

# Revising the Problem of Partial Labels from the Perspective of CNNs’ Robustness

1<sup>st</sup> Xin Zhang

*University of Southern Maine*  
Portland, United States  
xin.zhang@maine.edu

2<sup>nd</sup> Yuqi Song

*University of Southern Maine*  
Portland, United States  
yuqi.song@maine.edu

3<sup>rd</sup> Wyatt McCurdy

*University of Southern Maine*  
Portland, United States  
wyatt.mccurdy@maine.edu

4<sup>th</sup> Xiaofeng Wang

*University of South Carolina*  
Columbia, United States  
wangxi@cec.sc.edu

5<sup>th</sup> Fei Zuo

*University of Central Oklahoma*  
Edmond, United States  
fzuo@uco.edu

**Abstract**—Convolutional neural networks (CNNs) have gained increasing popularity and versatility in recent decades, finding applications in diverse domains. These remarkable achievements are greatly attributed to the support of extensive datasets with precise labels. However, annotating image datasets is intricate and complex, particularly in the case of multi-label datasets. Hence, the concept of partial-label setting has been proposed to reduce annotation costs, and numerous corresponding solutions have been introduced. The evaluation methods for these existing solutions have been primarily based on accuracy. That is, their performance is assessed by their predictive accuracy on the test set. However, we insist that such an evaluation is insufficient and one-sided. On one hand, since the quality of the test set has not been evaluated, the assessment results are unreliable. On the other hand, the partial-label problem may also be raised by undergoing adversarial attacks. Therefore, incorporating robustness into the evaluation system is crucial. For this purpose, we first propose two attack models to generate multiple partial-label datasets with varying degrees of label missing rates. Subsequently, we introduce a lightweight partial-label solution using pseudo-labeling techniques and a designed loss function. Then, we employ D-Score to analyze both the proposed and existing methods to determine whether they can enhance robustness while improving accuracy. Extensive experimental results demonstrate that while certain methods may improve accuracy, the enhancement in robustness is not significant, and in some cases, it even diminishes.

**Index Terms**—computer vision, multi-label classification, CNN robustness, partial labels

## I. INTRODUCTION

Convolutional Neural Networks (CNNs), with the ability to extract and learn features automatically from raw data have revolutionized the field of computer vision and have achieved state-of-the-art results on various tasks, including image classification [1], object detection [2], semantic segmentation [3], and so on [4]–[6]. Recently, with the rapid development of deep learning techniques, CNNs have also become an indispensable tool for many real-world computer vision applications, such as self-driving cars [7], security and surveillance systems [8], and medical diagnosis [9].

The tremendous success of CNNs is largely attributed to the support of accurate labeling. However, acquiring precise annotations is quite expensive. To economize on annotation costs, previous endeavors introduced the notion of the ‘partial-label setting’ and suggested various methodologies to tackle this problem, enabling CNNs to employ only a fraction of the labels during training [10]–[12]. After a comprehensive review of the literature on the partial-label problem, we noticed that prior works have primarily assessed their proposed methods based on accuracy alone. We consider it one-sided to conclude the effectiveness of the proposed methods in addressing the partial-label problem solely based on this type of evaluation. On the one hand, previous works merely demonstrated improved accuracy of their proposed solutions on predefined test sets without evaluating the test sets themselves. Therefore, such evaluation results may not be reliable. On the other hand, besides saving annotation costs, adversarial attacks are one of the reasons for partial-label problems. Extensive prior research has proven that CNNs are vulnerable to adversarial attacks [13], which is a type of attack used to deteriorate the performance of CNNs targeting datasets, image features, label information, or the models themselves. CNNs with poor robustness often experience significant performance degradation when subjected to adversarial attacks. Therefore, we insist that analyzing the partial-label problem solely from the perspective of accuracy is one-sided. We also need to analyze it from the standpoint of robustness, that is, analyzing CNN’s robustness with respect to label removal.

