7
CE + CMO $_{minor}$	37.9	58.3	40.4	11.2
CE + CMO $_{back}$	40.1	64.7	40.2	11.3
CE + CMO	**43.9**	**70.4**	**42.5**	**14.4**
LDAM [14]	41.7	61.4	42.2	18.0
LDAM + CMO $_{minor}$	31.7	50.2	33.2	8.4
LDAM + CMO $_{back}$	44.2	59.2	46.6	24.0
LDAM + CMO	**47.2**	**61.5**	**48.6**	**28.8**

CMO $_{minor}$ yields severe performance degradation using both methods. We suspect that this is because the rich context of the majority samples cannot be utilized.

In contrast, CMO $_{back}$ produces acceptable performance improvements, but far less than did the original CMO . This is because, using the CutMix method, there is a high probability that the object in the foreground image overlaps the background image. Therefore, we can expect a loss of information about minority classes in the background image, resulting in a limited performance boost.

Comparison with other minority augmentations. To further verify our design choice, we analyze the effectiveness of using different augmentation methods, including CutMix [178], Mixup [182], color jitter, and Gaussian blur. For Mixup, we use the same sampling strategy as for CMO . For color jitter and Gaussian blur, which do not interpolate two images, we apply augmentation only to the minority classes and oversample those classes. As evidenced in Table 4.13, other augmentation methods provide little performance gain compared to the gains using CutMix. We suspect that this is because the pixel-level transformations (*i.e.*, Gaussian blur and color jitter) are not effective in producing minority samples that have a rich context. Gaussian blur and color jitter do not combine two images; thus, it is hard to add a new context to minority samples. While Mixup combines two images, it does not distinguish the roles of the two samples, limiting the control of the source of the context and of the patch information. In contrast, CutMix can create diverse images with larger changes at pixel-level than can other methods.

4.4 Summary

We have proposed a novel context-rich oversampling method, CMO , to solve the data imbalance problem. We tackle the fundamental problem of previous oversampling methods that generate context-limited minority samples, which intensifies the overfitting problem. Our key idea is to transfer the rich contexts of majority samples to minority samples to augment minority samples. The implementation of CMO is sim-

Table 4.13: **Data augmentation methods.** Comparisons between augmentation methods for generating new minority samples on CIFAR-100-LT with an imbalance ratio of 100.

	All	Many	Med	Few
CMO w/ Gaussian Blur	31.1	54.7	28.8	6.2
CMO w/ Color Jitter	34.7	58.9	34.4	6.8
CMO w/ Mixup	38.0	54.8	40.2	**15.9**
CMO w/ CutMix	**43.9**	**70.4**	**42.5**	14.4

ple and intuitive. Extensive experiments on various benchmark datasets demonstrate not only that our CMO significantly improves performance, but also that adding our oversampling method to the basic losses advances the state-of-the-art.

Chapter 5

Influential Rank: Post-training for Noisy Labels

5.1 Overview

Real-world data inevitably contain some proportion of incorrectly labeled data, owing to perceptual ambiguity, or errors from human or machine annotations. These noisy labels negatively affect the generalization performance of a trained model since a deep neural network (DNN) can easily overfit to even noisy labels due to its high capacity [180]. Therefore, learning from noisy labels (LNL) has received much attention in recent years [67, 174, 147, 191, 112, 24, 76] due to the increasing need to handle noisy labels in practice.

To handle noisy-label problem, prior literature aims to distinguish between clean and mislabeled data, and use this information to train a robust classifier during training. To this end, prior works mainly rely on the assumption that the clean labels are more likely to have smaller losses before the model is overfitted [7]. However, due to the high capacity of deep neural networks (DNNs), DNNs can fit even noisy labels [180]; thus it is challenging to correctly detect mislabeled data during training. Hence, various methods have been proposed to use more robust models before overfitting, such as leveraging the model with early stopping [148, 108], or using multiple networks with co-training for sample selection [60, 177, 102].

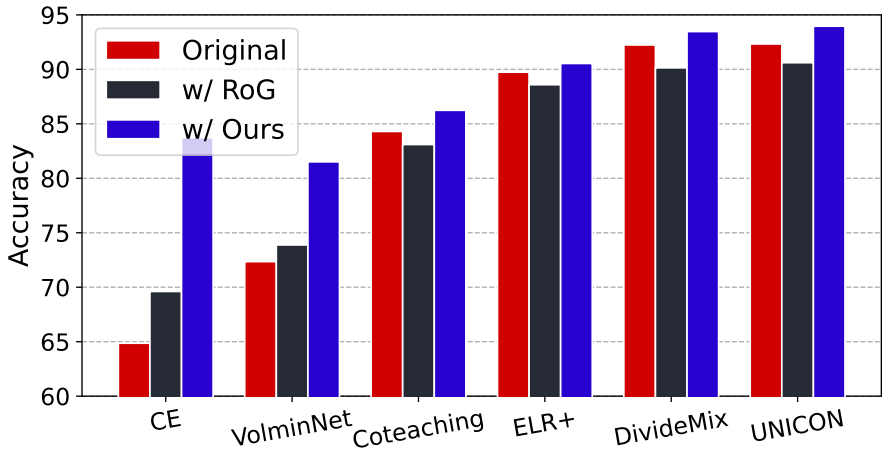


Figure 5.1: **Test accuracy improvement over various methods on CIFAR-10N (Worst).** As a post-training method, our proposed Influential Rank can improve various pre-trained models by large margin, compared to the post-processing baseline method, RoG [96]. The used CIFAR-10N (Worst) is a human-annotated real-world noisy dataset with about 40% noise rate [169].

Here, we introduce a different perspective against the mainstream research. We propose a new post-training LNL approach, which can synergize with the model trained using prior robust methods, further enhancing the generalization capability of the model. Given a pre-trained model, the proposed post-training scheme refines the model by exploiting the 'overfitting property' of mislabeled samples. 'Overfitting property' of mislabeled samples is derived from two following intuitions. (1) Mislabeled samples are more likely to distort the decision boundary than clean samples. Thus removing the mislabeled samples is likely to sway the decision boundary significantly. (2) The overfitted model predicts poorly on unseen data, and the mislabeled sample is usually the main culprit for the model to classify new data with incorrect labels. The details on these intuitions are discussed in Section 5.2.1.

These intuitions on overfitting motivate us to propose a novel method named **Influential Rank**, which leverages the samples' influence on the decision boundary and on unseen samples to enhance robustness. To this end, we propose *overfitting score on*

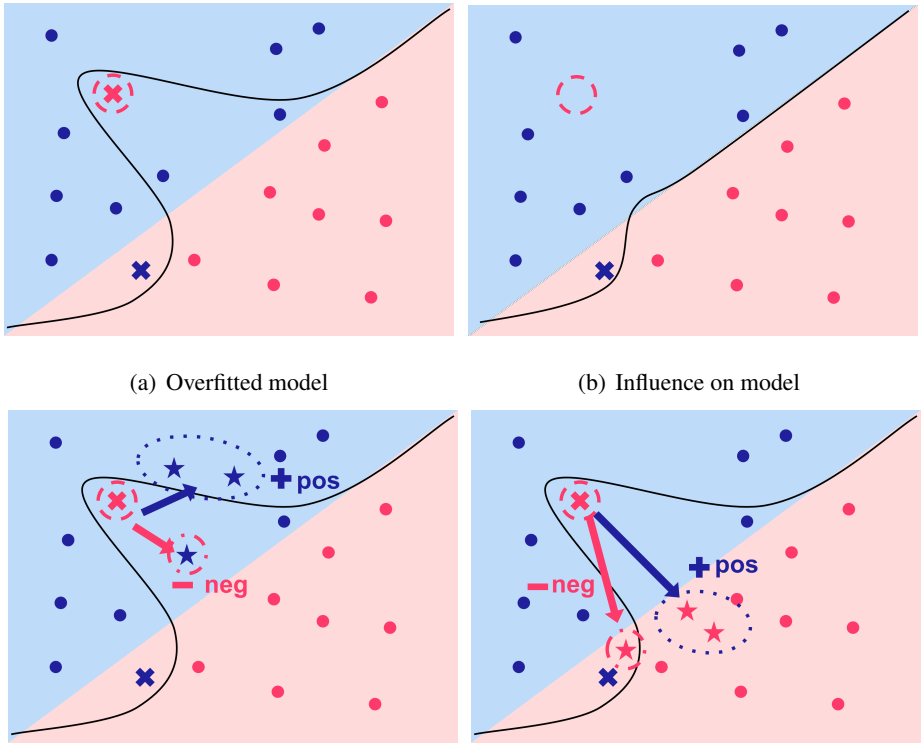
model (OSM) and *overfitting score on data* (OSD). OSM measures the influence of a training sample on changes in model parameters, and OSD measures the inconsistency of the sample's influence on the classification prediction for a small number of clean validation data. Based on OSM and OSD, Influential Rank updates the trained model by removing high influential samples and mitigating their negative influence on the classifier.

Since the post-training provides a new information (*i.e.*, sample's influence) to any pre-trained models, Influential Rank can effectively improve robustness of existing LNL methods. Through extensive experiments on multiple benchmark data sets, we demonstrate the validity of our method, and show that Influential Rank can improve the performance of the model consistently whether or not it is pre-trained with LNL methods, as shown in Figure 5.1. Furthermore, we show that Influential Rank is useful in two different applications other than LNL. The proposed overfitting scores can be effective for (1) data cleansing that filters out erroneous examples in *real-world video data* and (2) *regularization* that boosts the classification performance on clean data.

Our key contributions can be summarized as follows: (1) **Post-training**: Influential Rank is a novel post-training approach for LNL, which leverages the *overfitting scores* of training examples on the decision boundary. (2) **Practicability**: Influential Rank is applicable to any pre-trained models and works synergistically with other existing LNL methods. (3) **Extensibility**: Influential Rank can be easily extended to cleansing video dataset and a regularization for reducing overfitting arising from clean but influential samples.

5.2 Influential Rank

Our idea is to leverage the property of an overfitted model for post-processing. First, we present the observations that motivated our method in Section 5.2.1. Then, we propose two novel criteria in Section 5.2.2, and we describe the overall scheme of



(a) Overfitted model

(b) Influence on model

(c) Influence on data

(d) Influence on data

Figure 5.2: **Our intuition.** The red and blue points belong to different classes in binary classification. The $\times$ marks indicate mislabeled data. (a) Due to the mislabeled samples ($\times$), the model is overfitted. (b) $\times$ significantly affects the model because if the sample is removed, the parameter of the model is substantially changed. (c and d) Assume clean validation data ($\bigstar$) are given. The noisy-label sample ($\times$) exerts both positive and negative influences on correctly classifying the validation data in the same class, even when distances are near. The noisy-label data tend to have inconsistent effects on data within the same class.

robust post-training with overfitting scores, referred to as Influential Rank (Section 5.2.3). Finally, we empirically verify the effectiveness of the proposed criteria in post-training from a toy example (Section 5.2.4).

5.2.1 Intuition

Our post-training algorithm is based on two following intuitions. Mislabeled samples are likely to significantly distort the decision boundary, and to cause misclassification of nearby correctly labeled samples. Figure 5.2 illustrates our intuition. The red and blue points belong to different classes for binary classification, and the pink and light blue background indicates the ground-truth feature embedding space. Black line denotes a decision boundary predicted by the model. In Figure 5.2(a) the model overfits the mislabeld samples ($\times$ (red) mark), thus the decision boundary is distorted compared to the ground-truth boundary. When the mislabeled sample ($\times$ (red) mark) is removed, the trained model is substantially changed (Figure 5.2(b)). That is, the noisy label can exert great influence on the decision boundary of the model.

In addition, to evaluate whether a training sample causes a significantly overfitted classifier, we can use a small number of clean validation data. We consider a few validation data points[1] ($\bigstar$ (blue) marks) as shown in Figure 5.2(c). Because the fitted decision boundary is distorted toward the blue region to include the noisy label ($\times$ (red) mark), the $\bigstar$ (blue) enclosed by a red dotted circle is wrongly classified into the red class. Thus, the noisy label ($\times$ (red) mark) causes a clean sample to be misclassified (i.e., negative influence). Meanwhile, the validation samples upper the line (blue-dotted circle) are correctly classified that it can be said that the boundary created by this mislabeled sample ($\times$ (red)) has a positive influence on properly classifying other samples. Therefore, the noisy label is likely to have inconsistent influences on the clean validation samples, although their distances are near each other. The same claim can apply to the validation samples ($\bigstar$ (red)) in the other (red) category in Figure 5.2(d). We verify the inconsistent influences of noisy labels in Section 5.3.5.

From this observation, we present two novel criteria that measure the abnormal influences of a training sample. One is to measure how much a training sample affects the overfitting of model parameters, referred to as the *overfitting score on model*, and

[1] We use only 5 data per class.

the other measures how inconsistently a training sample affects the classification of clean validation data, which is referred to as the *overfitting score on data*.

5.2.2 Overfitting Scores

To identify overfitting on individual points for detecting noisy labels, we utilize two influence functions in [89]. One is to measure the influence of an example (x, y) on the model $f(x, \hat{\theta})$ trained on the dataset $\mathcal{D}$ via loss function $\ell(y, f(x, \theta))$, given by

$$\mathcal{I}_M(x; \hat{\theta}) = -H_{\hat{\theta}}^{-1} \, \nabla_\theta \ell(y, f(x, \theta))|_{\theta=\hat{\theta}}\,, \tag{5.1}$$

where $H_{\hat{\theta}} \overset{\text{def}}{=\!=} \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \nabla_\theta^2 \ell\big(y, f(x, \theta)\big)\big|_{\theta=\hat{\theta}}$. The other is to measure the influence of a training sample (x_i, y_i) on a test sample (x_t, y_t), given by

$$\mathcal{I}_D(x_i, x_t; \hat{\theta}) = \nabla_\theta \ell(y_t, f(x_t, \hat{\theta}))^\top \mathcal{I}_M(x_i; \hat{\theta}). \tag{5.2}$$

Overfitting Score on Model

$\mathcal{I}_M(x; \hat{\theta})$ can be used to estimate the effect of a noisy label on an overfitted model (Figure 5.2(b)). However, $\mathcal{I}_M(x; \hat{\theta})$ is a p-dimensional vector, where p is the number of model parameters. Thus, to measure the strength of the influence of a training point (x_i, y_i), we use $\|\mathcal{I}_M(x_i; \hat{\theta})\|$ as a metric. Using this metric, we define *overfitting score on model* (OSM) $\mathcal{O}_M(x_i; \hat{\theta})$ as the model (parameter)'s potential change caused by ignoring the example x_i for training,

$$\mathcal{O}_M(x_i; \hat{\theta}) = \frac{\|\mathcal{I}_M(x_i; \hat{\theta})\| - \mu_{x\in\mathcal{D}}\big(\|\mathcal{I}_M(x; \hat{\theta})\|\big)}{\sigma_{x\in\mathcal{D}}\big(\|\mathcal{I}_M(x; \hat{\theta})\|\big)}, \tag{5.3}$$

where $\mu_{x\in\mathcal{D}}(\cdot)$ and $\sigma_{x\in\mathcal{D}}(\cdot)$ denote mean and standard deviation of $\|\mathcal{I}_M(x; \hat{\theta})\|$ over $x \in \mathcal{D}$, respectively.

OSM $\mathcal{O}_M(x_i; \hat{\theta})$ measures a normalized global influence of a training sample x_i on the entire parameters. As in Figure 5.2(b), the noisy samples are likely to locate near the decision boundary, therefore, they will exhibit a higher OSM than examples with clean labels.

