Category-Wise Fine-Tuning for Image Multi-label Classification with Partial Labels

Chak Fong Chong, Xu Yang(✉), Tenglong Wang, Wei Ke, and Yapeng Wang

Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, China
{chakfong.chong,xuyang,p1807530,wke,yapengwang}@mpu.edu.mo

Abstract. Image multi-label classification datasets are often partially labeled (for each sample, only the labels on some categories are known). One popular solution for training convolutional neural networks is treating all unknown labels as negative labels, named *Negative* mode. But it produces wrong labels unevenly over categories, decreasing the binary classification performance on different categories to varying degrees. On the other hand, although *Ignore* mode that ignores the contributions of unknown labels may be less effective than *Negative* mode, it ensures the data have no additional wrong labels, which is what *Negative* mode lacks. In this paper, we propose **C**ategory-wise **F**ine-**T**uning (**CFT**), a new post-training method that can be applied to a model trained with *Negative* mode to improve its performance on each category independently. Specifically, CFT uses *Ignore* mode to one-by-one fine-tune the logistic regressions (LRs) in the classification layer. The use of *Ignore* mode reduces the performance decreases caused by the wrong labels of *Negative* mode during training. Particularly, Genetic Algorithm (GA) and binary crossentropy are used in CFT for fine-tuning the LRs. The effectiveness of our methods was evaluated on the CheXpert competition dataset and achieves state-of-the-art results, to our knowledge. A single model submitted to the competition server for the official evaluation achieves mAUC 91.82% on the test set, which is the highest single model score in the leaderboard and literature. Moreover, our ensemble achieves mAUC 93.33% (The competition was recently closed. We evaluate the ensemble on a local machine after the test set is released and can be downloaded.) on the test set, superior to the best in the leaderboard and literature (93.05%). Besides, the effectiveness of our methods is also evaluated on the partially labeled versions of the MS-COCO dataset.

Keywords: Partial Labels · Partial Annotations · Multi-Label Classification · Multi-Label Recognition

1 Introduction

Image multi-label classification (MLC) is a typical computer vision problem that classifies the presence (positive) or absence (negative) of multiple categories in

B. Luo et al. (Eds.): ICONIP 2023, CCIS 1965, pp. 332–345, 2024.
https://doi.org/10.1007/978-981-99-8145-8_26

each image. As an image usually contains multiple objects or concepts, it is more practical than its counterpart single-label classification and hence has a wide range of applications like medical image interpretation [6,7,21].

A crucial challenge of training convolutional neural networks (CNNs) for image MLC is the training data is often partially labeled [17,27]. That is, for each image sample, only the labels on some categories are known, and the rest are unknown. It is because the manual collection of fully labeled data is expensive [13], especially when the numbers of categories and samples are very large.

A popular and effective solution for training CNN with partially labeled data is treating all unknown labels as negative labels [2,3,26,34], named ***Negative* mode** [1]. This mode is based on the prior knowledge of MLC datasets that negative labels are usually much more than positive labels [28]. Nevertheless, this mode produces wrong labels to the training data, as some unknown labels' ground truths are positive labels instead of negative labels. These wrong labels are usually unevenly distributed over different categories [1]. The categories with more wrong labels suffer from more harm. Therefore, different categories suffer from varying degrees of performance decreases.

On the other hand, another solution is ignoring the contributions of unknown labels [1,13], named ***Ignore* mode** [1]. This mode may be less effective than *Negative* mode [26], as it does not utilize the prior knowledge that negative labels are in the majority. Even so, it ensures the training data have no additional wrong labels, which is a vital advantage that *Negative* mode lacks. Therefore, several work utilize this vital advantage of *Ignore* to improve *Negative* mode for training CNNs beginning with initial parameters [1,26].

In this paper, we propose **C**ategory-wise **F**ine-**T**uning (**CFT**), a new post-training method that can be applied to a CNN that has been trained with *Negative* mode to improve its binary classification performance on each category independently. Therefore, CFT is very different from most approaches that train a CNN from initial parameters [1,26]. Specifically, CFT uses *Ignore* mode to one-by-one fine-tune the logistic regressions (LRs) in the classification layer, in which each LR outputs the binary classification result on one category. The use of *Ignore* mode reduces the performance decreases caused by the wrong labels of *Negative* mode during training. The one-by-one fine-tuning can improve the performance on each category independently without affecting the performance on other categories.

While applying CFT to a CNN, the LRs may prefer different fine-tuning configurations (optimization methods, methods for handling untypical labels in particular MLC datasets, etc.) to achieve higher performance. Therefore, we additionally use a greedy selection for CFT to enable choosing the best configuration for each LR from multiple configuration candidates.