To conduct an analysis of the partial-label problem from a robustness perspective, we first require datasets that have been subjected to adversarial attacks. Such datasets should consist of training images where only a portion of the labels is known after the adversarial attacks. The current datasets are either fully labeled or partially labeled with a fixed quantity of missing labels, making it challenging to effectively verify how proposed methods are affected by varying degrees of label loss. For this purpose, we initially propose two attack

models: random attacks  $\mathcal{R}_p$  and targeted attacks  $\mathcal{T}_p$ . The former randomly removes  $p\%$  of the labels from the images in the training set, irrespective of whether they are positive or negative labels. The latter selectively targets only the positive labels in the training set, removing  $p\%$  of the positive labels while preserving all the negative labels. Furthermore, we introduce a lightweight solution to the partial-label problem. It leverages pseudo-labeling techniques and a well-designed loss function. Moreover, to evaluate whether our method and existing approaches enhance robustness concerning label removal, besides using the mAP evaluation metric, we also employed the D-Score [14] analysis method to assess the robustness of these methods.

Our Contributions are summarized as follows:

- We propose two adversarial attack models targeting image labels: targeted attacks and random attacks. These attack methods selectively remove certain labels, transforming the full-label setting into a partial-label setting. Experimental results demonstrate that this attack effectively reduces the performance of existing STOA methods.
- We present a lightweight approach to address the partial-label problem, which is achieved through the utilization of pseudo-labeling techniques and an improved loss function, without the need for additional statistical information or network structures.
- The extensive experiments on three large-scale public image datasets (COCO, NUS-WIDE, and Pascal VOC) demonstrate that our method outperforms the STOA methods, both in terms of accuracy (mAP) and robustness (D-Score).

The rest of the paper is organized as follows. Section II discusses the related work. Our proposed method is presented in Section III. Section IV shows the experimental settings and results. Finally, conclusions are drawn in Section V.

## II. RELATED WORKS

### A. Partial-label Problems

The partial-label problem means that for one input image in the training set, only a subset of all the labels for it can be observed and the rest remains unknown during the training process [10], [12]. Addressing this problem is meaningful for saving annotation costs.

A straightforward approach for the partial-label problem is BR [15], which decomposes the task into a number of binary classification problems, each for one label. Such an approach encounters many difficulties, mainly due to ignoring correlations between labels. PU-learning is an alternative solution [16], which studies the problem with a small number of positive examples and a large number of unlabeled examples for training. Most methods can be divided into the following three categories: two-step techniques [17], biased learning [18], and class prior incorporation [19]. However, all these methods require that the training data consists of

positive and unlabeled examples [20]. Pseudo-label [10] is another solution. Pseudo-labeling was first proposed in [21]. The goal of pseudo-labeling in partial-label problems is to generate pseudo-labels for the unobserved part [11].

### B. The Evaluation for CNNs' Robustness

To evaluate CNNs, researchers have proposed several approaches, which can be divided into two categories. The first category involves introducing the traditional software engineering testing method, mutation testing, to CNNs [22]–[24]. This approach applies carefully designed mutation operators [25] to the CNN model to generate multiple variants. The higher the number of differences between the predictions of the variant models and the original model, the higher the quality of the test set. However, the score itself remains a black box, and the reasons behind the low quality of the test set are still unknown. Additionally, effective methods for selecting and combining mutation operators to detect test set quality remain unexplored [26]. The second category of approaches is based on neuron coverage [27]–[29]. These methods use gradient ascent to solve a joint optimization problem that maximizes both neuron coverage and the number of potentially erroneous behaviors, and eventually generate a set of test inputs [27]. However, as noted in [30], higher neuron coverage can lead to fewer defects detected, less natural inputs, and more biased prediction preferences. Therefore, developing effective methods for providing white-box scores for CNNs and proposing methods for enhancing these scores is critical for improving robustness and accuracy of CNNs.

## III. METHODOLOGY

In this section, we introduce details of our proposed methods, including two adversarial attack models, the solution to the partial-label problem, and the evaluation methods.