Overfitting Score on Data

In contrast to a well-generalized decision boundary, an overfitted decision boundary by a mislabeled sample makes the mislabeled sample inconsistently affect clean validation samples, even though the validation samples belong to the same class (Figure 5.2(c) and 5.2(d)). Here, an influence of a training sample on a validation sample indicates how much a classification result of the validation sample changes after removing the training sample. Therefore, we suggest overfitting score on data (OSD) as the within-class influence consistency of a training sample x_i on m clean validation samples in $\mathcal{D}_k = \{x_1^v, \cdots x_m^v\}$ in the k-th class. Utilizing (5.2), OSD $\mathcal{O}_D^k(x_i; \hat{\theta})$ in the k-th class is defined by

$$\mathcal{O}_D^k(x_i; \hat{\theta}) = \frac{\sigma_k\big(\mathcal{I}_D\big(x_i, x^v; \hat{\theta}\big)\big) - \mu\big(\sigma_k\big(\mathcal{I}_D\big(x, x^v; \hat{\theta}\big)\big)\big)}{\sigma\big(\sigma_k\big(\mathcal{I}_D\big(x, x^v; \hat{\theta}\big)\big)\big)}, \tag{5.4}$$

where $\sigma_k(\cdot)$ is standard deviation of $\mathcal{I}_D\big(x, x^v; \hat{\theta}\big)$ over $x^v \in \mathcal{D}_k$, whereas $\mu(\cdot)$ and $\sigma(\cdot)$ denote mean and standard deviation of $\sigma_k(\cdot)$ over k.

5.2.3 Post-processing with Influential Rank

Algorithm 3 outlines the overall procedure of Influential Rank. Given a pre-trained model, Influential Rank updates the model parameter with the training dataset excluding highly influential examples (*i.e.*, potentially mislabeled examples) for a fixed number of post-training epochs, which is much smaller than the total training epochs of the pre-trained model. Specifically, given a pre-trained model $\hat{\theta}_0$, we calculate $\mathcal{O}_M(x_i; \hat{\theta})$ for the whole training dataset $\mathcal{D}$ (Line 3). Since our goal is to exclude examples that have high scores for both $\mathcal{O}_M(x_i; \hat{\theta})$ and $\mathcal{O}_D^k(x_i; \hat{\theta})$, we compute $\mathcal{O}_D^k(x_i; \hat{\theta})$ for the training samples whose $\mathcal{O}_M(x_i; \hat{\theta})$ are higher than the mean (*i.e.*, 0) for efficient computation.

To automatically quantify the number of influential samples that need to be eliminated, we assume a two-modality Gaussian mixture model (GMM). First, we fit the two-modality GMM to $\mathcal{O}_D^k(x_i; \hat{\theta})$ using the Expectation-Maximization algorithm. Next,

Algorithm 3 Influential Rank

INPUT: $\mathcal{D}$: data, $\hat{\theta}_0$: pre-trained model, $epochs$: post-training epochs, γ: consensus number

OUTPUT: θ_r: model parameter after post-training

1: $\mathcal{C} \leftarrow \mathcal{D}$ /* $\mathcal{C}$ is entire clean samples in $\mathcal{D}$ */
2: **repeat**
3: $\mathcal{D}_M \leftarrow \{x_i | \{\mathcal{O}_M(x_i; \hat{\theta}_r)\}_{i=1}^n \geq 0\}$ // Compute Eq. (5.3)
4: **for** class $k = 1$ **to** K **do**
5: /* Compute Eq. (5.4) and fit GMM (G_{low}, G_{high}) */
6: $\mathcal{D}_D^k \leftarrow \{x_i | \{\mathcal{O}_D^k(x_i; \hat{\theta}_r)\}_{i \in \mathcal{D}_M} \geq \mu(G_{low})\}$
7: **end for**
8: $\mathcal{S} \leftarrow \{x_i | \sum_{k=1}^K 1[x_i \in \mathcal{D}_D^k] \geq \gamma\}$
9: $\mathcal{C} \leftarrow \mathcal{C} - \mathcal{S}$ /* Update the clean set */
10: Post-train $\hat{\theta}_r$ on the refined clean set $\mathcal{C}$ for $epochs$
11: **until** acc$(\hat{\theta}_r)$ saturates
12: **return** $\hat{\theta}_r$

we select the training samples whose $\mathcal{O}_D^k(x_i; \hat{\theta})$ are higher than the smaller mean of the Gaussian component (Line 6). We referred to those samples as *noisy candidates*. Then, we decide the final influential samples if a noisy candidate is inconsistent for more than γ classes in common (*i.e.*, more than γ classes have *consensus* that the noisy candidate is inconsistent), which are referred as *noisy-probable samples* (Line 8).

After removing all the noisy-probable samples, the model is retrained for a small number of epochs using the new training set (Line 9, 10). If a meaningful improvement in the classification accuracy occurs, the noisy-probable samples are eliminated from the training set, and the algorithm is repeated. Otherwise, the noisy-probable samples are not removed, and the algorithm stops.

When the algorithm finishes, new labels of the removed samples are predicted by

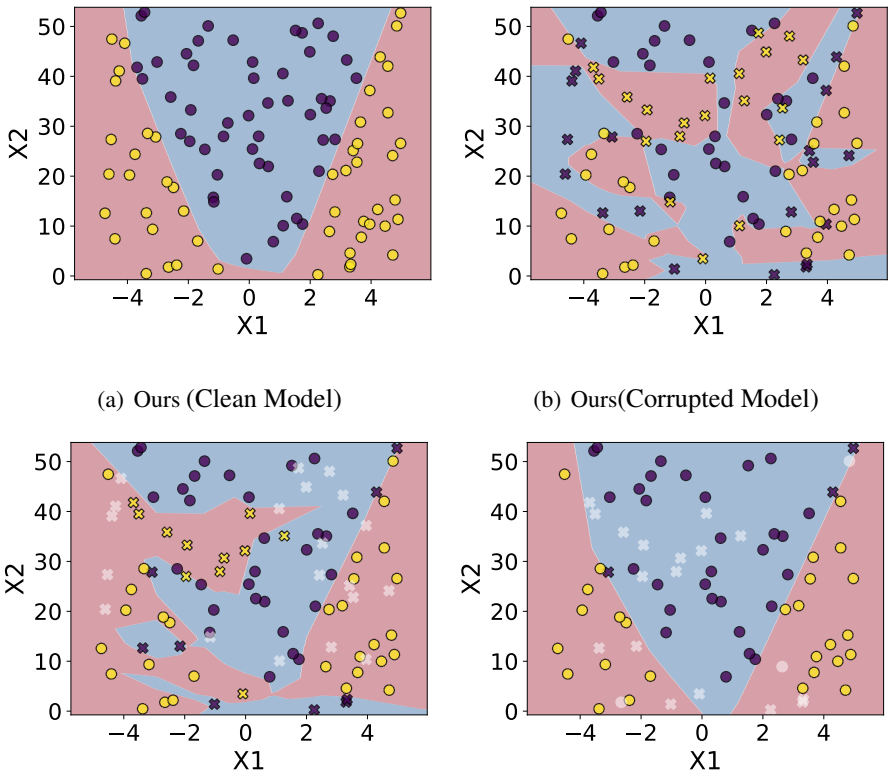
the classifier in the last iteration. Simply, we replace the labels of the noisy data with the newly corrected labels. Then, among the corrected training data, the new clean dataset includes only the data whose softmax outputs are higher than S(prediction threshold). Then, the model is newly trained on the new clean dataset and is evaluated for the test dataset.

This iterative design allows to remove more mislabeled examples in an iterative manner. As the model evolves, Influential Rank can incrementally find hard-to-identify mislabeled examples that could not be detected in the previous round. Especially under the high noise-level circumstances *e.g.*, 70% of label noise, multi-round post-training achieves significant performance gains.

5.2.4 Example: A Binary Classification

We present a toy example to verify and visualize our hypothesis and the efficacy of the proposed overfitting scores. Figure 5.3 illustrates the toy example. In Figure5.3, yellow and purple circles represent examples of two different classes, and blue and pink shades indicate their decision surfaces. For the two-dimensional binary classification problem, we first generate 100 data points from the uniform distribution, where $x_1 \sim \mathrm{Unif}(-5, 5)$ and $x_2 \sim \mathrm{Unif}(0, 55)$, and their true labels y are assigned following the binary rule depending on their (x_1, x_2) values, $y = 1$ if $x_2 \geq 3x_1^2$ and $y = 0$ if $\mathrm{otherwise}$.

Figure 5.3(a) shows that the decision boundary trained on clean data is well-formed close to the ground truth. Next, for the label noise scenario, 40% of the true labels are randomly corrupted in data, *i.e.*, $\times$ marks in Figure 5.3(b). Then, we fit a two-layer feedforward neural network with 50 hidden neurons. When trained with noisy labels shown in Figure 5.3 3(b), we observe that the trained model overfits to mislabeled examples, and forms a complex decision boundary such that many mislabeled examples locate near the overfitted decision boundary. When we post-train the model after excluding 20 examples with high overfitting scores (*i.e.*, white examples), the overfitted



(a) Ours (Clean Model)

(b) Ours(Corrupted Model)

(c) OursOurs (after 1st iter.)

(d) OursOurs (after 3rd iter.)

Figure 5.3: **Change in decision boundary by Influential Rank.** (a) DNN trained on clean data. (b) DNN trained on noisy data (randomly chosen 40% of labels are flipped). (c) DNN after first iteration by Ours. (d) DNN after third iteration by Ours.

decision boundary begins to recover in Figure 5.3 3(c). Again, after excluding total 20 more high influential examples after the third iteration in Figure 5.3 3(d), the decision boundary becomes almost similar to that of the clean model. Therefore, this toy example illustrates the validity of Influential Rank for robust post-training.

[1]There are 'random1', 'random2', and 'random3', but we use 'random1' since they have the same noise rate of 18%.

Table 5.1: **Summary of datasets.**

Dataset	# of training	noise ratio (ε)	noise type
CIFAR [92]	50K	20, 50, 70	synthetic
CIFAR-N [169]	40K	9, 18, 40	real-world
WebVision 1.0 [110]	2.4M	20	real-world
Clothing1M [172]	1M	38	real-world

5.3 Experiments

5.3.1 Experimental Settings

Datasets. We conduct classification on multiple benchmark datasets, including synthetic noisy labels and real-world noisy labels: CIFAR-10, CIFAR-100 [92], and their extension with real-world human labels CIFAR-N [169]; a large-scale real-world noisy data, WebVision 1.0 [110], Clothing1M [172] (Table 5.1).

For CIFAR-10 and CIFAR-100, noisy labels are injected using the symmetric noise [60] of flipping true labels into other labels with equal probability ε, *i.e.*, the noise ratio. Regarding the real-world noisy data, CIFAR-N [169] has various versions of human noise level. 'aggregate' (9%), 'random' (18%)[1], and 'worst' (40%), while CIFAR-100N has only a single version, 'noisy' (40%). Clothing1M includes about 38% real noisy labels, and WebVision 1.0 contains about 20% real-world noisy labels [149]. Following the previous work [20], we only use the first 50 classes of the Google image subset in WebVision. Lastly, we use a video stream data, HMDB-51 [146], to verify that our method can be effective as a detector for data cleaning.

To illustrate the applicability of our algorithm to video streams, we experiment on HMDB-51, a popular dataset frequently used in video action recognition [146]. Clothing1M and WebVision 1.0 are large-scale real-world datasets. Clothing1M includes about 38% real noisy labels and WebVision 1.0 contains about 20% noisy labels. Following previous work [20], we compare baseline methods on the first 50 classes of

the Google image subset. Furthermore, to illustrate the applicability of our algorithm to video streams, we experiment on HMDB-51, a popular dataset frequently used in video action recognition [146].

Implementation Details. Following the prior literature [116], all the compared methods are trained using ResNet-34, Inception-ResNet V2, and ResNet-50 for CIFAR, WebVision datasets, and Clothing1M respectively. For all experiments, the last fully connected (FC) layers in the networks are used as the overfitted classifiers. In addition, to reduce the number of the classifier parameters, we add a penultimate FC layer with 50, 100, 100 neurons, for CIFAR-100, WebVision 1.0, and HMDB-51, respectively. This allows to save the computational cost of hessian computation. Lastly, for label refinement, we set the threshold S to 0.8.

CIFAR and CIFAR-N

All networks are trained for 120 epochs for CIFAR-10(N), and 150 epochs for CIFAR-100(N) with Stochastic Gradient Descent (SGD) (momentum=0.9). Regarding to training with CE, we set the initial learning rate as 0.1, and reduce it by a factor of 10 after 40 and 80 epochs for CIFAR-10(N). For CIFAR-100(N), the initial learning rate is decayed at 60th and 100th epoch by 0.1. To implement LNL baselines, we set the hyperparameters and training scheme for the baselines as reported in their original thesiss [60, 116, 102, 112]. In all experiments, we use the standard data augmentation of horizontal random flipping and 32×32 random cropping after padding 4 pixels around images. Following the recent works, we also adopt the augmentation policy from [32].

For the results in Table 5.2 and 5.3, the algorithm is applied for 2 rounds with 20 epochs each. For post-training iteration, we set the initial learning rate as same as the one used in earlier pre-training, and drop it after 5 epochs. For cross-entropy (CE) loss, the learning rate at start is set to 0.1 and is decreased by a factor of 0.1 after the 5th and 15th epoch. By increasing the learning rate high at the first epoch in each retraining

iteration, we can encourage the network to explore a newly updated dataset and form a new classifier. We apply RoG and Influential Rank to the models from the last epoch. Influential Rank and RoG both use 500 validation samples. Experiments are conducted with three different noise realizations and the averaged test accuracies are reported.

WebVision

For WebVision 1.0, we use inception-resnet v2 [155] following [20]. For fair comparison with other baselines, both networks are trained for 80 epochs first, and then post-trained with Influential Rank for 20 epochs. We train the network with CE loss for 80 epochs using the SGD optimizer (momentum=0.9) with an initial learning rate 0.01, which is divided by 10 after 50 epochs. When training with DivideMix [102], we follow the setting in their original thesis. After 1 round of Influential Rank, 5K and 6K highly influential examples are removed in CE and DivideMix, respectively. When post-training, both networks are trained for 20 epochs with a learning rate 0.01, and the learning rate is dropped to 0.01 after 10 epochs.

Clothing1M

For Clothing1M, the network is initially trained for 80 epochs with learning rate 0.002 which is decreased by a factor of 0.1 after 40 epochs. We set a batch size to 64, and train the network using SGD optimizer (momentum=0.9) with CE. When training with DivideMix [102], we follow the setting in their original thesis. After 1 round of Influential Rank, 140K and 230K highly influential examples are removed in CE and DivideMix, respectively. For post-training with CE, the model is trained with a learning rate of 0.002 for 10 epochs and then the learning rate is dropped to 0.0002. For post-training with DivideMix, the model is trained with a learning rate of 0.0002 for 10 epochs and then the learning rate is dropped to 0.00002.

Calculation of Hessian

We calculate the Hessian matrix using only sampled n ($\ll N$) data to reduce the computation cost, which is a reasonable approximation by the law of large numbers when the volume of training data is large. For deep neural networks (DNNs), the Hessian matrix could not be positive definite, so we added a positive constant 0.01 to the diagonal following [89]. To efficiently calculate the inverse of the Hessian matrix, we also adopt the conjugate gradient method from optimization theory. The conjugate methods do not require explicitly computing the inverse of the hessian, thus computational complexity is only $O(np)$, where p is the number of parameters of the last fully connected layer. In most cases, we simply use open library to calculate the inverse of the Hessian because the number of parameters is sufficiently reduced and many open libraries, (e.g., NumPy), provide optimized solutions.

5.3.2 Robustness Comparison

Synthetic Label Noise

We conduct experiments on CIFAR dataset with different levels of symmetric noise, $\varepsilon \in \{20\%, 50\%, 70\%\}$. The overall classification (test) accuracies are provided in Table 5.2. The results show that Influential Rank consistently improves the performance of all LNL methods when combined. Also, it is noticeable that applying to a standard cross-entropy (CE) method shows the performance better than or comparable to VolMinNet. These results demonstrate that our post-processing of removing influential examples is effective under varying levels of label noise. Meanwhile, RoG shows inconsistent gains and fails to improve performance of some baselines like DivideMix and UNICON, which is attributed to the assumption of multivariate Gaussian distribution in feature representations. While we terminate the algorithm after the 2nd round, we show the results on more multiple rounds, and the noisy label detection results in 5.3.7.

Table 5.2: **Comparison on CIFAR with varying levels of symmetric label noises.** The averaged test accuracy (%) with LNL methods and their combination with RoG and Influential Rank. The mean accuracy is computed over three different noise realizations.