During experiments, we found using binary crossentropy (BCE) loss with backpropagation for fine-tuning an LR sometimes unwantedly decreases the performance like AUC (area under the receiver operating characteristic curve). On the other hand, Genetic Algorithm (GA) [29] for fine-tuning can directly improve the performance, avoiding performance drops caused by minimizing BCE.

Sufficient experiments were conducted on the CheXpert [21] competition dataset and the partially labeled versions of the MS-COCO [28] standard MLC dataset to evaluate the effectiveness of our methods. Especially, our methods achieve state-of-the-art on the CheXpert dataset, to the best of our knowledge. We submitted a single CNN to the competition server[1] for the official evaluation on the test set. It achieves mAUC 91.82%, which is the highest single model score in the leaderboard and literature. After that, the competition server was closed and the test set is released. Therefore, our ensemble composed of 5 single CNNs was evaluated on a local machine and achieves mAUC 93.33% on the test set, superior to the best in the leaderboard and literature (mAUC 93.05% [44]).

2 Related Work

Several approaches were proposed to address MLC with partial labels. Binary Relevance [15] converts MLC to multiple binary classification tasks, but it usually fails to model the label dependencies and is less scalable to a large number of categories. [23,41,43] adopted low-rank learning, [39] used a mixed graph to encode a network of label dependency, [3,12] predicted unknown labels by learning label relations, and [8,24,38] predicted unknown labels by posterior inference. However, most of these approaches cannot be well-adapted for training deep models, as they require putting all training data into memory or solving costly optimization problems.

Some approaches train deep models with partial labels by exploiting image and category dependencies. Durand *et al.* [13] proposed predicting unknown labels based on curriculum learning with graph neural networks to model the correlations between categories. IMCL [20] interactively learns a model with a similarity learner which discovers label and image dependencies. SST [5] and HST [4] explore the image-specific occurrence and category-specific feature similarities to complement unknown labels. SARB [32] complements unknown labels by learning and blending category-specific feature representation across different images. However, most of these approaches require particular model architectures or training schemes.

Negative mode and *Ignore* mode are more prevalent in contrast with the complex approaches aforementioned. *Ignore* mode simply ignores the contributions of unknown labels (*e.g.*, partial-BCE loss [13] and partial asymmetric loss [1]) while *Negative* mode [2,3,26,34] treats all unknown labels as negative labels. Several work (including this paper) aim to improve *Negative* mode with *Ignore* mode, as introduced in Sect. 1. Kundu *et al.* [26] proposed a method to soften the signal of the wrong labels of *Negative* mode by exploiting the image and label relationships, but it does not avoid some categories training on too many wrong labels. Ben-Brunch *et al.* [1] proposed *Selective* approach that can adjust the training mode for each category to be either *Negative* or *Ignore*, but it requires the presence frequency of every category which is unavailable in partially labeled datasets.

[1] https://stanfordmlgroup.github.io/competitions/chexpert/.

Unlike most previous approaches that aim to train high performance models beginning with initial parameters, the proposed CFT is a post-training method based on *Ignore* mode that can be applied to models trained with *Negative* mode to further improve the performance. Moreover, CFT can independently improve the classification performance on each category. Hence, CFT may be able to further improve the performance of the models trained with other approaches mentioned above.

3 Methods

This section presents the proposed CFT, the greedy selection for selecting fine-tuning configurations, and GA for fine-tuning, as summarized in Fig. 1.

Notations. Considering a C-category image MLC task with a training set $\mathcal{D} = \{(I, \mathbf{y})_i\}$. Each sample $(I, \mathbf{y})$ consists of an image I and a label vector $\mathbf{y} = [y_1, ..., y_C] \in \{-1, 1, 0\}^C$ where the c^{th} $(c \in \{1, ..., C\})$ element y_c is the label on category c and it is assigned to be either -1 (*negative*), 1 (*positive*), or 0 (*unknown*). A deep neural network (typically CNN) *Baseline* has been trained on the training set $\mathcal{D}$ with *Negative* mode. The architecture of *Baseline* consists of: (1) a backbone $\mathtt{b}$ transforms an input image I to a feature vector $\mathbf{z} = \mathtt{b}(I) \in \mathbb{R}^Z$; and (2) a C-unit fully-connected layer $\mathtt{h}$ with Sigmoid activation transforms a feature vector $\mathbf{z}$ to an output vector $\hat{\mathbf{y}} = \mathtt{h}(\mathbf{z}) = [\hat{y}_1, ..., \hat{y}_C] \in [0, 1]^C$, where the c^{th} element $\hat{y}_c$ is the output representing the binary classification result on category c. To better illustrate CFT, we equivalently regard the fully-connected layer $\mathtt{h}$ as C independent logistic regressions (LRs) $\mathtt{h}_1, ..., \mathtt{h}_C$, as shown in Fig. 1 left. The c^{th} LR $\mathtt{h}_c$ transforms a feature vector $\mathbf{z}$ to an output $\hat{y}_c = \mathtt{h}_c(\mathbf{z})$.