### A. Simulation of Targeting-label Attack

Label removal is one of the most prevalent adversarial attacks that specifically targets labels. It operates by altering the label distribution through the removal of ground-truth labels, thereby diminishing the model's accuracy and potentially impeding the training process. To validate our method's efficacy in combating adversarial attacks and fortifying the CNN's robustness, we must initially simulate this targeted label attack. We've devised two attack models based on the positive or negative attributes of the targeted labels.

- Targeted attacks. In this attack model, we directly eliminate all negative labels and a certain proportion of positive labels. We designate this attack model as  $\mathcal{T}_q$ , where  $q$  represents the deletion percentage of positive labels. Essentially, this attack method removes  $\hat{q}$  percent of the labels.:

$$\hat{q}\% = \frac{t_n + q\% \times t_p}{t}$$

where  $t$ ,  $t_n$ , and  $t_p$  stand for the number of total labels, the number of negative labels, and the number of positive labels, respectively.

- **Random attacks.** In this attack model, we do not distinguish between positive and negative labels; instead, we directly delete labels based on a specific proportion. We denote this attack model as  $\mathcal{R}_q$ , where  $q$  represents the percentage of labels deleted. It is worth noting that this attack model could result in an extreme scenario where all positive labels are removed. This implies that for an image in the training set, its corresponding label contains only negative labels. This situation could easily lead the model to generate a trivial solution, significantly reducing its accuracy. Hence, when designing a solution, it is crucial to address this extreme label imbalance.

These two attacking models are summarized in Figure. 1.

### B. The Solution for Targeting-label Attack

To avoid introducing additional computational burden to the model, we propose a lightweight solution that involves modifying only the loss function to enhance the CNN’s robustness against targeting-label attacks.

**Pseudo-label.** We can divide the labels associated with an image into two parts:  $E$ , which persists after the attack, and  $N$ , the labels removed due to the attack. For multi-label classification tasks, Binary Cross-Entropy (BCE) commonly serves as the primary loss function, as shown in Equation. 1,

$$\mathcal{L}_{bce}(\hat{y}, y) = -\frac{1}{L} \sum_{i=1}^L [(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]. \quad (1)$$

where  $y$  and  $\hat{y}$  stand for the ground-truth labels and predictions of the classifier, respectively. For part  $E$ , we compute the loss using the original ground-truth as the target. For the  $N$  part, we introduce pseudo-labels and employ them as targets to calculate the loss. The pseudo-label begins with an initialization of 1, and its value undergoes updates using a historical stack. This stack retains the model’s predictions for this label from the previous three epochs. These processes are summarized in Equation. 2,

$$\tilde{y}_{i_j} = \begin{cases} f(S_{i_j}, \alpha, \beta, \gamma), & \text{update} \\ 1, & \text{initialization} \end{cases} \quad (2)$$

where  $S_{i_j}$  stands for the historical stack, which always reserves the predictions of the last three epochs for the  $i$ th image’s  $j$ th category, that is,  $size(S_{i_j}) = 3$ .  $\alpha$ ,  $\beta$ , and  $\gamma$  stand for the weight for the three elements in  $S_{i_j}$  during the calculation of function  $f(\cdot)$ ,

$$f(S_{i_j}, \alpha, \beta, \gamma) = \alpha S_{i_j}[0] + \beta S_{i_j}[1] + \gamma S_{i_j}[2], \quad (3)$$

where  $S_{i_j}[k] = \hat{y}_{i_j}^{e-k}$ ,  $e$  represents the index of current epoch number, and  $\hat{y}_{i_j}$  stands for the prediction for the  $i$ th image’s  $j$ th category. The values of  $\alpha$ ,  $\beta$ , and  $\gamma$  are decided by extensive experiments, and  $\alpha + \beta + \gamma = 1$ .