Dataset	Method	Symm-20			Symm-50			Symm-70		
		Original	+ROG [96]	+Inf. Rank	Original	+ROG [96]	+Inf. Rank	Original	+ROG [96]	+Inf. Rank
CIFAR-10	CE	80.46 (+0.0)	86.97 (+6.51)	**91.08 (+10.62)**	48.84 (+0.0)	62.59 (+13.76)	**84.19 (+35.36)**	28.42 (+0.0)	44.92 (+16.50)	**70.59 (+42.17)**
	VolMinNet [112]	88.26 (+0.0)	88.49 (+0.23)	**91.89 (+3.63)**	71.13 (+0.0)	72.65 (+1.52)	**83.63 (+12.50)**	33.69 (+0.0)	42.08 (+8.40)	**66.07 (+32.39)**
	Co-teaching [60]	91.85 (+0.0)	90.22 (-1.62)	**93.10 (+1.25)**	85.44 (+0.0)	81.96 (-3.48)	**87.30 (+1.86)**	52.63 (+0.0)	53.93 (+1.30)	**60.95 (+8.33)**
	ELR [116]	91.88 (+0.0)	91.50 (-0.39)	**93.04 (+1.15)**	88.48 (+0.0)	87.62 (-0.86)	**89.60 (+1.12)**	77.26 (+0.0)	72.90 (-4.36)	**80.13 (+2.86)**
	ELR+ [116]	93.75 (+0.0)	93.00 (-0.75)	**94.73 (+0.98)**	92.05 (+0.0)	91.11 (-0.94)	**92.79 (+0.74)**	86.94 (+0.0)	83.73 (-3.21)	**88.21 (+1.27)**
	DivideMix [102]	95.64 (+0.0)	95.08 (-0.56)	**96.13 (+0.49)**	94.02 (+0.0)	93.50 (-0.53)	**94.83 (+0.80)**	91.27 (+0.0)	88.69 (-2.58)	**92.42 (+1.14)**
	UNICON [85]	91.95 (+0.0)	91.27 (-0.68)	**94.98 (+3.02)**	93.59 (+0.0)	92.38 (-1.22)	**95.05 (+0.09)**	91.44 (+0.0)	89.38 (-2.06)	**93.12 (+1.68)**
CIFAR-100	CE	64.35 (+0.0)	68.21 (+3.86)	**70.14 (+5.79)**	39.43 (+0.0)	56.94 (+17.51)	**59.31 (+19.88)**	15.50 (+0.0)	39.03 (+23.53)	**40.42 (+24.91)**
	VolMinNet [112]	65.11 (+0.0)	64.93 (-0.18)	**70.05 (+4.94)**	48.77 (+0.0)	53.91 (+5.14)	**58.41 (+9.64)**	28.64 (+0.0)	37.02 (+8.38)	**40.48 (+11.84)**
	Co-teaching [60]	70.85 (+0.0)	66.93 (-3.93)	**72.73 (+1.87)**	59.14 (+0.0)	56.42 (-2.72)	**61.29 (+2.16)**	35.40 (+0.0)	35.97 (+0.57)	**38.29 (+2.89)**
	ELR [116]	72.58 (+0.0)	70.14 (-2.44)	**74.23 (+1.66)**	64.01 (+0.0)	62.91 (-1.10)	**64.43 (+0.42)**	38.78 (+0.0)	**42.07 (+3.29)**	40.07 (+1.29)
	ELR+ [116]	74.15 (+0.0)	70.29 (-3.86)	**75.45 (+1.30)**	65.66 (+0.0)	65.65 (-0.01)	**68.74 (+3.08)**	50.19 (+0.0)	54.48 (+4.29)	**56.53 (+6.34)**
	DivideMix [102]	76.57 (+0.0)	72.29 (-4.28)	**78.63 (+2.06)**	72.29 (+0.0)	68.88 (-3.41)	**74.39 (+2.10)**	62.43 (+0.0)	58.73 (-3.69)	**65.41 (+2.98)**
	UNICON [85]	74.82 (+0.0)	69.84 (-4.98)	**79.61 (+4.79)**	73.96 (+0.0)	68.64 (-5.32)	**75.70 (+1.74)**	68.61 (+0.0)	63.22 (-5.39)	**69.51 (+0.90)**

Real-world Label Noise

CIFAR-10/100N. We further conduct experiment on real-world noisy CIFAR-N in Table 5.3. Although real-world noise is more challenging than a synthetic one, a similar trend in synthetic noisy CIFAR has been observed in real-world noisy CIFAR; the performance gain from Influential Rank is prone to increase with the increase in the

Table 5.3: **Comparison on CIFAR-N with varying levels of real-world label noise.** The averaged test accuracy (%) with LNL methods and their combination with RoG and Influential Rank. The mean accuracy is computed over three different noise realizations.

Method	CIFAR-10N									CIFAR-100N		
	Aggregate ($\varepsilon \approx 9\%$)			Random1 ($\varepsilon \approx 18\%$)			Worst ($\varepsilon \approx 40\%$)			Noisy ($\varepsilon \approx 40\%$)		
		+ROG [96]	+Inf. Rank		+ROG [96]	+Inf. Rank		+ROG [96]	+Inf. Rank		+ROG [96]	+Inf. Rank
CE	89.81	90.19	**91.85**	83.80	85.10	**90.05**	64.86	69.61	**83.73**	54.71	59.64	**62.32**
	(+0.0)	(+0.38)	**(+2.05)**	(+0.0)	(+1.30)	**(+6.25)**	(+0.0)	(+4.76)	**(+18.87)**	(+0.0)	(+4.93)	**(+7.61)**
VolMinNet	88.59	88.93	**91.61**	85.37	85.94	**90.42**	72.35	73.88	**81.51**	54.32	56.94	**59.55**
	(+0.0)	(+0.35)	**(+3.02)**	(+0.0)	(+0.57)	**(+5.05)**	(+0.0)	(+1.53)	**(+9.16)**	(+0.0)	(+2.62)	**(+5.23)**
Coteaching	92.79	91.64	**93.48**	91.59	90.41	**92.54**	84.30	83.10	**86.24**	61.07	58.20	**62.75**
	(+0.0)	(-1.16)	**(+0.69)**	(+0.0)	(-1.18)	**(+0.95)**	(+0.0)	(-1.20)	**(+1.93)**	(+0.0)	(-2.87)	**(+1.68)**
ELR	92.09	91.66	**93.03**	91.59	90.97	**92.41**	86.07	85.48	**87.42**	62.72	62.56	**64.65**
	(+0.0)	(-0.43)	**(+0.94)**	(+0.0)	(-0.62)	**(+0.82)**	(+0.0)	(-0.60)	**(+1.34)**	(+0.0)	(-0.16)	**(+1.94)**
ELR+	94.36	93.35	**94.61**	93.60	92.53	**94.26**	89.74	88.59	**90.54**	63.20	63.26	**64.89**
	(+0.0)	(-1.02)	**(+0.24)**	(+0.0)	(-1.07)	**(+0.66)**	(+0.0)	(-1.15)	**(+0.80)**	(+0.0)	(+0.06)	**(+1.69)**
DivideMix	94.99	94.34	**95.46**	94.90	94.05	**95.52**	92.24	90.14	**93.47**	69.29	65.39	**70.86**
	(+0.0)	(-0.66)	**(+0.46)**	(+0.0)	(-0.84)	**(+0.63)**	(+0.0)	(-2.09)	**(+1.23)**	(+0.0)	(-3.90)	**(+1.57)**
UNICON	90.82	90.10	**93.90**	91.87	90.71	**94.22**	92.33	90.61	**93.96**	68.33	63.47	**71.04**
	(+0.0)	(-0.72)	**(+3.08)**	(+0.0)	(-1.15)	**(+2.35)**	(+0.0)	(-1.71)	**(+1.63)**	(+0.0)	(-4.87)	**(+2.70)**

noise ratio, while RoG rather decreases test accuracy in many cases.

Webvision. From Table 5.4, when combining Influential Rank with the state-of-the-art robust approach, DivideMix, it achieves the best performance. The top-1 accuracy of 76.24% of DivideMix is further increased to 77.88%. In addition, it is noteworthy that our post-processing with the basic method CE shows superior performance to other complex LNL methods, such as Co-teaching and Iterative-CV.

Clothing1M. In Table 5.5, we compare the classification accuracy of Influential Rank with various state-of-the-art methods. Post-processing with Influential Rank to the basic training with CE loss improves the performance with a significant gap, outperforming many recent baselines. Also, applying Influential Rank to DivideMix outperforms the state-of-the-art methods. It is noteworthy that just increasing the number of training epochs cannot bring the meaningful improvement (*i.e.* DivideMix* (longer)).

Table 5.4: **Comparison on WebVision with real-world label noise of** 20%**.** The top-1 top-5 test accuracy. The results are taken from [102] and [116]. $*$ is re-trained in our experimental setup using the official code for post-training.

Method	WebVision		ILSVRC12	
	Top-1	Top-5	Top-1	Top-5
MentorNet [81]	63.00	81.40	57.80	79.92
Co-teaching [60]	63.58	85.20	61.48	84.70
Iterative-CV [20]	65.24	85.34	61.60	84.98
ELR [116]	76.26	91.26	68.71	87.84
ELR+ [116]	77.78	<u>91.68</u>	70.29	89.76
DivideMix [102] (reported)	77.32	91.64	75.20	90.84
DivideMix [102]* (reproduced)	76.24	91.40	73.44	91.60
UNICON [85]	<u>77.60</u>	**93.44**	**75.29**	**93.72**
CE + Influential Rank	72.64	89.20	69.40	90.60
DivideMix* + Influential Rank	**77.88**	91.56	<u>75.28</u>	<u>92.52</u>

While UNICON shows the superior performance, they train much longer hours with 350 epochs. Also, we believe that further performance improvement can be obtained if Influential Rank is applied for multiple rounds.

5.3.3 Comparison with Small-loss Removal

In this section, we show that Influential Rank can be more effective for post-training the pre-trained model than using 'small loss' tricks, which existing methods rely on.

First, we quantitatively show our overfitting scores are superior to the small-loss trick for post-training. Specifically, the loss of each example is used instead of the overfitting scores in Eqs. (5.3) and (5.4) for removing mislabeled examples. Hence, we design a modified version we call 'CE + Small-loss', which excludes high-loss examples following our proposed post-training pipeline. Table 5.6 compares Influential

Table 5.5: **Comparison with state-of-the-art methods in test accuracy(%) on Clothing1M** Results for baselines are copied from original papers, and $*$ are reproduced by the official code.

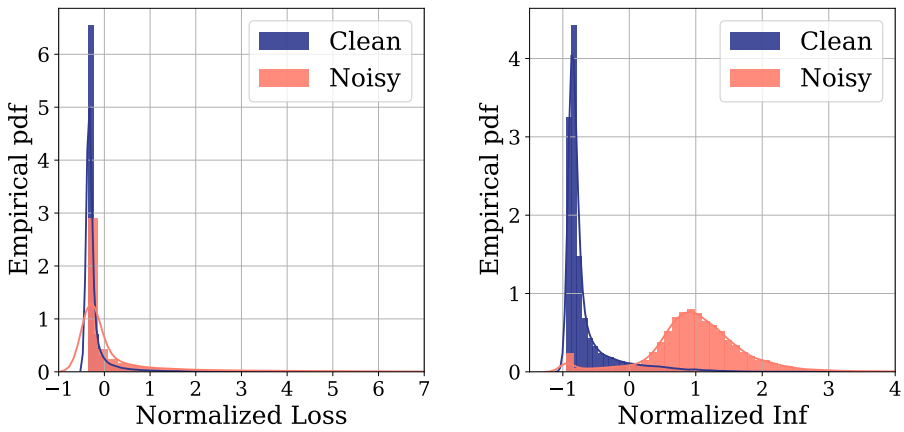
Method	Test Accuracy
Cross-Entropy	69.21
Joint-Optim [158]	72.16
VolMinNet [112]	72.42
Meta-Cleaner [188]	72.50
ELR [116]	72.87
ELR+ [116]	74.81
Meta-Learning [103]	73.47
P-correction [175]	73.49
DivideMix (reported) [102]	74.76
DivideMix* (reproduced) [102]	74.23
DivideMix* (longer) [102]	74.42
UNICON [85]	**74.98**
CE + Ours	72.80
DivideMix* + Ours	<u>74.90</u>

Rank with the modified version of robust post-training on CIFAR-10 with synthetic and real-world label noise. It is observed that the Influential Rank provides a much larger improvement compared to loss-based removal.

Next, Figure 5.4 compares the distribution of the normalized loss and OSM of training samples on the pretrained model with DivideMix. Since training losses are distributed close to 0, it is difficult to classify clean and mislabeled samples with losses after training is done. However, we argue that OSM can provide a new perspective to identify 'confusing' examples with incorrect labels.

Table 5.6: **Comparison with post-training using the small-loss trick on CIFAR-10 with synthetic and real-world noise.** We report the best test accuracy (%).

Method	CIFAR-10 (Symm-70)	CIFAR-10N (Worst)
CE	29.91	63.94
CE + Small-loss	53.43	76.16
CE + Inf. Rank	**75.98**	**84.27**



(a) Training Loss. (b) OSM.

Figure 5.4: **Loss and OSM distribution for all noisy training examples** after training CIFAR-10 with symmetric noise of 40%.

Table 5.7: **Mean test accuracy of training with longer epochs ('+ Longer')** on CIFAR-10 with synthetic and real-world label noise.

	CIFAR-10 (Symm-70)			CIFAR-10N (Worst)		
	Original	+Longer	+Inf.Rank	Original	+Longer	+Inf.Rank
CE	28.42	29.60	**70.59**	64.86	66.92	**83.73**
VolMinNet [112]	33.69	35.09	**66.07**	72.35	72.81	**81.51**
Coteaching [60]	52.63	53.51	**60.95**	84.30	84.83	**86.24**
ELR [116]	77.26	77.83	**80.13**	86.07	86.18	**87.42**
ELR+ [116]	86.94	87.59	**88.21**	89.74	00.00	**90.54**
DivideMix [102]	91.27	92.00	**92.42**	92.24	92.46	**93.47**
UNICON [85]	91.44	92.28	**93.12**	92.33	93.18	**93.96**

5.3.4 Training with Longer Epochs

It is of interest to see whether or not the performance improvement comes from additional training epochs used for post-training, though it is reasonably shorter than the total number of epochs used for pre-training. Table 5.7 shows the performance of the existing state-of-the-art robust methods when training the model with longer epochs, where the number of post-training epochs (*i.e.*, 40) is added to the original epochs (see the columns marked with '+Longer'). In general, the performance of the robust methods remains similarly even with longer training epochs. Therefore, our post-training approach is more desirable than simply increasing the training epochs.

5.3.5 Validity of OSD.

To show the validity of OSD, we investigate the distribution of the $\mathcal{I}_D(x_i, x_t^v; \hat{\theta})$ on real-world images. We use 1,000 'dog' and 'fish' images from ImageNet [141], where 20% labels are randomly flipped. After training the model on this noisy dataset, we calculate OSD using 80 clean validation samples. The OSD distribution is illustrated in Figure 5.5 The horizontal axis is the index of the training data, and the vertical axis is

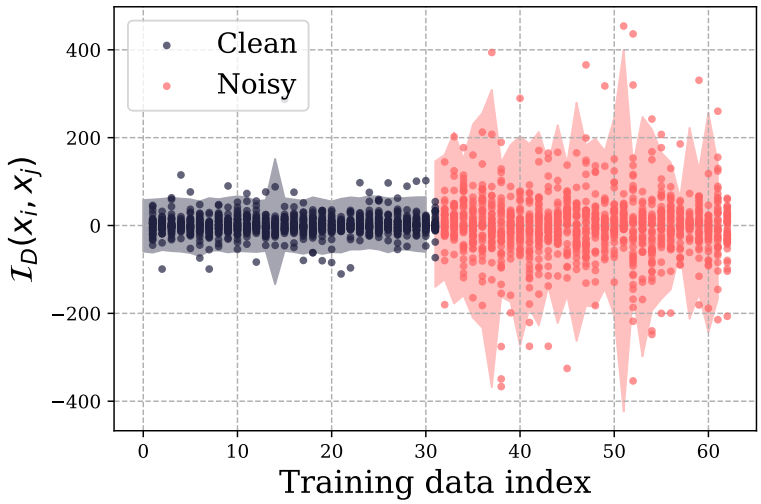
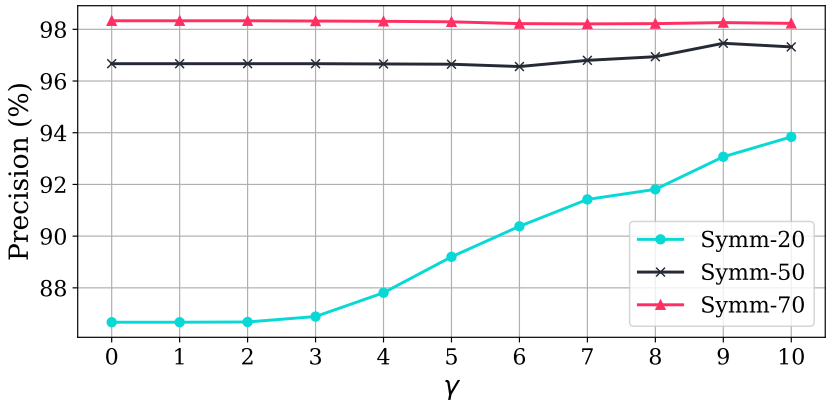


Figure 5.5: **OSD distribution of training samples on validation samples.** Shaded areas show the variance of $\mathcal{I}_D$s of each training sample. The difference in variance between the clean and noisy sets is clearly distinguished.