3.1 Category-Wise Fine-Tuning (CFT)

The proposed CFT is a post-training method that can be applied to *Baseline*. CFT uses *Ignore* mode to one-by-one fine-tune the LRs $\mathtt{h}_1, ..., \mathtt{h}_C$ to improve its performance on each category independently. Therefore, the backbone $\mathtt{b}$ is always unchanged.

Specifically, the procedure of CFT has C steps (*i.e.*, determined by the number of categories C). The goal of the c^{th} step $(c = \{1, ..., C\})$ is to independently improve the performance on category c through fine-tuning *Baseline*. That is, the fine-tuning only improves the performance on category c, meanwhile, keeping the performance on other categories unchanged. Hence, each category can be independently improved without any concerns of harming other categories.

To achieve this goal, at the c^{th} step, only the c^{th} LR $\mathtt{h}_c$ is fine-tuned instead of the whole *Baseline*. It is because changing all the parameters of *Baseline* will change the performance on all categories, which does not match the goal. On the other hand, changing the parameters of $\mathtt{h}_c$ only affects the output $\hat{y}_c$ on category c and does not affect the outputs on other categories, which matches the goal.

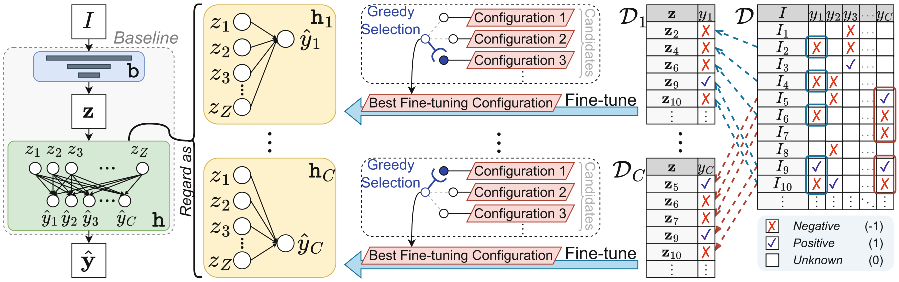

Fig. 1. The overview of CFT and the greedy selection.

At the c^{th} step, the c^{th} LR $\mathtt{h}_c$ is fine-tuned using binary crossentropy (BCE) loss with backpropagation (BP), which is popular for optimizing binary classification models. *Ignore* mode is used to reduce the performance decrease caused by the wrong labels of *Negative* mode during training. Particularly, $\mathtt{h}_c$ is fine-tuned on a new training set $\mathcal{D}_c$ generated from the original training set $\mathcal{D}$ for the use of *Ignore* mode and reducing computation cost, as shown in Fig. 1 right. We first select the samples from $\mathcal{D}$ where the label on category c is known (*i.e.*, $y_c \in \{-1, 1\}$) to be the samples in $\mathcal{D}_c$. This selection ensures $\mathtt{h}_c$ is fine-tuned with *Ignore* mode. Then, as the backbone $\mathtt{b}$ is always the same, we convert the image I of each sample to a feature vector $\mathbf{z} = \mathtt{b}(I)$ in advance to avoid unnecessary computation during fine-tuning. Lastly, the unnecessary labels on other categories are dropped. Formally, the new training set $\mathcal{D}_c = \big\{(\mathbf{z}, y_c)_i\big\}$ is generated by: $\mathcal{D}_c = \big\{\mathtt{T}((I, \mathbf{y}))\big|(I, \mathbf{y}) \in \mathcal{D}, y_c \in \{-1, 1\}\big\}$ where $\mathtt{T}((I, \mathbf{y})) = \big(\mathtt{b}(I), y_c\big) = (\mathbf{z}, y_c)$.

3.2 Greedy Selection for Fine-Tuning Configuration Selection

While applying CFT to *Baseline*, as the LRs are independent to each other, each LR can be fine-tuned with different configurations to achieve higher performance. The configurations can be different optimization methods (*e.g.*, BCE loss and the below-introduced GA), methods for handling the untypical labels that appear in the CheXpert dataset (see Sect. 4.1), batch sizes, learning rates, etc.