It is essential to note that initializing the pseudo-label as 1 stems from the prevalence of numerous negative labels in image datasets. This often leads to label imbalance, potentially prompting the model to generate trivial solutions, that is, directly predicting each category as negative. The initialization of the pseudo-label as 1 effectively alleviates this issue. While updating the pseudo-label, the historical stack aids in tracking the label value fluctuations over the past three instances, ensuring a smoother update.

**Loss function.** We employ Binary Cross Entropy (BCE) as our loss function. In the part  $E$ , we compute the loss value using ground-truth as the target, while for the  $N$  part, we calculate the loss value using the pseudo label as the target, as illustrated in Equation. 4,

$$\mathcal{L} = \mathcal{L}_{bce}(\hat{y}, y) + \mathcal{L}_{bce}(\hat{y}, \tilde{y}), \quad (4)$$

where  $y$ ,  $\hat{y}$ , and  $\tilde{y}$  stand for the ground-truth labels, the predictions, and the pseudo labels respectively. In Equation. 4, the first term represents for the loss value of the  $E$  part, and the second term stands for the  $N$  part. Building upon this, we introduce an attention-shifting parameter  $e(\cdot)$  to progressively redirect attention from the part  $E$  to the part  $N$  during the loss function computation. The rationale behind this design is that at the initial training stages, since pseudo labels are initially set to 1, they may significantly deviate from the actual labels, potentially resulting in unreliable loss value calculations for this part. Therefore, during the early training phase, we aim to focus more on the  $E$  part. As training advances and the pseudo labels are continually updated, their reliability gradually increases. Consequently, as training progresses, we gradually shift attention towards the  $N$  part.

In previous research, the attention-shifting parameter  $e(\cdot)$  has often been applied using a linear function. However, in this context, we employ an exponential function for implementation, as Equation 5,

$$e(n_c, n_t) = e^{n_c - n_t}, \quad (5)$$

where  $n_c$  and  $n_t$  stand for the index of the current epoch number and the total epoch numbers respectively. Compared to the linear function, the exponential function exhibits faster changes toward the end of the training process, while its alterations are more gradual during the initial training stages. This approach ensures that the pseudo labels have ample time for updating during training and gradually become dominant in calculating the loss value as training progresses.

In addition, to prevent the occurrence of trivial solutions, we design an approach for penalizing such outcomes. This approach calculates the difference between the current model’s predictions and the trivial solution. When this disparity is tiny, it indicates that the current model has potentially produced a trivial solution. In such cases, we apply a penalty  $\mathcal{P}$  to these predictions. We use the L2 norm to compute the difference

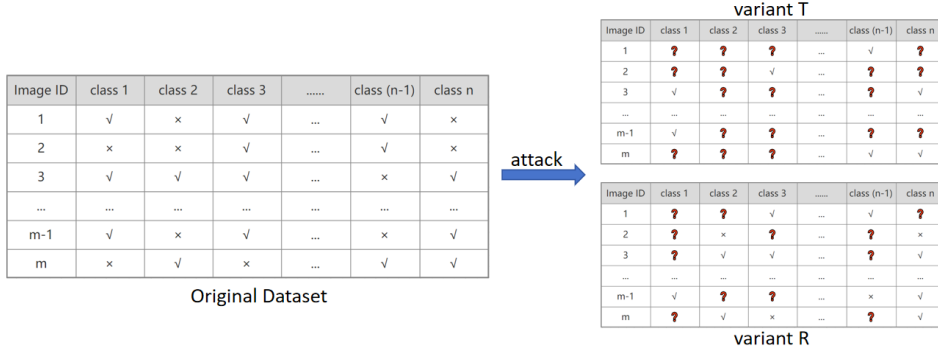


Fig. 1. Two attacking models. Variant T is generated under attack  $\mathcal{T}$ , which removes all negative labels and some positive labels. Variant R, on the other hand, is generated under attack  $\mathcal{R}$ , where the positive or negative nature of the label is disregarded, and the deletion of labels is entirely random.

between the predictions and the trivial solution, as depicted in Equation. 6,

$$\mathcal{P} = 1 - \sqrt{\sum_{i=0}^m (\hat{y}_i - \tilde{y}_i)^2}, \quad (6)$$

where  $m$  stands for the number of existing labels for this image, and because  $\sqrt{\sum_{i=0}^m (\hat{y}_i - \tilde{y}_i)^2} \in [0, 1]$ ,  $\mathcal{P}$  is always larger than 0.