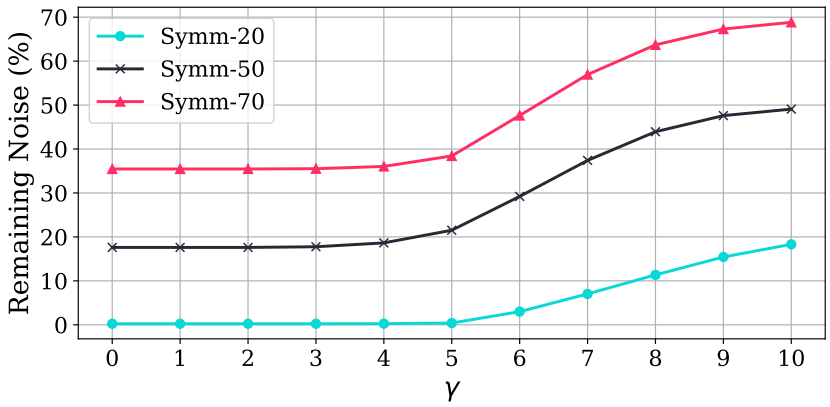
OSD of a training sample x_i on a validation sample x_t^v, i.e., $\mathcal{I}_D(x_i, x_t^v; \hat{\theta})$. We measure OSD on 40 validation samples for each training sample. As illustrated in Figure 5.5, the variation of the influence of a noisy training sample is much larger than a clean training sample. It verifies our intuition that the mislabeled samples exert much more inconsistent influences on validation data than the clean samples do. Therefore, the variance of influences, $\sigma_k(\mathcal{I}_D(x_i, x^v; \hat{\theta}))$ in Eq. (5.4) can be used to find the mislabeled samples. This distribution appears consistently in other categories.

5.3.6 Effects of hyperparameter.

To analyze the effects of the hyperparameter γ, we experiment with different values of γ on CIFAR-10 trained with DivideMix. Choosing a high γ increases the precision of the detected noisy label since it means that a training point exerts inconsistent influences to many classes (Figure 5.6). On the other hand, to meet the high standard (*e.g.,* unanimous consensus among all classes), it cannot but select less noisy samples, which results in the ratio of the remaining noisy labels to be high. Therefore, choosing



(a) Precision.

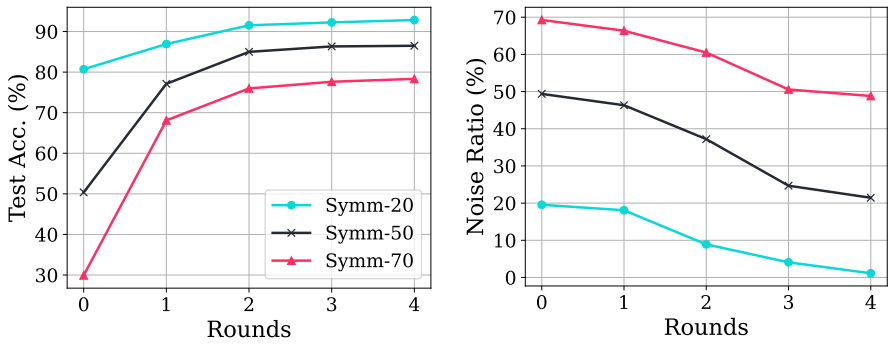


(b) Remaining Noise.

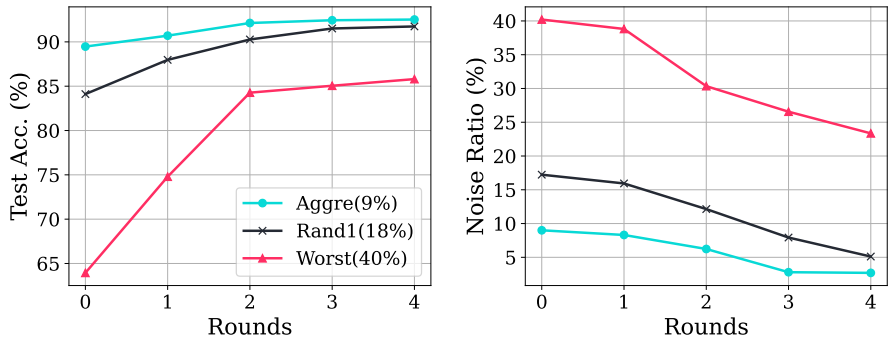
Figure 5.6: **Effects of γ.** Influential Rank is applied to the model trained on CIFAR-10 with DivideMix.

γ is a tradeoff between the more accurate detection and the faster cleansing. Therefore, in our experiments, the gamma is set to 5 in order to fix the data faster when the noise ratio is more than 40%, and set to 8 in the other cases.

Furthermore, setting $\gamma = 0$ is equivalent to using only OSM in the algorithm. Hence, it is verified that OSD helps to increase the precision of noisy label detection. Therefore, choosing γ is a tradeoff between the more accurate detection and the faster



(a) CIFAR-10 with Synthetic Noise.



(b) CIFAR-10N with Real-world Noise.

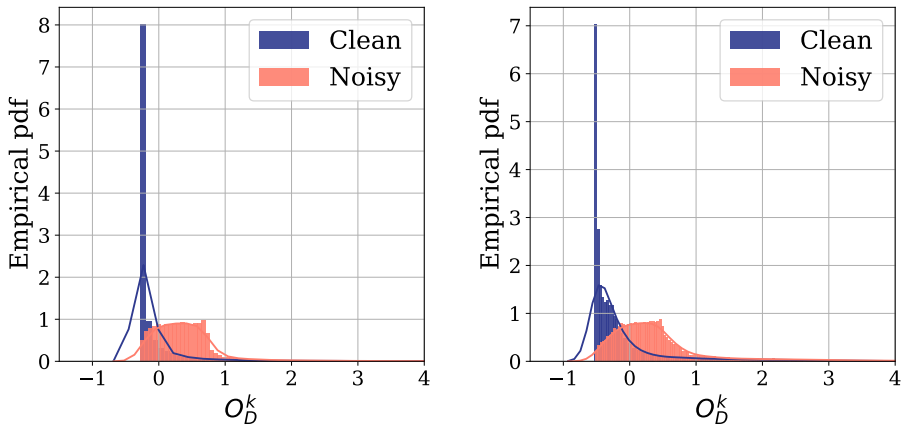
Figure 5.7: **Effect of multi-round post-training on CIFAR-10 with synthetic label noise and real-world.** (Left: Test accuracy over rounds by Influential Rank over rounds, Right: Noise ratio of the refined data.)

cleansing.

5.3.7 Effects of Multi-round Post-training

To verify the potential benefit of using multi-round post-training, we set the number of total rounds to 4, and post-train the network, which is pre-trained using the plain CE. Figure 5.7 depicts the effect of the multi-round post-training on CIFAR-10 and CIFAR-10N, where the round 0 means the model before any post-training. Overall, the noise ratio of the refined data by Influential Rank reduces gradually as the round goes up. In

CIFAR-10 of Figure 5.7(a), the test accuracy is largely improved to 92.83%, 86.49%, and 78.34% from the initial accuracy of 80.71%, 50.37%, 29.91%, respectively. In addition, the initial noise ratios of 20%, 50%, and 70% become 1.12%, 21.43%, and 48.81% at the final round of post-training. Consistently, this improvement trend is exactly the same in CIFAR-10N with real-world noise in Figure 5.7(b). Particularly, the improvement in noise ratio and test error becomes larger when data is corrupted with heavier noise. While performance increase can be expected with multi-rounds, we discover that setting only 2-3 rounds can be sufficiently beneficial in terms of increasing computational burdens.



(a) CE (b) DivideMix

Figure 5.8: **OSD distribution for all noisy training examples** after training CIFAR-10 with symmetric noise of 50%.

5.3.8 Distribution of OSD

To find noisy candidates, we fit a two-modality Gaussian mixture model (GMM) to $\mathcal{O}_D^k(x_i;\hat{\theta})$ for k-th class. To justify if GMM can detect noisy candidates, we plot the distribution of training samples' $\mathcal{O}_D^k(x_i;\hat{\theta})$ (*i.e.,* 6th class) in Figure 5.8. We calculate

OSD from two models trained on CIFAR-10 (Symm-50) with CE and DivideMix, respectively. As shown in Figure 5.8, OSD of clean and noisy samples is bi-modal and separable. Thus, we fit the two-modality GMM into the OSD of all training examples to choose noisy candidates in the proposed algorithm. This observation is consistent even when the model is trained with the existing robust methods.

5.3.9 Noisy Label Detection with Influential Rank

We report the noise ratio change after applying Influential Rank (2 rounds) on CIFAR and CIFAR-N in Table 5.8 and 5.9. As can be seen from the tables, the original noise ratio has been largely alleviated after applying Influential Rank. As can be seen from Section 5.3.7, applying Influential Rank for more rounds can further alleviate the noise ratio in datasets.

Table 5.8: **Averaged noise ratio (%) after Influential Rank (2 rounds). (CIFAR with symmetric noise)**

	CIFAR-10			CIFAR-100		
	Symm-20	Symm-50	Symm-70	Symm-20	Symm-50	Symm-70
No Post-processing	20	50	70	20	50	70
CE	8.83	37.01	62.71	12.15	45.11	64.67
VolMinNet [112]	4.42	36.68	61.02	5.53	31.46	55.15
Co-teaching [60]	3.99	35.26	64.95	6.63	31.81	59.40
ELR [116]	5.51	39.99	63.73	6.52	30.34	60.48
DivideMix [102]	3.84	22.79	42.11	7.18	26.53	49.81
UNICON [85]	4.34	29.36	54.72	5.26	30.43	55.31

Furthermore, we present the noisy label detection precision in Table 5.10 and 5.11. We can observe that mislabeled samples are detected with high precision on both symmetric and real-world noisy data.

Table 5.9: **Averaged noise ratio (%) after Influential Rank (2 rounds). (CIFAR-N)**

	CIFAR-10N			CIFAR-100N
	Aggregate	Random1	Worst	Noisy
No Post-processing	9	18	40	40
CE	5.30	11.07	29.79	34.20
VolMinNet [112]	2.48	4.54	38.61	31.24
Co-teaching [60]	2.32	4.33	26.55	28.56
ELR [116]	1.05	3.26	26.51	27.67
DivideMix [102]	1.11	2.98	18.32	27.28
UNICON [85]	1.55	6.52	23.63	25.23

Table 5.10: **Averaged precision (%) of noise detection after Influential Rank (2 rounds).(CIFAR with symmetric noise)**

	CIFAR-10			CIFAR-100		
	Symm-20	Symm-50	Symm-70	Symm-20	Symm-50	Symm-70
CE	82.84	92.23	93.36	62.38	73.35	86.34
VolMinNet [112]	91.56	99.79	94.01	94.80	98.63	97.13
Co-teaching [60]	96.41	99.90	88.82	94.37	98.52	92.42
ELR [116]	96.37	99.55	99.58	83.14	92.48	92.76
DivideMix [102]	93.91	96.97	98.69	74.56	91.95	96.06
UNICON [85]	86.88	97.96	99.44	85.35	97.47	97.64

5.3.10 Experimental results after one-round

In this section, we present the results after one round of Influential Rank on CIFAR in Table 5.12, and on CIFAR-N in Table 5.13. We can observe that applying only one round of Influential Rank can considerably improve the classification accuracy. Thus, when time budget is limited, applying Influential Rank for once can be sufficient.

Table 5.11: **Averaged precision (%) of noise detection after Influential Rank (2 rounds). (CIFAR-N)**

	CIFAR-10N			CIFAR-100N
	Aggregate	Random1	Worst	Noisy
CE	59.48	85.07	90.19	74.32
VolMinNet [112]	61.35	74.33	98.96	89.38
Co-teaching [60]	64.46	88.49	98.89	89.71
ELR [116]	65.31	91.48	98.92	89.29
DivideMix [102]	70.19	90.68	95.51	86.55
UNICON [85]	54.77	80.11	96.51	85.54

Table 5.12: **Comparison on CIFAR with varying levels of label noises (1 round).** The averaged test accuracy (%) with LNL methods and their combination with Influential Rank. The mean accuracy is computed over three different noise realizations.

Method	CIFAR-10						CIFAR-100					
	Symm-20		Symm-50		Symm-70		Symm-20		Symm-50		Symm-70	
	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank
CE	80.46 (+0.0)	**87.46 (+7.00)**	48.84 (+0.0)	**78.14 (+29.31)**	28.42 (+0.0)	**65.33 (+36.91)**	64.35 (+0.0)	**67.20 (+2.85)**	39.43 (+0.0)	**47.36 (+7.93)**	15.50 (+0.0)	**25.26 (+9.76)**
VolMinNet [112]	88.26 (+0.0)	**90.90 (+2.64)**	71.13 (+0.0)	**82.05 (+10.92)**	33.69 (+0.0)	**63.50 (+29.82)**	65.11 (+0.0)	**68.48 (+3.37)**	48.77 (+0.0)	**56.15 (+7.38)**	28.64 (+0.0)	**36.86 (+8.22)**
Co-teaching [60]	91.85 (+0.0)	**92.77 (+0.92)**	85.44 (+0.0)	**87.04 (+1.61)**	52.63 (+0.0)	**56.92 (+4.30)**	70.85 (+0.0)	**71.42 (+0.56)**	59.14 (+0.0)	**61.01 (+1.87)**	35.78 (+0.0)	**37.56 (+2.16)**
ELR [116]	91.88 (+0.0)	**92.52 (+0.64)**	88.48 (+0.0)	**89.13 (+0.65)**	77.26 (+0.0)	**79.20 (+1.94)**	72.58 (+0.0)	**73.41 (+0.83)**	64.01 (+0.0)	**64.36 (+0.36)**	38.78 (+0.0)	**38.89 (+0.11)**
ELR+ [116]	93.75 (+0.0)	**94.07 (+0.32)**	92.05 (+0.0)	**92.40 (+0.35)**	86.94 (+0.0)	**87.56 (+0.62)**	74.15 (+0.0)	**74.93 (+0.78)**	65.66 (+0.0)	**68.52 (+2.86)**	50.19 (+0.0)	**52.55 (+2.36)**
DivideMix [102]	95.64 (+0.0)	**95.96 (+0.32)**	94.02 (+0.0)	**94.61 (+0.59)**	91.27 (+0.0)	**93.28 (+2.01)**	76.57 (+0.0)	**77.83 (+1.25)**	72.29 (+0.0)	**73.49 (+1.20)**	62.43 (+0.0)	**64.43 (+2.00)**
UNICON [85]	91.95 (+0.0)	**94.52 (+2.56)**	93.59 (+0.0)	**94.75 (+1.16)**	91.44 (+0.0)	**92.84 (+1.40)**	74.82 (+0.0)	**79.22 (+4.40)**	73.96 (+0.0)	**75.36 (+1.40)**	68.61 (+0.0)	**69.63 (+1.02)**

5.3.11 Detector for Video Data Cleaning

In this section, we show that the proposed overfitting score can be expanded to detecting mislabeled videos. Data cleaning for real-world video data is gaining significant attention due to the growth in the popularity of video-based tasks [86, 126, 170]. How-

Table 5.13: **Comparison on CIFAR-N with varying levels of real-world noises (1 round).** The averaged test accuracy (%) with robust methods and their combination with RoG and Influential Rank. The mean accuracy is computed over three different noise realizations.