Hence, for each LR, we can additionally compare multiple fine-tuning configuration candidates and select the best one based on the results, referred to as *greedy selection*, as shown in Fig. 1 middle. For example, assume we apply CFT to *Baseline* that has 5 LRs $\mathtt{h}_1, ..., \mathtt{h}_5$ (5 categories). We can additionally compare BCE loss and GA, then choose the best configuration for each LR. A possible result is, $\mathtt{h}_1, \mathtt{h}_4, \mathtt{h}_5$ uses BCE loss, while $\mathtt{h}_2, \mathtt{h}_3$ uses GA.

3.3 Fine-Tuning Logistic Regressions (LRs) Using Genetic Algorithm

During the experiments on the CheXpert dataset (performance metric is AUC, higher is better), we found that fine-tuning an LR using BCE loss sometimes unwantedly decreases AUC. A concrete example is in Fig. 2 which shows the learning curves of fine-tuning the LR of the "Atelectasis" category. In both the training curves and the validation curves, minimizing BCE can cause AUC decreases. It is because minimizing BCE is generally used for optimizing classification accuracy [40], which does not necessarily achieve the best possible AUC [40] or AP (average precision) [33] that are popular metrics for image MLC.

Therefore, we propose using Genetic Algorithm (GA) [29] to fine-tune each LR. GA is a global search algorithm inspired by the principle of the evolution theory. In nature, individuals which are more adapted to the environment have higher chances to survive and produce offspring. This process keeps repeating over generations until the best individual is found.

GA has shown its feasibility for training neural networks [10,18,30] and has several advantages in comparison to BCE loss. (1) GA is a direct search method [37] that can directly improve the performance computed by a metric, which avoids the potential performance decreases caused by minimizing BCE; and (2) BCE loss relies on backpropagation which is easy to trap in local optima and difficult to escape it to find a better solution [18]. GA runs multiple solutions simultaneously, which helps to escape from local optima [37].

4 Experimental Results and Discussion

We conducted sufficient experiments on the CheXpert competition dataset (Sect. 4.1) and the partially labeled versions of the MS-COCO [28] standard MLC dataset (Sect. 4.2) to evaluate the effectiveness of the proposed methods.

4.1 The CheXpert Chest X-Ray Image MLC Competition Dataset

Dataset. CheXpert [21] is a large-scale chest X-ray image 14-category MLC competition dataset. **The training set** has 223,414 image samples. Labels are automatically extracted from the free text reports. Labels are either *positive*, *negative*, *unknown* (the term is *blank* in the original paper), or *uncertain*. Noteworthy, the uncertain labels in this dataset are untypical in partially labeled datasets and have different semantic meanings from unknown labels. An uncertain label captures both the uncertainty in diagnosis and ambiguity in the report, while an unknown label implies no mentions are found in the report. Hence, we do not simply consider the uncertain labels as unknown labels. We handle the uncertain labels in other ways instead, as described in the experimental settings below. **The validation set** has 234 image samples. A label is manually assigned as either *positive* or *negative*. **The test set** has 668 image samples. A label is manually assigned as either *positive* or *negative*. The test set is private and is reserved for the competition. Models must be submitted to the competition server for the official evaluation on the test set. The competition leaderboard is available at https://stanfordmlgroup.github.io/competitions/chexpert/. **The official performance metric** is used, which is computed by the mean AUC (mAUC) on the 5 categories: Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion.

Baseline Training. *Baseline* is a DenseNet-121 [19] CNN with an input resolution 224^2. The parameters trained on ImageNet [11] are used as the initial parameters. *Baseline* is trained on the training set for 10 epochs. We follow the previous state-of-the-art [31,44] to treat unknown labels as negative (*Negative* mode) and treat uncertain labels as positive with label smoothing [31]. Images are rescaled to $[0, 1]$. We use the same data augmentation as in [6,7]: horizontal flip, rotate ±20°, and scale ±3%. BCE loss with batch size 32 and Adam ($lr = 1 \times 10^{-4}$) [25] is used to update parameters. The checkpoint that achieves the highest validation mAUC is saved. *Baseline* achieves mAUC 89.6% on the validation set (as reported in Table 1) which is already very high for a single CNN. *E.g.*, the single CNN of 2^{nd} place on the competition leaderboard achieves mAUC 89.4% [31].

Ablation Study on CFT. We apply CFT to *Baseline* to improve its performance. The default BCE loss is used to fine-tune each LR, referred to as (CFT-BCE). Besides, we study two variants of CFT-BCE:

1. CFT-BCE-simu: All the LRs are fine-tuned **simu**ltaneously (*i.e.*, fine-tune the whole classification layer), instead of fine-tuning each LR one-by-one. Partial-BCE loss [13] is used to enable *Ignore* mode.
2. CFT-BCE-Nega: Each LR is fine-tuned with ***Nega**tive* mode, instead of *Ignore* mode.