Combining all these, our final loss function is shown as Equation. 7,

$$\mathcal{L} = \mathcal{L}_{bce}(\hat{y}, y) + e^{n_c - n_t} \times \mathcal{L}_{bce}(\hat{y}, \tilde{y}) + \mathcal{P}, \quad (7)$$

### C. Evaluating the Robustness of Proposed Solutions

To demonstrate that our proposed solutions can indeed improve the robustness, we adopt D-Score [14] to analyze the presented method and other comparisons. D-Score is a quantitative method for analyzing the robustness of CNNs. It analyzes the model’s attention distribution and the dataset’s feature distribution through the deletion of mutation operators and feature shifting. Subsequently, it evaluates the CNN’s robustness by computing the similarity between these two distributions.

## IV. EXPERIMENTS

The detailed experimental settings and results are summarized in this section.

### A. Datasets

We conduct comprehensive experiments on three large-scale multi-label image datasets: COCO [31], NUS-WIDE [32], and Pascal VOC [33]. Each instance in these three datasets is fully annotated with clean labels that can be used as the GT in performance evaluation.

### B. Network Structure and Hyper-Parameters

Following [10], we adopt the same network structure by using an end-to-end network for all experiments: a ResNet-50 [34], pre-trained on ImageNet [35], as the backbone and a fully connected layer, which is the same as the multi-label

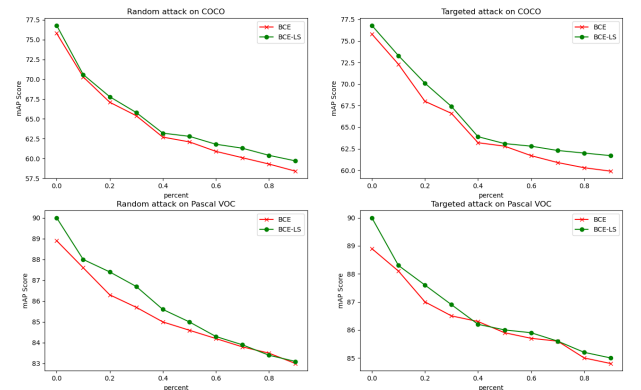


Fig. 2. The effectiveness of our attacking models in decreasing the mAP scores. The first row is conducted on the COCO dataset, and the second row is on Pascal VOC.

classifier under FOL setting. Our approach does not add any extra structure to the network.

Additionally, we also follow the same training hyperparameter selection to control variables. That is, we train our classifier for 10 epochs, and for the learning rate and batch size, we use a hyperparameter search method and select the hyperparameters with the best mAP on the validation set, where the learning rate is in  $[1e - 3, 1e - 4, 1e - 5, 1e - 6]$  and batch size is in  $[8, 16]$ .

### C. Experimental Results

**The effectiveness of attacking models.** To demonstrate the significant threat our proposed attack models pose to CNNs, we opted to assess the performance of two commonly employed solutions for multi-label classification problems: Binary Cross-Entropy (BCE) and BCE with Label Smoothing (BCE-LS). This evaluation was conducted on COCO and Pascal VOC, two extensive image datasets, under varying degrees of attack. The experimental results are shown in Figure. 2, which fully demonstrate the effectiveness of our proposed attacking models.