Method	CIFAR-10N						CIFAR-100N	
	Aggre		Rand1		Worst		Noisy	
	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank	Original	+Inf. Rank
CE	89.81	**90.79**	83.80	**87.98**	64.86	**78.56**	54.71	**59.77**
	(+0.0)	**(+0.98)**	(+0.0)	**(+4.18)**	(+0.0)	**(+13.70)**	(+0.0)	**(+5.06)**
VolMinNet [112]	88.59	**90.72**	85.37	**88.95**	72.35	**78.97**	54.32	**56.94**
	(+0.0)	**(+2.14)**	(+0.0)	**(+3.58)**	(+0.0)	**(+6.63)**	(+0.0)	**(+4.36)**
Co-teaching [60]	92.79	**93.28**	91.59	**92.13**	84.30	**86.03**	61.07	**62.36**
	(+0.0)	**(+0.49)**	(+0.0)	**(+0.54)**	(+0.0)	**(+1.72)**	(+0.0)	**(+1.29)**
ELR [116]	92.09	**92.78**	91.59	**92.09**	86.07	**87.21**	62.72	**64.02**
	(+0.0)	**(+0.69)**	(+0.0)	**(+0.50)**	(+0.0)	**(+1.14)**	(+0.0)	**(+1.31)**
ELR+ [116]	94.36	**94.40**	93.60	**93.85**	89.74	**90.39**	63.20	**64.28**
	(+0.0)	**(+0.04)**	(+0.0)	**(+0.25)**	(+0.0)	**(+0.65)**	(+0.0)	**(+1.07)**
DivideMix [102]	94.99	**95.35**	94.90	**95.37**	92.24	**93.24**	69.29	**70.67**
	(+0.0)	**(+0.35)**	(+0.0)	**(+0.47)**	(+0.0)	**(+1.00)**	(+0.0)	**(+1.38)**
UNICON [85]	90.82	**93.49**	91.87	**93.96**	92.33	**93.79**	68.33	**70.68**
	(+0.0)	**(+2.67)**	(+0.0)	**(+2.09)**	(+0.0)	**(+1.46)**	(+0.0)	**(+2.35)**

ever, detecting video clips with incorrect labels are time-consuming for human annotators more than exploring images because it requires to play and watch the video clip one by one; thus, automatic cleaning of video data can help reduce extreme labeling costs. Therefore, we extend our work to video action recognition for data cleaning.

A few seconds of a video consists of a sequence of frames, ranging from tens to hundreds of consecutive images. Therefore, in general, when predicting the action class, frames are sampled and predicted for each frame. Then, the prediction scores of sampled frames are averaged and the action with the highest prediction score is determined as the final action class.

Consider a video action recognition task with n training videos $(v_1, y_1), \cdots, (v_n, y_n)$, where v_i is the ith video and y_i is its label. Let m_i be the number of sampled frames in the ith video, and x_{ij} be the jth frame in the ith video. Then, the empirical risk for the video dataset is given by $R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{m_i}\sum_{j=1}^{m_i}\ell(y_i, f(x_{ij}, \theta)))$, where

$\ell(x_{ij}, \theta)$ is the loss for a frame x_{ij}. Now, when we denote the loss of a video v_i as $\ell(y_i, f(v_i, \theta)) = \frac{1}{m_i} \sum_{j=1}^{m_i} \ell(y_i, f(x_{ij}, \theta))$, the empirical risk can be rewritten as $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(v_i, \theta))$. Given the empirical risk $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(v_i, \theta))$, the fully optimized (overfitted) model parameters $\hat{\theta}$ minimizes the given empirical risk $R(\theta)$ as $\hat{\theta} \stackrel{\text{def}}{=\!=} \text{argmin}_\theta R(\theta)$. Then, a new parameter when removing the video v is derived as $\hat{\theta}_{v,\omega} \stackrel{\text{def}}{=\!=} \text{argmin}_\theta R(\theta) + \omega \ell(y, f(v, \theta))$. Then, we can use equation (5.1) by definition. Therefore, a video in the video action recognition task can be easily mapped to an image in the image classification problem, and we can simply use the equations derived in this thesis for the video dataset.

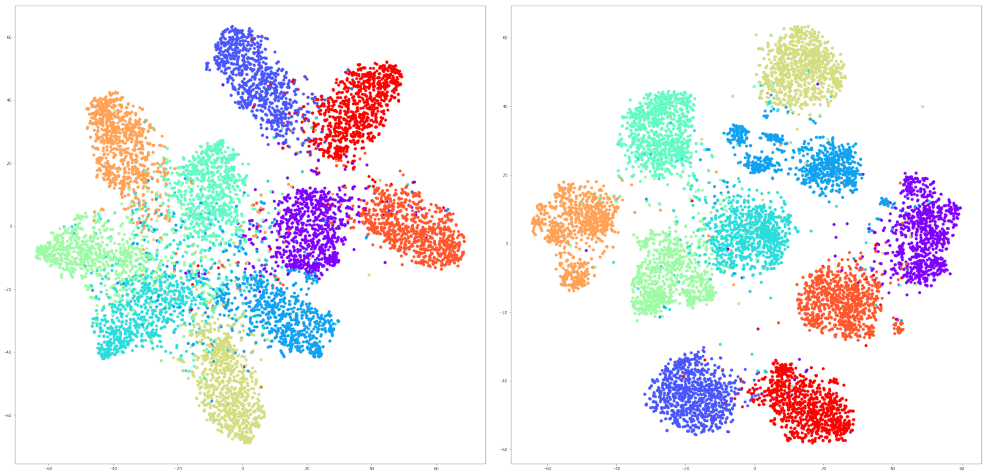
In this thesis, we used Temporal Segment Networks (TSN) [164], which is one of the representative video action recognition models, on HMDB-51 data [93] for action recognition. We train the networks based on public code by Xiong [2], without changing the given hyperparameter settings, except the addition of a hidden layer. We train the network for 300 epochs. We set the initial learning rate to 0.001 and drop it by a factor of 0.1 after 30 and 200 epochs. To deal with both spatial information and long-range temporal structure, TSN adopts two-stream networks that each network processes an RGB image and the stacked optical flows [71], respectively. Therefore, we compute $\mathcal{O}(v_i; \hat{\theta})$ for both networks and analyze the commonly influential video clips from both networks. Since each video clip has multiple scenes, the overfitting score of the clip is computed by averaging the score for randomly sampled scenes in the clip. Then, we filter out mislabeled video clips based on the proposed OSD. We present examples of the detected noisy-label videos in Figure 5.10. Figure 5.10 shows some examples of detected mislabeled video clips by Influential Rank. While HMDB-51 has been known to be clean, surprisingly, we observe that some videos are incorrectly labeled and do not contain any scene corresponding to the label.

[2] https://github.com/yjxiong/tsn-pytorch

5.3.12 Regularizer for Performance Boosting

As another use case, CMO can be considered as a regularizer to avoid overfitting, when there is no apparent label noise in training data. Recently, many regularization techniques have been proposed to reduce the generalization gap of DNNs [151, 183]. Our method post-processes the overfitted decision boundary by squeezing out the negative impact of highly influential examples. Thus, it has the potential to be used as regularization to smooth decision boundaries.

As a case study, we conduct an experiment on clean CIFAR-10 using the same experimental configuration. Table 5.14 and Figure 5.9 shows that Influential Rank can also improve the model trained on clean dataset. We conjecture that this is because Influential Rank removes spurious or isolated data points leading the decision boundary astray, and get a well-generalized decision boundary.



(a) (b) W. Influential Rank.

Figure 5.9: **t-SNE visualization for the learned representation of the trained models.**

Table 5.14: **Result of Influential Rank on clean CIFAR-10.**

	# of training	Accuracy
Original	50,000	94.2
+Inf. Rank	48,989	**96.6**

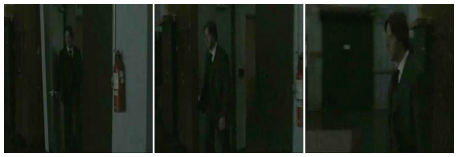
5.4 Summary

We have proposed a post-training method named Influential Rank, which sways the overfitted decision boundary to be correct, in the presence of noisy labels. Unlike the existing methods, Influential Rank starts from an overfitted model and makes the model more robust against noisy labels progressively. We have conducted extensive experiments on real-world and synthetic noisy benchmark datasets. The results demonstrate that Influential Rank consistently provides performance gain when combined with multiple state-of-the-art robust learning methods. In addition, we have shown that Influential Rank performs as a detector for video data cleaning or a regularizer to smooth the decision boundary.

(a) Mislabeled as ‘Run’.

(b) labeled as ‘Jump’.

(c) labeled as ‘Dive’.



(d) labeled as ‘Hit’.

(e) labeled as ‘Dive’. (Jump?)

(f) labeled as ‘Punch’. (Kick?)

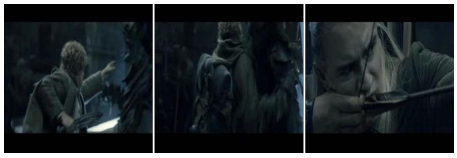
(g) labeled as ‘Punch’. (Hit? Sword?)

(h) labeled as ‘Kick’. (Run?)

(i) labeled as ‘Kick’. (Sword?)

(j) labeled as ‘Kick’. (Ride bike?)

(k) labeled as ‘Kick’. (Turn?)



(l) labeled as ‘Sword’. (Shoot bow?)

Figure 5.10: **Training examples with the highest $\mathcal{O}(\cdot)$ (HMDB-51).** Some videos are incorrectly labeled and do not contain any scene corresponding to the label. The other videos are partly noisy and include scenes corresponding to other labels that seem more suitable. The other possible labels are shown in parentheses. (Best viewed magnified on screen.)

Chapter 6

RoCOCO: Robustness Benchmark of MS-COCO to Stress-test Image-Text Matching Models

6.1 Overview

Understanding the visual world with language is a crucial aspect of artificial intelligence, which has inspired the research of image-text matching. Recent advancements in visual semantic embedding methods [104, 29, 19] and large-scale vision-language pretraining models [137, 187, 100] have significantly improved image-text matching accuracy (i.e., recall@1) on the popular MS-COCO [114] benchmark dataset. However, it is important to question the reliability of these results and their performance in real-world scenarios. Assessing the robustness of trained models in practical applications is crucial, considering their significant impact on various individuals.

Users today actively generate content through platforms like blogs, Instagram, and YouTube, creating vast amounts of data in platform databases, where people can freely search for that content. However, this also opens the door for malicious users to manipulate search results, leading them away from users' intended content. For example, as depicted in Figure 6.1 (a), it is possible to upload images with inserting malicious images, such as pornography or hateful content, into legitimate images. Similarly, by

modifying the semantic details of texts, poisoned text can be prioritized in search results instead of the original text (Figure 6.1 (b)). In scenarios like defense industry applications, the use of such models can pose a significant risk, as innocent civilians may be mistakenly identified as threats.

Based on this motivation, we propose a Robustness benchmark of MS-COCO (RoCOCO) that can stress-test the model by attacking the gallery set. To generate fooling data, we employ two principles. Firstly, we make perceptible changes by altering the meaning of the text and mixing the images that humans can easily detect. We expect robust models to resist such explicit modifications, as they should possess a comprehensive understanding of the overall semantic meaning and visual elements. Secondly, to create challenging text and images, we introduce minimal changes in the embedding outputs. This idea is inspired by the common practice in which models measure similarity between the embedding outputs of image and text encoders [137, 19, 100]. By applying the principles, we construct four text datasets and two image datasets, on which we reevaluate various state-of-the-art methods. Surprisingly, despite the simplicity of the attack, many state-of-the-art models show considerable performance degradation on the proposed benchmarks (e.g., $81.9\% \rightarrow 64.5\%$ in BLIP [100], $66.1\% \rightarrow 37.5\%$ in VSE∞ [19] for Image-to-Text retrieval). These findings highlight the tendency of current image-text retrieval models to overlook subtle details and show more attention to specific words or image parts.

Our key contributions can be summarized as follows:

- We provide various robustness-evaluation benchmarks and discover the significant performance drops across all models regardless of the extent of large-scale pretraining.
- We study vulnerabilities of image-text retrieval models and observe that these models often tend to focus on specific words or image components rather than comprehending the overall context.
- To address the vulnerability, we propose Semantic Contrastive Loss that can learn

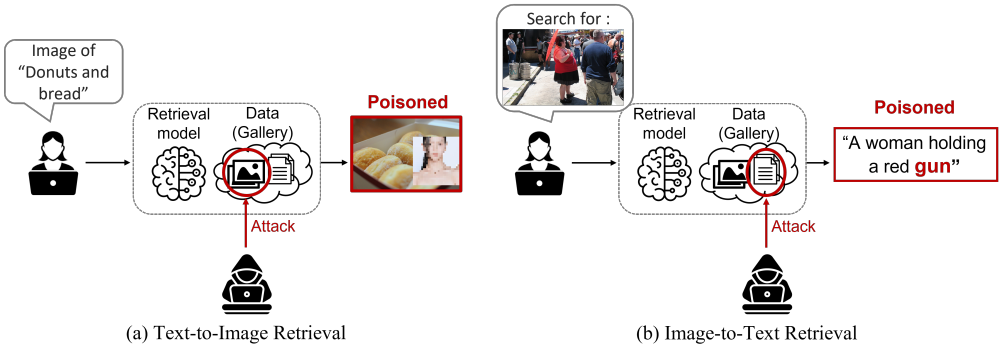


Figure 6.1: **Attack Scenario**. By inserting malicious images and text into the searching pool (gallery), an attacker can induce the model to extract undesired images and text contrary to the user's intentions.

semantic details.

6.2 Robustness-Evaluation Benchmark

6.3 Robustness-Evaluation Benchmark

6.3.1 Observations motivating the proposed approach

Our goal is to quantitatively evaluate how well ITM models understand both text and image. Specifically, we measure the robustness of a ITM model through our proposed benchmark, which assesses how robustly the model retrieves the ground-truth image/caption instead of our newly generated adversarial image/caption.

Based on the examples observed from the BLIP [100] model, we have developed adversarial images and captions that are capable of assessing the model's vulnerability. Figure 6.2 illustrates our observation. Firstly, we generate an adversarial image with noticeable changes by simply inserting an unrelated image into an original image (Figure 6.2 (a)). Surprisingly, even though the adversarial image is easily discernible by humans, we observe that the ITM model often favors a mixture of unintended images

(a) Fooling image

(b) Fooling caption

Figure 6.2: **Illustration of an adversarial image and caption tested with the state-of-the-art BLIP [100]**. When we add a new image created by inserting an unrelated image to the original one, this new image is ranked as top 1 (Text-to-image). Likewise, when we add a new caption with only one word changed from "umbrella" to "gun", this new caption is retrieved as top 1 (Image-to-text).

rather than the desired (ground-truth) ones. As it is easy for anyone to download images from the internet and re-upload images after manipulation, this can be a common and feasible attack scenario.

Likewise, we create an adversarial caption by replacing one word in the caption to alter the meaning of the sentence. For example, replacing "umbrella" with "gun" as in Figure 6.2 (b). Again, we discover that the model often tends to prioritize retrieving the adversarial captions over the ground-truth captions. Therefore, to assess the model's ability for understanding the overall details between the image and text, we introduce adversarial captions to make the image-to-text task more challenging.