Full-batch gradient descent ($lr = 1 \times 10^{-4}$) is used to update parameters for stability. We treat the uncertain labels as unknown labels, so the uncertain labels are ignored in CFT. The number of epochs is 500.

Table 1 shows the results. CFT-BCE and its variants successfully improve the mAUC of *Baseline*. Particularly, CFT-BCE achieves the highest improvement (mAUC +0.3%). CFT-BCE-simu is less effective (+0.1%), because one-by-one fine-tuning allows individually saving the best checkpoint for each LR, thus achieving better mAUC. CFT-BCE-Negative is also less effective (+0.1%), demonstrating the use *Ignore* Mode can effectively reduce the performance decreases caused by the wrong labels of *Negative* mode during training.

Table 1. Ablation study on CFT, AUC%.

Method	Ate	Car	Con	Ede	P.E	*Mean*
Baseline	85.5	84.2	93.3	92.7	92.3	*89.6*
CFT-BCE-simu	**85.7**	84.0	93.3	92.8	92.4	*89.7*
CFT-BCE-Nega	85.6	84.2	93.4	**92.9**	92.4	*89.7*
CFT-BCE	85.6	**85.0**	**93.5**	**92.9**	**92.5**	***89.9***

Table 2. Ablation study on GA, AUC%.

Config	Ate	Car	Con	Ede	P.E	*Mean*
CFT-BCE	85.6	85.0	93.5	92.9	92.5	*89.9*
CFT-WMW	87.2	87.9	**94.7**	92.9	92.5	*91.0*
CFT-AUCM	**89.1**	87.7	93.8	**93.2**	92.4	*91.2*
CFT-GA	88.8	**88.6**	94.5	93.0	**92.7**	***91.5***

Ablation Study on GA. We study four different optimization methods for fine-tuning LRs to investigate the effectiveness of GA: (1) the default BCE loss used above (CFT-BCE), (2) GA (CFT-GA), (3) the loss proposed in [40], referred to as WMW loss (CFT-WMW), and (4) AUC margin loss (CFT-AUCM) [44]. WMW and AUC margin losses are particularly designed for AUC maximization.

For CFT-GA, we use the GA implementation of PyGAD [14]. The number of generations is 500. An individual represents the parameters of the LR, where one position of the individual represents one parameter. Decoding is the inverse operation of encoding. The number of individuals is 30. All individuals are initialized by encoding the original parameters. The fitness function is set to be the training mAUC. Roulette wheel selection is used to select 14 individuals as parents. 10 of the parents are additionally kept as individuals in the next generation. 2-point crossover is used with a probability of 80%. Mutation probability is set to be 2%. When a mutation occurs, 1% of the positions are mutated by adding random scalars drawn from $[-0.02, 0.02]$. The individual that attains the highest fitness score at every generation is validated instead of all individuals to reduce the risk of overfitting. The individual that achieves the highest validation mAUC is decoded and saved. For CFT-WMW, stochastic gradient descent ($lr = 1 \times 10^{-3}, momentum = 0.9$) with batch size 32768 is used due to memory lack. For CFT-AUCM , we follow the original paper [44] to use PESG ($lr = 1 \times 10^{-2}, margin = 1$) [16]. Full batch size is used.

Table 2 shows all methods successfully improve the AUCs on all 5 categories. Particularly, GA is the most effective (mAUC +1.9%), followed by AUCM loss (+1.6%), WMW loss (+1.4%). BCE loss is the least effective (+0.3%).

Although WMW and AUCM losses are designed for AUC maximization, they are less effective than GA, probably they rely on backpropagation which is easy to trap on local optima. On the other hand, GA can directly optimize AUC and is easier to escape from local optima. BCE loss is the least effective, as minimizing BCE can lead to AUC drops. *E.g.*, on "Atelectasis" category (Fig. 2).

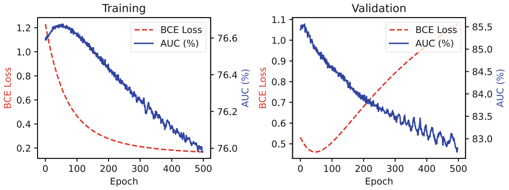

Fig. 2. Learning curves of using BCE loss to fine-tune the LR on Atelectasis. Minimizing BCE loss can decrease AUC.

Table 3. Greedy selection for exploiting uncertain labels, AUC%.