**The effectiveness of our proposed method in addressing the partial-label problem.** To demonstrate the effectiveness of our approach, we choose several state-of-the-art methods as comparisons, including AN [36], WAN [37], ROLE [38]. Then, we conduct targeted attacking and random attacking on three public image datasets, that is, COCO, Pascal VOC and NUS-WIDE, to generate several variants of training sets. The experimental results under targeted attacking are summarized in Table. I. There is also a special variation in this table, namely  $\mathcal{T}_s$ . This variant is generated through a specific form of targeted attack, wherein for each image, we retain only one positive label and eliminate all other labels. This variant holds particular practical significance: in such scenarios, we only require a single annotation for each image, leading to a substantial reduction in annotation costs. Hence, we specifically examine the performance of our approach concerning this variant. Table. II summarizes the results by random attacking. These results demonstrate the effectiveness of our proposed method in addressing the partial-label problem and resisting the two proposed attacks. **The analysis of robustness.** We summarize the results of the D-Score analysis in Table. III.

## V. CONCLUSION

The remarkable success of CNNs relies heavily on the support of large, high-quality labeled datasets. However, acquiring such datasets is costly due to the extensive manual annotation involved, particularly in multi-label datasets. To address this challenge, numerous methods have been proposed to train CNNs using partial-label datasets. Yet, the evaluation of these solutions has been limited to accuracy, which we deem insufficient. It is crucial to include robustness in the assessment, as the quality of the test sets used for evaluation remains unverified and the partial-label issue may stem from adversarial attacks, closely linked to CNNs' robustness. To tackle these challenges, we introduce two adversarial attack models aimed at removing specific labels and generating partial-label datasets. Subsequently, we propose a lightweight solution for partial-label problems using pseudo-label techniques. Finally, we conduct an analysis using D-Score and mAP evaluation metrics to assess both the robustness and accuracy of our proposed method and some state-of-the-art methods. Experimental results demonstrate that while our method significantly enhances accuracy, it also notably improves robustness. Conversely, certain existing methods exhibit improved accuracy but a simultaneous decline in robustness.

## REFERENCES

- [1] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.
- [2] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.
- [3] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *European conference on computer vision*. Springer, 2012, pp. 746–760.
- [4] Y. Song, M. Gao, J. Yu, and Q. Xiong, "Social recommendation based on implicit friends discovering via meta-path," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 197–204.
- [5] Y. Song, E. M. D. Siriwardane, Y. Zhao, and J. Hu, "Computational discovery of new 2d materials using deep learning generative models," *ACS Applied Materials & Interfaces*, vol. 13, no. 45, pp. 53 303–53 313, 2021.
- [6] R. Dong, Y. Zhao, Y. Song, N. Fu, S. S. Omeel, S. Dey, Q. Li, L. Wei, and J. Hu, "Deepxrd, a deep learning model for predicting xrd spectrum from material composition," *ACS Applied Materials & Interfaces*, vol. 14, no. 35, pp. 40 102–40 115, 2022.
- [7] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.
- [8] V. Kafedziski, S. Pecov, and D. Tanevski, "Detection and classification of land mines from ground penetrating radar data using faster r-cnn," in *2018 26th telecommunications forum (TELFOR)*. IEEE, 2018, pp. 1–4.
- [9] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.
- [10] X. Zhang, R. Abdelfattah, Y. Song, and X. Wang, "An effective approach for multi-label classification with missing labels," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2022, pp. 1713–1720.
- [11] W. Shi, Y. Gong, C. Ding, Z. M. Tao, and N. Zheng, "Transductive semi-supervised deep learning using min-max features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 299–315.
- [12] X. Zhang, Y. Song, F. Zuo, Z. Zhou, and X. Wang, "Towards imbalanced large scale multi-label classification with partially annotated labels," in *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*. IEEE, 2023, pp. 195–200.
- [13] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Applied Sciences*, vol. 9, no. 5, p. 909, 2019.
- [14] X. Zhang, Y. Song, X. Wang, and F. Zuo, "D-score: A white-box diagnosis score for cnns based on mutation operators," *arXiv preprint arXiv:2304.00697*, 2023.
- [15] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.
- [16] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *IJCAI*, vol. 3, no. 2003. Citeseer, 2003, pp. 587–592.
- [17] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML*, vol. 2, no. 485. Sydney, NSW, 2002, pp. 387–394.
- [18] S. Sellamanickam, P. Garg, and S. K. Selvaraj, "A pairwise ranking based approach to learning with positive and unlabeled examples," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 663–672.
- [19] L. Jiang, H. Zhang, and Z. Cai, "A novel bayes model: Hidden naive bayes," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 10, pp. 1361–1371, 2008.
- [20] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020.
- [21] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.
- [22] Q. Hu, L. Ma, X. Xie, B. Yu, Y. Liu, and J. Zhao, "Deepmutation++: A mutation testing framework for deep learning systems," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019, pp. 1158–1161.
- [23] N. Humbatova, G. Jahangirova, and P. Tonella, "Deepprime: mutation testing of deep learning systems based on real faults," in *Proceedings of*