6.3.2 Adversarial Image Generation

To generate adversarial images containing undesired content, we employ two techniques for image insertion. One is the Mixup-style approach [182], where two images are blended together in different proportion (Mix). The other method inserts a patch of an undesired (fake) image onto the original image, as in Cutmix [178] (Patch). The undesired (fake) image is randomly selected from the COCO test set. When inserting a fake image x^f into an original image x^o, we use two mixing ratios λ and $\mathbf{M}$ for Mix

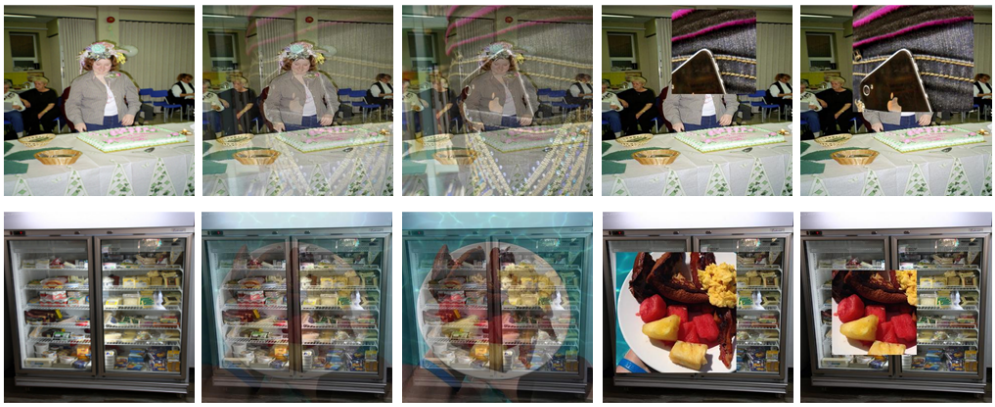


Figure 6.3: **Example of adversarial images with different λ.**

and Patch, respectively, as follows:

$$\text{Mix} : \tilde{x} = \lambda x^o + (1 - \lambda) x^f,$$

$$\text{Patch} : \tilde{x} = \mathbf{M} \odot x^o + (\mathbf{1} - \mathbf{M}) \odot x^f,$$

where $\mathbf{M} \in \{0, 1\}^{W \times H}$ denotes a binary mask indicating a randomly chosen location of the fake patch, where W is the width and H is the height of the image. In Patch, λ is calculated by $\lambda = \frac{\sum_{i,j} \mathbf{M}_{i,j}}{W \times H}$. That means that the portion of 1 in M is adjusted according to λ value. Figure 6.3 shows the examples of created adversarial images. Creating these adversarial images and adding them to the gallery set provides an easy yet effective method to measure the robustness of the model.

6.3.3 Adversarial Caption Generation

Source Word Selection via Embedding-Influence

We create adversarial captions by substituting one word in the original caption with an unrelated word. To introduce discernable changes in the meaning of the caption, we focus nouns for replacement. For effective attacks, we choose words that have minimal impact on the embedding outputs. This idea is inspired by the common practice

in which models measure similarity between the embedding outputs of image and text encoders trained on image-text pairs [137, 19, 100]. We hypothesize that even with considerable changes in the semantic meaning, the model would be confused with the original caption if the embedding outputs change little. We will empirically demonstrate this claim in our experiments.

To estimate the influence of a word, we propose embedding-influence (EI) score. EI sore measures the change in embedding when the word is removed from the caption. Given a text encoder f_T, and a caption $C = \{c_m \mid m = 1, \cdots, M\}$, where M is the number of words in C, the embedding-influence (EI) score of a word, c_s, is defined by

$$EI(c_s) = 1 - \frac{< f_T(C), f_T(C \setminus c_s) >}{\parallel f_T(C) \parallel \parallel f_T(C \setminus c_s) \parallel}, \tag{6.1}$$

where $<, >$ denotes the dot(inner) product operation. A low EI score means that the word has little influence on the embedding output of the caption. Given its limited influence on the embeddings compared to other words, substituting this word with a different word is expected to have low impact on the overall embeddings.

Using four representative models (i.e., VSRN [104], CLIP [137], VSE∞ [19], BLIP [100]), we measure the EI score of each word to assess its influence. We select the word with the least influence across the models. If the word is chosen by the majority of models, it is replaced by a target word (see Section 6.3.3). If there are multiple options, we randomly choose one. Interestingly, the words with the lowest embedding influence exhibit little variation across the models. We will provide further details in Section 6.4.3.

Target Word Selection for Diverse Adversarial Caption Dataset

To generate confusing captions covering various scenarios, we need to determine a target word replacing the source word chosen in Section 6.3.3. To this end, we employ four different policies. First, we use concept groups from GRIT benchmark [58], which categorizes nouns from popular datasets including COCO into 24 concept groups such

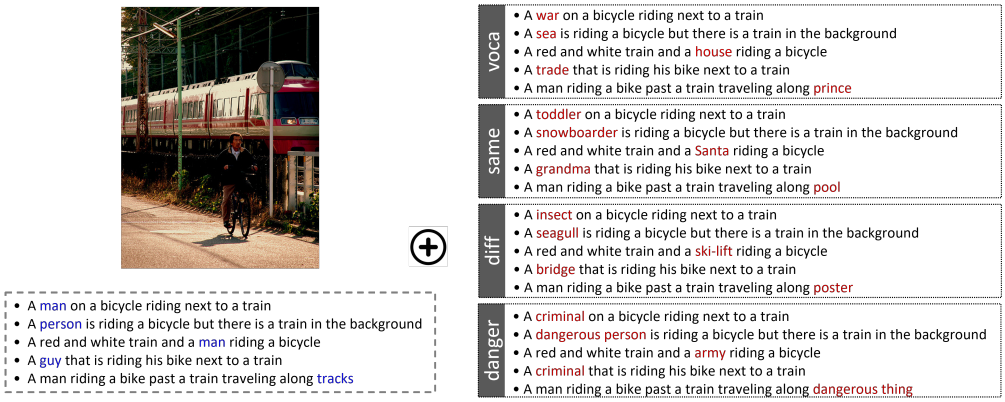


Figure 6.4: **Example of generated captions.** (Left) Original COCO image and captions. (Right) Our generated captions, Rand-voca, Same-concept, Diff-concept, and Danger from top to bottom. The model is to retrieve the most appropriate caption from a pool of both original and newly generated captions. Our assumption is that the robust model should be able to retrieve the original captions well without being confused by new captions with different meanings.

as food, people, and places. We add 7 concept groups for words not covered by GRIT. We include more details in Appendix. We then create **Same-concept** and **Diff-concept** captions by replacing words based on concept groups For example, **Same-concept** replaces "umbrella" with a word in the same concept (i.e., tools), which can be "rope" or "boxes". **Diff-concept** replaces "umbrella" with a word selected randomly from different concepts, such as "pizza" from "food" concept, or "monkey" from "animal" concept.

Next, we employ the BERT [37] vocabulary (**Rand-voca**) to stress-test with a wide range of words. We randomly select words consisting of only English letters, excluding those in other languages or special characters. Additionally, we create a special case (**Danger**) by using words related to public security. This allows us to evaluate the models' ability to comprehend critical situations that could potentially pose a threat to human safety. For instance, we replace "umbrella" with "gun" or "weapon". Examples

of the generated captions can be seen in Figure 6.4.

6.4 Experiments and Results

6.4.1 Experimental setting

In this section, we evaluate the existing image-text matching (ITM) models on our new dataset, RoCOCO. For Image-to-Text retrieval, we expand MS-COCO test data by adding 25,000 newly generated adversarial captions using our approach to the existing 25,000 original captions, creating a gallery of 50,000 captions. We then retrieve text from this expanded gallery. Conversely, for Text-to-Image retrieval, we include 5,000 newly generated adversarial images to the 5,000 original images, resulting in an image gallery of 10,000 images.

Evaluation Metrics Recall@k, especially Recall@1 (R@1), is the most popular metric for evaluating the existing ITM methods. In this paper, we propose two metrics, *Drop Rate* and *Incorrect Recall*@1 (IR@1) in addition to R@1. Drop rate measures the relative decrease in R@1 compared to the evaluation on the original COCO 5K testset. We calculate drop rate as $(\mathrm{R@1} - \mathrm{R_{New}@1})/\mathrm{R@1}$. Incorrect Recall@1 calculates the percentage of newly added adversarial captions/images that are retrieved as top 1. This can quantitatively estimate the vulnerability of a model.

Models for Evaluation We compare 14 state-of-the-art Vision-Language (VL) models, whose trained weights are available to the public. They can be categorized into two groups; large-scale vision-language(VL) pre-training and visual semantic embedding groups. Large-scale VL pre-training group includes CLIP with ViT-B/32, ViT-B/16 and ViT/L14 backbones [137], fine-tuned ALBEF [101], and zero-shot and fine-tuned BLIP with ViT-B and ViT-L backbones [100]. While 'zero-shot' and 'fine-tuned' models are both pre-trained on large-scale datasets, 'zero-shot' refers to not being fine-tuned with COCO train set. Visual semantic embedding group includes models using region features based on bottom-up attention [5] and SCAN [97]: VSRN [104], SAF,

Table 6.1: **Image-to-Text retrieval results.** Models are re-evaluated on four new benchmark datasets: Rand-voca, Same-concept, Diff-concept, and Danger. Recall@1 (R@1)(↑), drop rate(↓), False Recall@1 (FR@1)(↓) are shown. We can see consistent degradation across all vision-language models regardless of using pre-training datasets and different methods. The biggest performance drops are marked in bold.

	COCO 5K	Rand-voca			Same-concept			Diff-concept			Danger		
	R@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1
Large-scale VL pre-training models													
CLIP ViT-B/32 (zero-shot) [137]	50.10	36.44	27.27	34.63	**35.77**	**28.60**	**36.64**	37.48	25.18	32.27	42.18	15.81	19.69
CLIP ViT-B/16 (zero-shot) [137]	52.44	**38.18**	**27.19**	**34.87**	38.36	26.85	34.40	40.23	23.28	30.57	44.67	14.81	18.19
CLIP ViT-L/14 (zero-shot) [137]	56.04	**39.90**	**28.81**	33.95	40.90	27.02	**34.86**	42.66	23.88	24.07	46.48	17.06	30.16
ALBEF [101]	77.58	**60.13**	**22.49**	**26.07**	60.55	21.95	25.09	61.84	20.29	23.75	63.37	18.32	20.43
BLIP ViT-B (zero-shot) [100]	70.54	**35.28**	**49.98**	**54.58**	47.77	32.28	37.45	45.58	35.39	40.89	42.39	39.90	43.99
BLIP ViT-B [100]	81.90	**64.50**	**21.25**	**23.72**	68.74	16.07	18.74	69.20	15.51	17.36	67.81	17.21	18.92
BLIP ViT-L (zero-shot) [100]	73.66	**45.96**	**37.60**	**40.49**	55.38	24.82	28.27	55.69	24.39	27.56	55.93	24.07	26.54
BLIP ViT-L [100]	82.36	**66.84**	**18.85**	**21.18**	71.16	13.60	16.02	72.70	11.72	13.86	72.37	12.13	13.73
Visual Semantic Embedding models													
VSRN [104]	52.66	**42.22**	**19.82**	**22.14**	44.56	15.38	18.06	46.12	12.41	14.47	46.78	11.17	12.77
SAF [39]	55.46	**39.30**	**29.14**	**31.54**	42.04	24.20	28.35	45.00	18.85	22.24	42.77	22.88	26.35
SGR [39]	57.22	**41.69**	**27.14**	**30.43**	43.61	23.79	28.02	46.56	18.63	22.07	44.90	21.53	24.72
VSE∞ (BUTD region) [19]	58.02	**31.71**	**45.34**	**47.99**	39.79	31.42	35.12	36.91	36.38	39.86	37.66	35.09	37.38
VSE∞ (BUTD grid) [19]	59.40	**32.24**	**45.72**	**48.75**	41.12	30.77	33.58	38.71	34.84	38.40	39.71	33.15	35.32
VSE∞ (WSL grid) [19]	66.06	**37.54**	**43.17**	**46.07**	48.76	26.19	29.59	44.86	32.09	35.06	45.39	31.29	33.07

SGR [39], and VSE∞ [19].

6.4.2 Re-evaluation on RoCOCO

Image-to-Text Retrieval

Table 6.1 reports the image-to-text retrieval results on our new datasets. First, we can observe the highest performance degradation on Rand-voca. This can be attributed to the fact that Rand-voca contains numerous unexpected words that are not commonly appear together in captions. In contrast, Same-concept and Diff-concept datasets consist of words belonging to the same COCO dataset. This observation suggests that models are vulnerable to sentences comprising unfamiliar word combinations that rarely appear in the trained captions.

Furthermore, we can observe consistent degradation across all vision-language models, regardless of methods or the scale of pre-training datasets (e.g., 400M im-

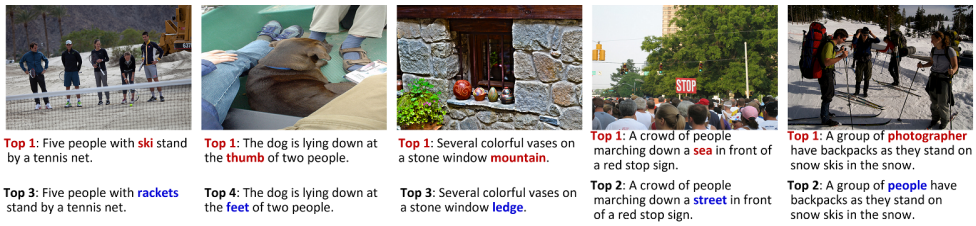
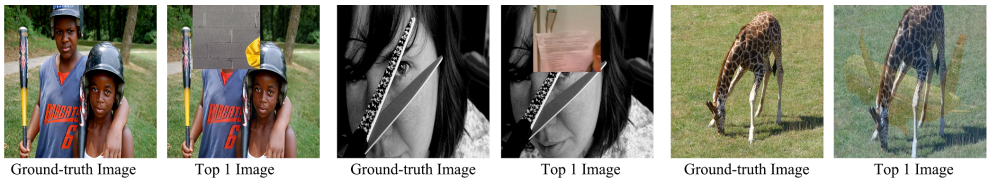


Figure 6.5: **Examples of incorrectly retrieved texts with BLIP from Same-concept (Image-to-Text).** suggest that the model is overlooking the semantic details of the sentence.



(a) Query: Two black children wearing baseball hats and holding bats.

(b) Query: The woman is getting ready to cut her bangs with scissors.

(c) Query: A single young giraffe eats from a grassy field.

Figure 6.6: **Examples of incorrectly retrieved images with BLIP when $\lambda = 0.8$ (Text-to-Image).** The first two examples are from the Patch, while the last one is from the Mix. In the Patch examples, some salient parts are obscured, while in the Mix example, unrelated image of a 'plane' is visible.

age pairs in CLIP [137], 129M in BLIP [100], 14M in ALBEF [101]). We assume that commonly used image-text matching loss might be vulnerable to a single-word change in the caption because the loss is used to minimize the distance between image-text pairs for learning multimodal representations. In addition, Figure 6.5 presents qualitative examples evaluated with BLIP (ViT-B) from Same-concept dataset. Our results highlight the importance of developing a robust training strategy for ITM model that can better capture word-level semantic meaning and align it with images.

Table 6.2: **Text-to-Image retrieval.** Models are evaluated with our new benchmark: Mix and Patch with different λ. Recall@1 (R@1)(↑), drop rate(↓), False Recall@1 (FR@1)(↓) are shown. The results are averaged over image generations with three different random seeds. We can see consistent degradation across all vision-language models.