Method	Ate	Car	Con	Ede	P.E	*Mean*
Unknown	**88.8**	**88.6**	94.5	93.0	92.7	*91.5*
Positive	88.6	87.9	93.8	**93.1**	**92.8**	*91.3*
Negative	85.5	88.2	**95.6**	93.0	92.4	*90.9*
Greedy	**88.8**	**88.6**	**95.6**	**93.1**	**92.8**	***91.8***

Greedy Selection for Exploiting Uncertain Labels. In the above ablation studies, treating uncertain labels as unknown may be sub-optimal, as previous studies in this dataset show that treating uncertain labels as positive tends to achieve higher performance [31]. Therefore, we compare three methods for handling uncertain labels with CFT-GA: treat as unknown labels (same as in ablation studies), positive labels [21], and negative labels [21].

Table 3 shows that different categories prefer different methods. Hence, we use the greedy selection to select the best method for each LR, eventually achieving mAUC 91.8%, which is +2.2% higher than *Baseline* mAUC 89.6%. In the following comparison section, we refer to this model as *CFT-GA-Greedy*.

Table 4. Comparison to other state-of-the-art approaches on the **test** set, AUC%.

Model Type	Rank	Approach	Ate	Car	Con	Ede	P.E	*Mean*
Single Model	147	Chong *et al.* [7]	85.67	89.30	82.15	90.92	95.56	*88.72*
	151	Multiview (R-50) [22]	85.60	90.85	81.07	89.45	95.85	*88.60*
	134	Multiview (D-121) [22]	86.49	**90.95**	83.99	89.62	**96.34**	*89.50*
	127	PTRN + Single Model [6]	85.66	89.06	86.89	90.94	95.47	*89.61*
	53	**CFT-GA-Greedy**	**88.58**	90.20	**90.99**	**93.06**	96.26	***91.82***
Ensemble	102	PTRN + Ensemble [6]	85.73	89.90	90.57	91.66	95.04	*90.58*
	98	Stanford Baseline [21]	85.50	89.77	89.76	91.56	96.67	*90.65*
	5	YWW [42]	88.18	**93.96**	**93.43**	92.72	96.15	*92.89*
	2	Hierarchical Learning [31]	90.13	93.18	92.11	92.89	**96.68**	*93.00*
	1	DAM [44]	88.65	93.72	93.21	93.00	96.64	*93.05*
	-	**CFT-GA-Greedy-Ensemble**	**91.52**	93.73	91.57	**93.33**	96.50	***93.33***

Comparison to State-of-the-art Approaches. We compare CFT-GA-Greedy to other state-of-the-art approaches on the **test** set. Most approaches treat unknown labels as negative labels, hence can be considered as strong baselines of *Negative* mode for the comparison. Table 4 shows the comparison.

Single Model. We submitted CFT-GA-Greedy to the competition server for official evaluation. It achieves mAUC 91.82% which is the highest single model AUC in the leaderboard and literature, to the best of our knowledge.

Ensemble. We build an ensemble composed of CFT-GA-Greedy and another 4 CNNs developed by our proposed methods, referred to as *CFT-GA-Greedy-Ensemble*. Similar to 2$^{\text{nd}}$ on the competition leaderboard [31], we use test time augmentation [36] for more robust predictions: scale ±5%, rotate ±5°, translate ±5°. Since the competition was suddenly closed, our ensemble cannot be submitted for the official evaluation. After the test set was released and can be downloaded, we evaluate our ensemble on a local machine. Our ensemble achieves mAUC 93.33% which superiors the best in the leaderboard and literature, to the best of our knowledge.

4.2 Partially Labeled Versions of MS-COCO

Dataset. MS-COCO [28] (2014 split) is a standard MLC dataset comprising 80 categories. The training and the validation sets consist of around 80k and 40k image samples, respectively. We follow the work on MS-COCO (*e.g.*, [34]) to use mean AP (mAP) as the performance metric.

As the training data is fully labeled, different schemes of partial labels can be simulated by dropping some labels. Particularly, we study our methods under the proportions of known labels of 10%, 20%, ..., 90%, respectively. To simulate these schemes, we randomly drop 90%, 80%, ..., 10% of labels, respectively.