TABLE I

THE MAP RESULTS ON COCO, PASCAL VOC, AND NUSWIDE DATASETS ATTACKED BY  $\mathcal{T}$ .  $\mathcal{T}_{0.2}$ ,  $\mathcal{T}_{0.4}$ , AND  $\mathcal{T}_{0.6}$  REPRESENT RANDOMLY RETAINING 80%, 60% AND 40% OF POSITIVE LABELS FOR EACH INSTANCE, RESPECTIVELY, AND DISCARDING ALL NEGATIVE LABELS. IN EACH COLUMN, WE BOLD THE BEST-PERFORMING METHOD.

	COCO				Pascal VOC				NUSWIDE			
	$\mathcal{T}_s$	$\mathcal{T}_{0.6}$	$\mathcal{T}_{0.4}$	$\mathcal{T}_{0.2}$	$\mathcal{T}_s$	$\mathcal{T}_{0.6}$	$\mathcal{T}_{0.4}$	$\mathcal{T}_{0.2}$	$\mathcal{T}_s$	$\mathcal{T}_{0.6}$	$\mathcal{T}_{0.4}$	$\mathcal{T}_{0.2}$
AN [36]	63.9	66.4	67.0	69.2	84.7	86.8	87.3	88.0	40.0	45.1	48.3	50.3
WAN [37]	64.8	69.1	70.8	71.2	85.9	87.5	87.9	88.0	43.7	46.9	48.5	50.6
ROLE [38]	65.9	73.1	75.4	77.1	87.0	88.9	90.0	90.3	43.2	48.0	50.4	52.0
Ours	<b>67.1</b>	<b>74.0</b>	<b>75.8</b>	<b>77.4</b>	<b>87.2</b>	<b>89.2</b>	<b>90.3</b>	<b>90.8</b>	<b>44.9</b>	<b>48.7</b>	<b>50.9</b>	<b>52.8</b>

TABLE II

THE MAP RESULTS ON COCO, PASCAL VOC, AND NUSWIDE DATASETS ATTACKED BY  $\mathcal{R}$ .  $\mathcal{R}_{0.2}$ ,  $\mathcal{R}_{0.4}$ , AND  $\mathcal{R}_{0.6}$  REPRESENT RANDOMLY RETAINING 80%, 60% AND 40% OF LABELS FOR EACH INSTANCE, RESPECTIVELY. IN EACH COLUMN, WE BOLD THE BEST-PERFORMING METHOD.

	COCO			Pascal VOC			NUSWIDE		
	$\mathcal{R}_{0.6}$	$\mathcal{R}_{0.4}$	$\mathcal{R}_{0.2}$	$\mathcal{R}_{0.6}$	$\mathcal{R}_{0.4}$	$\mathcal{R}_{0.2}$	$\mathcal{R}_{0.6}$	$\mathcal{R}_{0.4}$	$\mathcal{R}_{0.2}$
AN [36]	60.1	62.3	63.7	78.6	80.1	82.3	38.7	40.5	44.1
WAN [37]	60.8	63.1	64.3	80.2	81.7	84.4	40.2	41.1	43.7
ROLE [38]	61.1	63.9	65.0	81.0	83.1	84.9	40.8	41.9	44.2
Ours	<b>61.6</b>	<b>64.3</b>	<b>66.0</b>	<b>81.8</b>	<b>85.4</b>	<b>85.9</b>	<b>42.1</b>	<b>42.7</b>	<b>44.7</b>