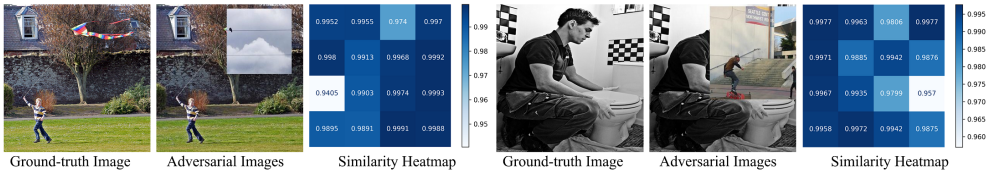
	COCO 5K	Mix ($\lambda = 0.9$)			Mix ($\lambda = 0.8$)			Patch ($\lambda = 0.9$)			Patch ($\lambda = 0.8$)		
	R@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1
Large-scale VL pre-training models													
CLIP ViT-B/32 (zero-shot) [137]	30.14	20.29	32.68	33.55	22.79	24.39	26.03	22.49	25.38	28.63	24.15	19.87	23.69
CLIP ViT-B/16 (zero-shot) [137]	33.03	20.05	39.30	39.00	23.57	28.64	29.88	22.58	31.64	35.18	24.70	25.22	29.41
CLIP ViT-L/14 (zero-shot) [137]	36.14	25.49	29.47	28.99	27.75	23.22	24.29	27.56	23.74	27.64	29.09	19.51	23.97
ALBEF [101]	60.67	44.13	27.27	26.60	48.02	20.85	21.11	48.86	19.47	19.58	51.80	14.62	15.30
BLIP ViT-B (zero-shot) [100]	56.36	39.03	30.75	31.54	43.94	22.04	22.28	41.96	25.55	27.56	45.05	20.07	22.79
BLIP ViT-B [100]	64.31	40.71	36.70	39.93	46.97	26.96	30.84	48.40	24.74	42.57	52.61	18.19	21.45
BLIP ViT-L (zero-shot) [100]	58.18	44.29	23.87	25.13	47.61	18.17	19.96	46.79	19.58	21.07	49.50	14.93	16.50
BLIP ViT-L [100]	65.06	41.87	35.64	42.45	48.92	24.81	33.91	48.55	25.38	29.17	49.50	23.92	22.10
Visual Semantic Embedding models													
VSRN [104]	40.34	27.04	32.97	39.05	31.36	22.26	28.87	30.08	25.43	31.11	32.50	19.43	24.80
SAF [39]	40.11	30.90	22.96	27.84	33.37	16.80	22.87	32.50	18.97	23.78	34.03	15.16	19.69
SGR [39]	40.45	30.71	24.08	28.08	33.41	17.40	22.57	32.40	19.90	23.95	34.08	15.75	19.90
VSE∞ (BUTD region) [19]	42.46	31.57	25.65	30.74	35.61	16.13	20.45	34.17	19.52	23.51	36.48	14.08	17.28
VSE∞ (BUTD grid) [19]	44.07	30.22	31.43	36.68	35.26	19.99	25.00	35.70	18.99	23.52	38.75	12.07	15.82
VSE∞ (WSL grid) [19]	51.55	34.31	33.44	38.60	40.40	21.63	26.26	43.67	15.29	18.39	46.87	9.08	11.31

Text-to-Image Retrieval

We evaluate VL methods on new image set with $\lambda = 0.9, 0.8$ in Table 6.2. The images are generated using three random seeds, and the averaged results are reported. It can be also observed that all VL methods consistently exhibit degradation in performance. In addition, in Figure 6.6, we present examples of incorrect image retrievals using BLIP (ViT-B) when λ is set to 0.8. While humans would not prefer the mixed images to the original images, we observe that the models easily confuse the two images. We argue that this evaluation is simple yet effective for assessing the robustness of the models. More results with different λ values can be found in the Appendix.

6.4.3 Analysis and Discussions

The influence of each spatial parts on the embedding varies within a single image. To examine why the model can be deceived by unrelated images, we analyze the



Ground-truth Image Adversarial Images Similarity Heatmap

(a) Query: "A little boy flying his kite in the yard"

Ground-truth Image Adversarial Images Similarity Heatmap

(b) Query: "A young man bending next to a toilet"

Figure 6.7: **The influence of spatial part of the image on the embedding.** Even when specific parts are mixed, the model can confuse two images since other more influential parts remain.

impact of each spatial location in the image on the embedding output. We divide the image into 16 parts and mask each part to zeros, to observe the changes in the embedding. The heatmap in Figure 6.7 shows the cosine similarity between the embedding of the original image and the image embedding when each corresponding part is masked. In the cases where adversarial images are retrieved as top 1, we can observe that influential parts like "boy" or "toilet" still remain despite obscuring some important parts like "kyte" or "a man's face". This finding indicates that certain parts of the image have a stronger impact on the retrieval outcomes than the other parts.

Each word within a caption has a different impact on the embedding. In Section 6.3.3, we introduce the Embedding-Influence (EI) score. Figure 6.8 demonstrates the varying influences of words within each caption, with the red color indicating higher influence. The noun with the highest EI score is underlined in red, while the noun with the lowest score is underlined in gray. Notably, nouns like "umbrella" and "man" have significant meaning but relatively low influence on the embedding outputs. Thus, substituting these words can result in a significant change in semantic meaning without substantially affecting the original embedding.

Manipulating words with low EI scores proves to be an effective approach for adversarial attacks. To demonstrate this, we evaluate model performance by removing words in captions using different methods. The "Random" method randomly removes a noun, while the "Large EI" removes the noun with the highest EI score,

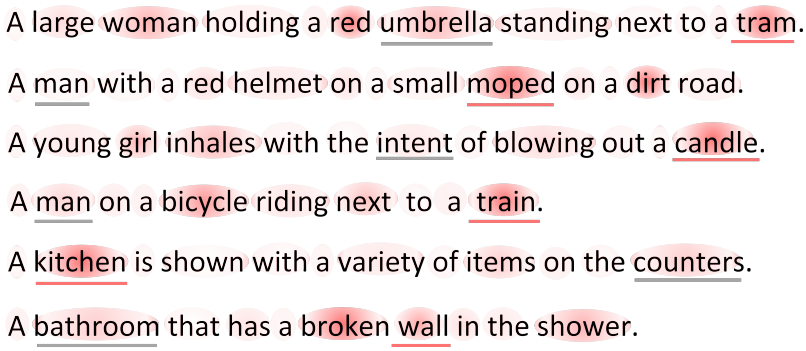


Figure 6.8: **Influence of a word in a caption.** The darker the red color of a word, the greater its influence. For each caption, the noun with the highest EI score is underlined in red, and the noun with the lowest EI score is underlined in gray. We can observe that some semantically important nouns such as 'man' and 'bathroom' have low EI scores, which can make a model not robust.

and vice versa for the "Low EI". We create new captions by simply deleting the source word without replacement to mitigate the impact of the changed word. Table 6.3 shows that deleting words with low EI scores is the most effective approach for fooling the models, while deleting words with high EI scores results in minimal performance degradation. This finding supports our hypothesis that leveraging the influence of words on embedding features can effectively confuse the models. Thus, manipulating words with low EI scores can be a valuable method for assessing the robustness of newly trained models.

Words with the lowest EI scores exhibit little variation across different VL models. Figure **??** displays the level of agreement among models in selecting the word with the lowest EI scores. The x-axis represents the maximum number of agreements among the four models in selecting the word with the lowest EI score, while the y-axis represents the number of captions. Interestingly, in over 70% of cases, two or more models select the same word, despite being trained using different architectures and datasets (e.g., more pre-training data). This highlights a common vulnerability in the

Table 6.3: **Effects of using EI scores.** Deleting a source word with the lowest EI score shows the largest performance drop.

	COCO	Random Deletion			High EI Deletion			Low EI Deletion		
	R@1(↑)	R@1(↑)	drop rate(↓)	FR@1(↓)	R@1(↑)	drop rate(↓)	FR@1(↓)	R@1(↑)	drop rate(↓)	FR@1(↓)
CLIP ViT-B/32 (zero-shot) [137]	50.10	38.58	22.99	29.66	42.76	14.65	21.84	**36.04**	**28.06**	**32.30**
CLIP ViT-L/14 (zero-shot) [137]	56.04	42.54	24.09	30.4	48.58	13.31	20.42	**39.22**	**30.01**	**33.74**
BLIP ViT-B (zero-shot) [100]	70.54	45.58	35.38	40.54	57.14	19.00	25.80	**36.34**	**48.48**	**52.48**
BLIP ViT-B [100]	81.90	65.54	22.46	19.98	72.74	11.18	14.06	**59.28**	**27.62**	**30.10**
VSRN [104]	52.66	44.7	15.12	18.02	43.46	17.47	22.56	**38.56**	**26.78**	**29.36**
VSE∞ (BUTD region) [19]	58.02	34.2	41.05	45.58	40.58	30.06	38.06	**30.02**	**48.26**	**50.72**
VSE∞ (BUTD grid) [19]	59.40	34.3	42.26	46.46	39.92	32.79	39.78	**30.46**	**48.72**	**51.54**
VSE∞ (WSL grid) [19]	66.06	40.8	38.24	41.68	47.32	28.37	33.76	**36.56**	**44.66**	**47.14**

current image-text matching approach, suggesting that attacks can have a universal impact.

VL models can be fooled by highly nonsensical sentences with multiple word replacements. To further investigate the vulnerability of VL models, we conduct experiments where 2 to 5 words are randomly replaced in the captions using words from the Bert vocabulary. Interestingly, the results in Table 6.4 show meaningful performance degradation across the entire model, even when the original semantic meaning is significantly disrupted. Large-scale pretraining methods exhibited better robustness than VSE models when multiple words are changed simultaneously.

Additionally, Figure 6.10 presents top 1 retrieval examples of captions with four word replacements by BLIP (ViT-B). We can observe that the broken captions contain at least one correct keyword, such as “motorcyclist” in the first image. These findings suggest that the model may focus more on specific words rather than considering the entire sentence.

6.4.4 Semantic Contrastive Loss for Adversarial Captions

Throughout our study, we have observed that VL models tend to overlook semantic details. To address this issue, we propose the Semantic Contrastive (SC) Loss, which encourages the model to distinguish between images and text when introducing various changes to the text.

Table 6.4: **Image-to-Text retrieval on dataset with multiple words substitutions.** We generate captions by randomly replacing more words and add to COCO test set. The results are averaged over generations with three different random seeds. Recall@1 (R@1)(↑), drop rate(↓), False Recall@1 (FR@1)(↓) are shown. Models can confuse sentences even when the semantic meaning is more largely damaged.

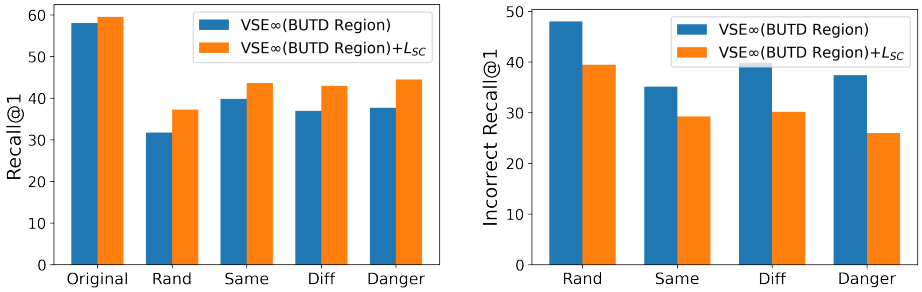
	COCO	2 words substitution			3 words substitution			4 words substitution			5 words substitution		
	R@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1	R@1	drop rate	FR@1
Large-scale VL pre-training models													
CLIP ViT-B/32 (zero-shot) [137]	50.10	42.89	14.39	19.71	46.07	8.04	12.67	47.45	5.29	8.15	48.37	3.45	5.46
CLIP ViT-B/16 (zero-shot) [137]	52.44	45.35	13.52	19.07	48.43	7.65	11.89	49.97	4.71	8.01	50.61	3.49	5.95
CLIP ViT-L/14 (zero-shot) [137]	56.04	47.35	15.51	22.18	50.22	10.39	15.78	51.99	7.23	11.56	53.07	5.30	8.27
ALBEF [101]	77.58	72.43	6.64	2.40	73.03	5.86	0.88	73.23	5.61	0.43	73.26	5.57	0.32
BLIP ViT-B (zero-shot) [100]	70.54	53.04	24.81	30.75	62.99	10.70	14.72	67.95	3.67	5.44	69.73	1.15	1.86
BLIP ViT-B [100]	81.90	73.62	10.11	12.76	77.45	5.43	7.10	79.54	2.88	4.05	80.48	1.73	2.51
BLIP ViT-L (zero-shot) [100]	73.66	60.35	18.07	21.66	67.99	7.70	10.16	71.63	2.76	3.93	72.87	1.07	1.61
BLIP ViT-L [100]	82.36	73.93	10.24	12.65	77.93	5.38	7.45	79.81	3.10	4.23	80.98	1.68	2.54
Visual Semantic Embedding models													
VSRN [104]	52.66	45.07	14.41	17.79	47.89	9.06	11.33	49.89	5.26	7.08	50.99	3.17	4.29
SAF [39]	55.46	44.06	20.56	20.29	47.22	14.86	26.71	50.02	9.81	15.12	51.71	6.76	10.85
SGR [39]	57.22	43.57	23.86	28.53	46.98	17.90	22.79	49.81	12.95	17.49	51.91	9.28	13.09
VSE∞ (BUTD region) [19]	58.02	33.94	41.50	46.81	37.15	35.98	42.66	40.39	30.39	37.79	43.17	25.60	33.01
VSE∞ (BUTD grid) [19]	59.40	34.79	41.44	45.95	38.03	35.98	41.75	41.17	30.68	37.14	44.97	24.30	30.57
VSE∞ (WSL grid) [19]	66.06	39.95	39.52	43.79	44.04	33.33	38.44	48.29	26.90	32.85	51.73	21.69	27.51

Given a text encoder f_T, an image encoder f_I, an image x, and an adversarial caption c_p, SC loss is defined by:

$$L_{SC} = \frac{< f_T(c_p), f_I(x) >}{\parallel f_T(c_p) \parallel \parallel f_I(x) \parallel}. \tag{6.2}$$

In each batch, we generate an adversarial caption c_p by randomly selecting words within the caption to be replaced with a probability of p (set to 0.3). These selected words are then substituted with random words from the BERT vocabulary with a probability of q (set to 0.6), or masked with a probability of $1 - q$.

Figure 6.9 illustrates the results of applying the SC loss during the training of the BUTD region in the VSE∞ model. Apart from the addition of the SC loss, we adhere to the official code for training details. The figure demonstrates the improved robustness across the proposed benchmark datasets. By training the model to align closely with the original caption while distancing itself from the adversarial captions, the model can effectively capture word-level details.



(a) Recall@1(↑)

(b) Incorrect Recall@1(↓)

Figure 6.9: **Improvement using Semantic Contrastive Loss.**

Substituting more words

To further analyze the vulnerability of the VL models, we conduct experiments by replacing multiple words. We wonder if the model would confuse even when the original semantic meaning is more broken. Thus, we randomly select between 2 and 5 words and substitute them with words in Bert vocabulary. Since many captions are not long, words are not limited to nouns and are randomly selected.

We show the results in Table 6.4. Although it is presumed to be an easy task, meaningful performance degradation occurs in the entire model when multiple words are changed. When more than two words are substituted, large-scale VL pre-training models show more robust performance compared to VSE models. Especially, VSE∞ shows the vulnerability even for captions with 5 words changed. We think that VSE∞'s simple pooling operator can be overfitted to COCO dataset.

Meanwhile, Figure 6.10 displays the examples of newly created captions which BLIP (ViT-B) has retrieved as top 1. The figure shows the results when 2 to 5 words are replaced. We observe that Top 1 retrieved caption includes at least one correct key word, such as "motorcyclist" in the first image. While the created captions are not natural, they include some keywords. These results lead us to suspect that the model seems to be paying more attention to certain words than whole sentences.

Top 1: a pile disguised teddy star and dolls in a toy box.

Top 2: a pile of teddy bears and dolls in a toy box.

Top 1: arrival cat looking annoyance arrival large group of pigeons.

Top 2: a cat looking at a large group of pigeons.

Top 1: coincidentally metallic refrigerator freezer sitting in coincidentally muscles.

Top 2: two refrigerators side by side in a kitchen.

Top 1: a television that is on with a white man talking neighbors disgusting signs.

Top 2: a television that is on with a white man talking and campaign signs.

Top 1: a young zebra cynical an adult zebra standing on a elevators brown landscape.

Top 2: dishes zebra standing next to katie zebra in dishes dry grass field.

(a) Two words substitution

Top1: motorcyclist on permits white and blue developers illumination permits suspiciously.

Top2: it's a cloudy night for a ride on the motorcycle.

Top1: a train station sad an awning klan concurrency sad a train on the right platform.

Top2: a train platform with trains on the other side.

Top1: a baths light bent over wartime seo wasn.

Top2: a grey computer mouse and a silver metal key.

Top1: a person umm a partial as viscount ride a named board.

Top2: a man stands next to a very small plane.

Top1: three men in baseball uniforms simplicity fighting yourselves hospitals crowd shuttle them.

Top2: interesting antics by two men during a baseball game.

(b) Three words substitution

Top 1: daughters mound tease cake in their dining room while moms get marriott.

Top 2: daughters frosting a cake in their dining room while moms get water.

Top 1: rounding yellow fire hydrant eliot mommy sidewalk in an urban area.

Top 2: a yellow fire hydrant near the curb of a street.

Top 1: a she chalmers four women walking down a virtual in the rain.

Top 2: a group of four women walking down a street in the rain.

Top 1: a guy cutting off another guy inflicted dazzling from his assure.

Top 2: a guy cutting off another guy's cast from his arm.

Top 1: a detrimental player in peripheral much with striped shorts.

Top 2: undercover facto player swings his racket at undercover facto ball lithuania undercover facto court.

(c) Four words substitution

Top 1: grow boy sits freely bed beds leans over nigel metal laptop.

Top 2: a curious toddler reaches out to touch a laptop computer.

Top 1: mohamed glide grazing together fatally pissed green grassy appealing.

Top 2: two brown horses in a pasture eating grass.

Top 1: comb young curtains in comb pat uniform throwing comb ball.

Top 2: little league baseball player throwing a baseball from the mound.

Top 1: governmental male skate-boarder give governmental white fortification doing governmental conquer.

Top 2: a young skate board rider on top of a metal box.

Top 1: a skier violin projection shouting snow looking lecturer shouting profit.

Top 2: skier in austen flight mundane crossed skiis above composed nat sweating.

(d) Five words substitution

Figure 6.10: **Example of substituting multiple random words.**

6.5 Summary

In this thesis, we propose a robust-evaluation benchmark that can measure the robustness of image-text matching (ITM) models. To the best of our knowledge, it is the first benchmark to test robustness in image-text matching task. Unlike existing studies for the robustness test in computer vision and natural language processing (NLP) area, which generate semantic-preserving texts and images with imperceptible changes, we propose a strategy in the opposite direction to the existing adversarial attack strategy. Our main idea is to create fooling captions and images by minimal changes in embedding feature. From evaluation on various state-of-the-art vision language (VL) models, we discover that both models with and without large-scale pre-training data show significant performance degradation and retrieve the incorrect caption/image at a high rate. Our empirical results raise up necessity of new robust ITM models and our benchmark dataset could promote further robustness studies in ITM task.

Limitations. In the process of randomly replacing words, some unnatural sentences such as "A war on bicycle riding next to a train (man $\rightarrow$ war)" are created. However, these sentences do not violate our intention to test how well the ITM model understands both visual and semantic meaning. Creating benchmarks is a very challenging but important study that can boost improvements of the existing algorithms. We hope that our study can inspire researchers in ITM task and more robustness benchmarks can be created.

Chapter 7

CONCLUSION

In this thesis, we presented advances in robust deep learning against practical challenges in the wild. We discussed the challenges that arise in real-world scenarios, such as noisy labels, imbalanced datasets, and robustness test. To address these challenges, we proposed various solutions, including the Influential Rank post-training, the Influence-Balanced loss function, and the context-rich oversampling method. We also introduced RoCOCO, a robust benchmark dataset that can be used to evaluate the robustness of image-text matching models.

First, our proposed a post-training method named Influential Rank sways the overfitted decision boundary to be correct in the presence of noisy labels. Unlike the existing methods, Influential Rank starts from an overfitted model and makes the model more robust against noisy labels progressively. We have conducted extensive experiments on real-world and synthetic noisy benchmark datasets. The results demonstrate that Influential Rank consistently provides performance gain when combined with multiple state-of-the-art robust learning methods. In addition, we have shown that Influential Rank performs as a detector for video data cleaning or a regularizer to smooth the decision boundary. A limitation of our method is that it requires a small number of clean validation samples to calculate the OSD. It can be difficult to collect image data in some domains despite the limited number of 5 images per class in our exper-

iments. Future work could be developing the metric that does not require additional clean validation samples.

Second, as a cost-sensitive re-weighting method, we proposed a novel influence-balanced loss to solve the overfitting of the majority classes in a class imbalance problem. A model trained on imbalanced class data is susceptible to overfitting due to the high capacity of DNN and the scarcity of samples in certain classes. Therefore, as learning progresses, existing methods are likely to produce undesirable results, such as assigning higher weights to samples from majority classes. Unlike the existing methods, IB loss can robustly assign weights because it directly focuses on a sample's influence on the model. We conducted experiments to demonstrate that our method can improve generalization performance under a class imbalance setting. In addition, our method is easy to be implemented and integrated into existing methods. In the future, we plan to extend our method by incorporating data-level methods or other recent meta-learning methods.

Next, we proposed a novel context-rich oversampling method, as a data-level approach. We tackled the fundamental problem of previous oversampling methods that generate context-limited minority samples, which intensifies the overfitting problem. Our key idea is to transfer the rich contexts of majority samples to minority samples to augment minority samples. The implementation of CMO is simple and intuitive. Extensive experiments on various benchmark datasets demonstrated not only that our CMO significantly improves performance, but also that adding our oversampling method to the basic losses advances the state-of-the-art. However, in some cases, the performance improvement for the minority classes occured with the degraded performance of the majority classes. Future work should be designed to improve the performance of all classes without sacrificing the performance of the many-shot classes.

Finally, with the need of ensuring the robustness of deep learning models to adopt them in real-world applications, we proposed a new benchmark dataset to stress-test the robustness of image-text matching models. Unlike existing studies for the robust-

ness test in computer vision and natural language processing (NLP) area, which generate semantic-preserving texts and images with imperceptible changes, we proposed a strategy in the opposite direction to the existing adversarial attack strategy. Our main idea is to create fooling captions and images by minimal changes in embedding feature. From evaluation on various state-of-the-art vision language (VL) models, we discovered that both models with and without large-scale pre-training data showed significant performance degradation and retrieved the incorrect caption/image at a high rate. Our empirical results raise up necessity of new robust ITM models and our benchmark dataset could promote further robustness studies in ITM task. Since it is the first attempt to create robustness benchmark for image-text matching models, there exists a limitation. In the process of randomly replacing words, some unnatural sentences such as "A war on bicycle riding next to a train (man $\rightarrow$ war)" are created. However, these sentences do not violate our intention to test how well the ITM model understands both visual and semantic meaning. In the future, we plan to create more challenging benchmark with more natural sentences and images.

In conclusion, this thesis contributes to the field of deep learning by proposing novel methods to address practical challenges in the wild. The proposed methods are useful in real-world scenarios where robustness is crucial, such as in medical imaging, autonomous driving, and security systems. The development of RoCOCO provides a benchmark dataset to evaluate the robustness of image-text matching models. The results presented in this thesis highlight the importance of addressing practical challenges to achieve robust and reliable deep learning models for real-world applications. Overall, the contributions presented in this thesis provide a solid foundation for future research in the field of robust deep learning and pave the way for the development of more robust and reliable AI systems in real-world scenarios.

Bibliography

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[2] Héctor Allende, Rodrigo Salas, and Claudio Moraga. A robust and effective learning algorithm for feedforward neural networks based on the influence function. In *Pattern Recognition and Image Analysis*, 2003.

[3] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[4] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 1993.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[6] Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017.

[7] Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 2017.

[8] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[9] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

[10] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012.

[11] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, 2022.

[12] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 2018.

[13] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique

for handling the class imbalanced problem. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining*, 2009.

[14] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[15] Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. *arXiv preprint arXiv:2203.14713*, 2022.

[16] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022.

[17] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017.

[18] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 2002.

[19] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[20] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070, 2019.

[21] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *International Conference on Machine Learning*, 2020.

[22] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[23] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, 2020.

[24] De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Masashi Sugiyama. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] Mengjun Cheng, Yipeng Sun, Longchao Wang, Xiongwei Zhu, Kun Yao, Jie Chen, Guoli Song, Junyu Han, Jingtuo Liu, Errui Ding, et al. Vista: vision and scene text aggregation for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[26] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *Computer Vision – ECCV 2020 Workshops*, 2020.

[27] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Computer Vision – ECCV 2020*, 2020.

[28] Sanghyuk Chun, Wonjae Kim, Song Park, Minsuk Chang, and Seong Joon Oh. Eccv caption: Correcting false negatives by collecting machine-and-human-

verified image-caption associations for ms-coco. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, 2022.

[29] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[30] R. Dennis Cook and Sanford Weisberg. Residuals and Influence in Regression. 1982.

[31] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[32] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[33] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, 2020.

[34] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] 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.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, 2019.

[38] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[39] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[40] Q. Dong, S. Gong, and X. Zhu. Class rectification hard mining for imbalanced deep learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[41] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*, 2021.

[42] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[43] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[44] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

[45] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 2009.

[46] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. 2018.

[47] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[48] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 2013.

[49] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[50] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Vqa-lol: Visual question answering under the lens of logic. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 2020.

[51] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

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

[54] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, 2017.

[55] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[56] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[57] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[58] Tanmay Gupta, Ryan Marten, Aniruddha Kembhavi, and Derek Hoiem. Grit: general robust image task benchmark. *arXiv preprint arXiv:2204.13653*, 2022.

[59] F.R. Hampel. *Robust Statistics: The Approach Based on Influence Functions*. Probability and Statistics Series. Wiley, 1986.

[60] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31*, pages 8527–8537. Curran Associates, Inc., 2018.

[61] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang,

Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, 2005.

[62] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd. In *Advances in Neural Information Processing Systems 32*. 2019.

[63] Haibo He and E.A. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21, 2009.

[64] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[65] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[66] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[67] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems 31*. 2018.

[68] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems*, 2018.

[69] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[70] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6626–6636, June 2021.

[71] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. Technical report, USA, 1980.

[72] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[73] Peng Hu, Xi Peng, Hongyuan Zhu, Liangli Zhen, and Jie Lin. Learning cross-modal retrieval with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[74] C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[75] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[76] Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. Learning with neighbor consistency for noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[77] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL-HLT*, 2018.

[78] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pages 429–449, 2002.

[79] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.

[80] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[81] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2309–2318. PMLR, 2018.

[82] Justin Johnson and Taghi Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6:27, 2019.

[83] H. Kahn and A. W. Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Operations Research*, 1(5):263–278, 1953.

[84] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for

long-tailed recognition. In *International Conference on Learning Representations*, 2020.

[85] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[86] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[87] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[88] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[89] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894, Sydney, Australia, 2017. PMLR.

[90] Jedrzej Kozerawski, Victor Fragoso, Nikolaos Karianakis, Gaurav Mittal, Matthew Turk, and Mei Chen. Blt: Balancing long-tailed datasets with adversarially-perturbed images. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020.

[91] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 2017.

[92] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[93] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[94] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[95] Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

[96] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*. PMLR, 2019.

[97] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[98] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[99] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4899–4908, 2019.

[100] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding

and generation. In *International Conference on Machine Learning*. PMLR, 2022.

[101] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 2021.

[102] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

[103] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[104] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[105] Linjie Li, Zhe Gan, and Jingjing Liu. A closer look at the robustness of vision-and-language pre-trained models. *arXiv preprint arXiv:2012.08673*, 2020.

[106] Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[107] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[108] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020.

[109] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221, June 2021.

[110] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[111] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, 2020.

[112] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[113] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[114] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014.

[115] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[116] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

[117] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236, 2020.

[118] Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang. MESA: boost ensemble imbalanced learning with meta-sampler. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[119] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[120] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[121] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[122] Haoyu Lu, Nanyi Fei, Yuqi Huo, Yizhao Gao, Zhiwu Lu, and Ji-Rong Wen. Cots: Collaborative two-stream vision-language pre-training model for cross-

modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15692–15701, 2022.

[123] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553, 2020.

[124] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Computer Vision – ECCV 2018*, 2018.

[125] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *2011 International Conference on Computer Vision*, 2011.

[126] Feng Mao, Xiang Wu, Hui Xue, and Rong Zhang. Hierarchical video frame sequence representation with deep convolutional graph network. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[127] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

[128] Francisco J Moreno-Barea, Fiammetta Strazzera, José M Jerez, Daniel Urda, and Leonardo Franco. Forward noise adjustment scheme for data augmentation. In *2018 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2018.

[129] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *2019 IEEE/CVF International Conference on*

Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, 2019.

[130] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems 26*. 2013.

[131] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. SELF: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020.

[132] Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for ms-coco. *arXiv preprint arXiv:2004.15020*, 2020.

[133] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[134] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 735–744, October 2021.

[135] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[136] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[137] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 2021.

[138] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[139] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015.

[140] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019.

[141] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[142] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Deep imbalanced attribute classification using visual attention aggregation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[143] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[144] Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Magana, Tristan Thrush, Wojciech Galuba, Devi Parikh, and Douwe Kiela. Human-adversarial visual question answering. *Advances in Neural Information Processing Systems*, 2021.

[145] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[146] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 568–576, 2014.

[147] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

[148] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. How does early stopping help generalization against label noise? In *International Conference on Machine Learning Workshop*, 2020.

[149] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[150] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[151] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[152] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 2017.

[153] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[154] Zeren Sun, Fumin Shen, Dan Huang, Qiong Wang, Xiangbo Shu, Yazhou Yao, and Jinhui Tang. Pnp: Robust learning from noisy labels by probabilistic noise prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[155] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 2017.

[156] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[157] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[158] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

[159] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[160] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[161] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*. PMLR, 2019.

[162] Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[163] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, 2020.

[164] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 20–36, 2016.

[165] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021.

[166] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, 2017.

[167] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *NeurIPS*, 2019.

[168] Yun Wang, Tong Zhang, Xueya Zhang, Zhen Cui, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Wasserstein coupled graph learning for cross-modal retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[169] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representation*, 2022.

[170] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1884–1894, June 2021.

[171] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 2015.

[172] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[173] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[174] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems 32*. 2019.

[175] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[176] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[177] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. PMLR, 2019.

[178] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

[179] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[180] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[181] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[182] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[183] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[184] J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

[185] Kun Zhang, Zhendong Mao, Quan Wang, and Yongdong Zhang. Negative-aware attention framework for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15661–15670, 2022.

[186] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[187] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[188] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[189] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

[190] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020.

[191] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020.

[192] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[193] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[194] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: From clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2021.

초 록

딥러닝은 다양한 인공지능 문제를 해결하는 데에서 놀라운 성공을 거두었다. 그러나 실제 환경에서 적용할 때, 데이터 불균형, 잘못된 (노이지) 라벨 및 신뢰도 테스트와 같은 문제로 인해 기계학습 모델의 일반화 성능과 강건성이 종종 도전 받는다.

본 논문에서는 이러한 문제를 해결하기 위한 전략과 딥러닝 모델의 강건성 향상을 제안한다. 구체적으로, 본 논문에서는 데이터 불균형 및 노이지 라벨을 다루기 위한 새로운 방법을 제시한다. 제안된 방법은 가장 인기 있는 벤치마크 데이터셋에서 평가되었으며, 결과는 딥러닝 모델의 일반화 성능과 강건성을 크게 향상시킬 수 있음을 보여준다.

또한, 본 논문에서는 멀티모달 모델의 강건성을 스트레스 테스트하기 위한 새로운 벤치마크 데이터셋인 RoCOCO를 소개한다. 이 데이터셋은 실제 세계의 변화를 시뮬레이션하여, 인공지능 모델의 강건성을 평가하는 보다 현실적이고 도전적인 테스트베드를 제공한다.

결론적으로, 이 논문에서 제시된 연구는 기계학습 모델을 실제 환경에서 적용할 때 발생하는 도전을 더 잘 다룰 수 있는 강건한 딥러닝 기술의 발전에 기여한다. 하지만, 향후 연구에서는 제안된 방법들의 한계점을 극복하고 더 많은 현실적인 시나리오에서의 강건성 평가를 위해 노력해야 할 것이다.

주요어: 데이터 불균형, 긴꼬리분포, 이미지 분류, 오버샘플링, 데이터 증강, 노이지 라벨, 강건한 인공지능, 멀티모달, 이미지-텍스트 매칭, 벤치마크 데이터

학번: 2019-35916