TABLE III

THE MAP, ROBUSTNESS SCORE, FITNESS SCORE AND D-SCORE RESULTS UNDERGOING  $\mathcal{T}_{0.4}$  ON COCO AND PASCAL VOC.  $\mathcal{T}_{0.4}$  REPRESENTS RANDOMLY RETAINING 60% OF POSITIVE LABELS FOR EACH INSTANCE, RESPECTIVELY, AND DISCARDING ALL NEGATIVE LABELS. FOR THE ROBUSTNESS SCORE, THE SMALLER THE BETTER. AS FOR THE MAP, FITNESS SCORE AND D-SCORE, THE HIGHER THE BETTER. IN EACH COLUMN, WE BOLD THE BEST-PERFORMING METHOD AND UNDERLINE THE SECOND-BEST ONE.

	COCO $\mathcal{T}_{0.4}$				Pascal VOC $\mathcal{T}_{0.4}$			
	mAP	robust	fitness	D-Score	mAP	robust	fitness	D-Score
AN [36]	67.0	0.397	0.765	0.368	87.3	0.288	0.816	0.528
WAN [37]	70.8	0.416	0.777	0.361	87.9	0.306	0.831	0.525
Role [38]	<u>75.4</u>	<u>0.289</u>	<u>0.801</u>	<u>0.512</u>	<u>90.0</u>	<u>0.261</u>	<u>0.847</u>	<u>0.568</u>
Ours	<b>75.8</b>	<b>0.272</b>	<b>0.809</b>	<b>0.537</b>	<b>90.3</b>	<b>0.237</b>	<b>0.852</b>	<b>0.615</b>

the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2021, pp. 67–78.

- [24] W. Shen, J. Wan, and Z. Chen, “Munn: Mutation analysis of neural networks,” in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2018, pp. 108–115.
- [25] L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, Y. Liu, J. Zhao *et al.*, “Deepmutation: Mutation testing of deep learning systems,” in *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2018, pp. 100–111.
- [26] A. Panichella and C. C. Liem, “What are we really testing in mutation testing for machine learning? a critical reflection,” in *2021 IEEE/ACM 43rd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE, 2021, pp. 66–70.
- [27] K. Pei, Y. Cao, J. Yang, and S. Jana, “Deepxplore: Automated whitebox testing of deep learning systems,” in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.
- [28] Y. Feng, Q. Shi, X. Gao, J. Wan, C. Fang, and Z. Chen, “Deepgini: prioritizing massive tests to enhance the robustness of deep neural networks,” in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020, pp. 177–188.
- [29] J. Yu, Y. Fu, Y. Zheng, Z. Wang, and X. Ye, “Test4deep: an effective white-box testing for deep neural networks,” in *2019 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*. IEEE, 2019, pp. 16–23.
- [30] F. Harel-Canada, L. Wang, M. A. Gulzar, Q. Gu, and M. Kim, “Is neuron coverage a meaningful measure for testing deep neural networks?” in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 851–862.
- [31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [32] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “Nus-wide: a real-world web image database from national university of singapore,” in *Proceedings of the ACM international conference on image and video retrieval*, 2009, pp. 1–9.
- [33] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [36] K. Kundu and J. Tighe, “Exploiting weakly supervised visual patterns to learn from partial annotations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 561–572, 2020.
- [37] O. Mac Aodha, E. Cole, and P. Perona, “Presence-only geographical priors for fine-grained image classification,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9596–9606.
- [38] E. Cole, O. Mac Aodha, T. Lorieul, P. Perona, D. Morris, and N. Jojic, “Multi-label learning from single positive labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 933–942.