Table 5. Results on **partially labeled versions** of MS-COCO dataset. In mAP %. "Average" column is the average mAP over label proportions 10% to 90%. (**Bolded** is the best, underlined is the 2$^{\text{nd}}$ best)

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	*Average*
Baseline	54.8	63.1	68.9	72.0	74.1	75.9	77.9	79.2	80.6	71.84
CFT-BCE-simu	54.7	63.1	68.9	73.0	74.9	76.7	78.4	79.6	80.6	72.20
CFT-BCE-Negative	56.6	65.3	70.4	73.4	75.3	77.0	78.8	80.0	81.3	73.11
CFT-BCE	59.3	67.7	**72.6**	75.0	76.6	78.2	79.6	80.7	81.6	74.58
CFT-GA	57.4	65.6	71.0	73.8	75.6	77.4	79.1	80.4	81.4	73.52
CFT-Greedy	**59.3**	**67.7**	**72.6**	**75.0**	**76.6**	**78.2**	**79.7**	**80.8**	**81.7**	**74.61**

Baseline Training. We follow most of the settings of [34] to train *Baseline*, as they achieved state-of-the-art CNN on the original MS-COCO (*i.e.*, fully labeled). *Baseline* is a TResNet-L [35] with an input resolution 448^2 . The parameters trained on ImageNet are used as the initial parameters. *Negative* mode is used to handle the unknown labels. We use batch size 8, asymmetric loss [34], and Adam ($lr = 2 \times 10^{-4}$) to update the parameters. We use AutoAugment [9] with pretrained ImageNet policy as the data augmentation method. Normalization of mean 0 and variance 1 is applied to the input images. The checkpoint that achieves the highest validation mAP is saved. The performance of *Baseline* under different label proportions are reported in Table 5.

Ablation Study on CFT. We apply CFT to *Baseline* to improve its performance. The default BCE loss is used to fine-tune each LR, referred to as *CFT-BCE*. Similar to the experiments on CheXpert, we also study the two variants of CFT-BCE: CFT-BCE-simu and CFT-BCE-Negative. Full-batch gradient descent ($lr = 1 \times 10^{-2}, monentum = 0.9$) is used and the number of epochs is 5000.

CFT-BCE improves the average mAP by 2.74%, CFT-BCE-simu improves 0.36%, and CFT-BCE-Negative improves 1.27%. Both variants are less effective than CFT-BCE, demonstrating the effectiveness of one-by-one fine-tuning and *Ignore* mode.

Noteworthy, CFT-BCE-Negative does not use *Ignore* mode. Although it is less effective than using *Ignore* mode, it still can improve the average mAP. It implies that this improvement is likely to be gained from somewhere else instead of from reducing the performance decreases caused by the wrong labels of *Negative* mode during training. Therefore, CFT may be able to improve models trained with fully labeled data, which requires further investigation.

Ablation Study on GA. We compare GA to the default BCE loss (used in above) for fine-tuning each LR. The number of generations is 2000. The population size is 50. All the individuals of the initial population are encoded from the original parameters. The best individual of the current generation is selected as one individual of the next generation. The parents are selected using roulette wheel selection. During crossover, 20% of the positions of two parents are randomly switched to produce offspring. Each offspring has a 50% chance of being mutated by adding a random scalar between $[-0.001, 0.001]$ to each position.

GA improves the average mAP by 1.68%. However, it is generally less effective than BCE loss (2.74%). The key reasons may be (1) minimizing BCE does not necessarily lead to AP drops, and (2) BCE loss relies on backpropagation which is generally more efficient than GA.

Greedy Selection. We use greedy selection for choosing the best optimization methods between BCE loss and GA for each LR, referred to as *CFT-Greedy*. CFT-Greedy improves the average mAP by 2.77%, which is further higher than CFT-BCE by 0.03%. It implies that the greedy selection has chosen GA for the fine-tuning of a small proportion of LRs.

5 Conclusion

In this paper, we propose a new post-training method called CFT which one-by-one fine-tunes the LRs in a model trained with *Negative* mode to improve its classification performance of each category independently further. Two optimization methods (BCE loss and GA) are tested for fine-tuning LRs. The effectiveness is evaluated on the CheXpert competition dataset and the partially labeled versions of the MS-COCO standard MLC dataset. Especially, CFT achieves state-of-the-art on the CheXpert dataset (single model AUC 91.82% and ensemble AUC 93.33%, on the test set).

Acknowledgements. This work is supported by Macao Polytechnic University under grant number RP/ESCA-01/2021.

References

1. Ben-Baruch, E., et al.: Multi-label classification with partial annotations using class-aware selective loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4764–4772 (2022)

2. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: CVPR 2011, pp. 2801–2808. IEEE (2011)
3. Chen, M., Zheng, A., Weinberger, K.: Fast image tagging. In: International Conference on Machine Learning, pp. 1274–1282. PMLR (2013)
4. Chen, T., Pu, T., Liu, L., Shi, Y., Yang, Z., Lin, L.: Heterogeneous semantic transfer for multi-label recognition with partial labels. arXiv preprint arXiv:2205.11131 (2022)
5. Chen, T., Pu, T., Wu, H., Xie, Y., Lin, L.: Structured semantic transfer for multi-label recognition with partial labels. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, No. 1, pp. 339–346 (2022)
6. Chong, C.F., Wang, Y., Ng, B., Luo, W., Yang, X.: Image projective transformation rectification with synthetic data for smartphone-captured chest X-ray photos classification. Comput. Biol. Med. **164**, 107277 (2023)
7. Chong, C.F., Yang, X., Ke, W., Wang, Y.: GAN-based Spatial transformation adversarial method for disease classification on CXR photographs by smartphones. In: 2021 Digital Image Computing: Techniques and Applications (DICTA), pp. 01–08. IEEE (2021)
8. Chu, H.-M., Yeh, C.-K., Wang, Y.-C.F.: Deep generative models for weakly-supervised multi-label classification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 409–425. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_25
9. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113–123 (2019)
10. David, O.E., Greental, I.: Genetic algorithms for evolving deep neural networks. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 1451–1452 (2014)
11. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
12. Deng, J., Russakovsky, O., Krause, J., Bernstein, M.S., Berg, A., Fei-Fei, L.: Scalable multi-label annotation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3099–3102 (2014)
13. Durand, T., Mehrasa, N., Mori, G.: Learning a deep convnet for multi-label classification with partial labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 647–657 (2019)
14. Gad, A.F.: PyGAD: an intuitive genetic algorithm python library. arXiv: 2106.06158 (2021)
15. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894 (2013)
16. Guo, Z., Yan, Y., Yuan, Z., Yang, T.: Fast objective & duality gap convergence for nonconvex-strongly-concave min-max problems. arXiv preprint arXiv:2006.06889 (2020)
17. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5356–5364 (2019)
18. Gupta, J.N., Sexton, R.S.: Comparing backpropagation with a genetic algorithm for neural network training. Omega **27**(6), 679–684 (1999)
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

20. Huynh, D., Elhamifar, E.: Interactive multi-label CNN learning with partial labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9423–9432 (2020)
21. Irvin, J., et al.: CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 590–597 (2019)
22. Jansson, P. et al.: Multi-view automated chest radiography interpretation (2021)
23. Jing, L., Yang, L., Yu, J., Ng, M.K.: Semi-supervised low-rank mapping learning for multi-label classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1483–1491 (2015)
24. Kapoor, A., Viswanathan, R., Jain, P.: Multilabel classification using bayesian compressed sensing. In: Advances In Neural Information Processing Systems 25 (2012)
25. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Kundu, K., Tighe, J.: Exploiting weakly supervised visual patterns to learn from partial annotations. Adv. Neural. Inf. Process. Syst. **33**, 561–572 (2020)
27. Kuznetsova, A., et al.: The open images dataset V4. Int. J. Comput. Vis. **128**(7), 1956–1981 (2020). https://doi.org/10.1007/s11263-020-01316-z
28. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
29. Mitchell, M.: An Introduction to Genetic Algorithms. MIT press (1998)
30. Montana, D.J., et al.: Training feedforward neural networks using genetic algorithms. In: IJCAI, vol. 89, pp. 762–767 (1989)
31. Pham, H.H., Le, T.T., Tran, D.Q., Ngo, D.T., Nguyen, H.Q.: Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels. Neurocomputing **437**, 186–194 (2021)
32. Pu, T., Chen, T., Wu, H., Lin, L.: Semantic-aware representation blending for multi-label image recognition with partial labels. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, No. 2, pp. 2091–2098 (2022)
33. Qi, Q., Luo, Y., Xu, Z., Ji, S., Yang, T.: Stochastic optimization of areas under precision-recall curves with provable convergence. Adv. Neural. Inf. Process. Syst. **34**, 1752–1765 (2021)
34. Ridnik, T., et al.: Asymmetric loss for multi-label classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 82–91 (2021)
35. Ridnik, T., Lawen, H., Noy, A., Ben Baruch, E., Sharir, G., Friedman, I.: TResNet: high performance GPU-dedicated architecture. In: proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 1400–1409 (2021)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328
38. Vasisht, D., Damianou, A., Varma, M., Kapoor, A.: Active learning for sparse bayesian multilabel classification. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 472–481 (2014)
39. Wu, B., Lyu, S., Ghanem, B.: ML-MG: multi-label learning with missing labels using a mixed graph. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4157–4165 (2015)

40. Yan, L., Dodier, R.H., Mozer, M., Wolniewicz, R.H.: Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. In: Proceedings of the 20th International Conference on Machine Learning (icml-03), pp. 848–855 (2003)
41. Yang, H., Zhou, J.T., Cai, J.: Improving multi-label learning with missing labels by structured semantic correlations. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 835–851. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_50
42. Ye, W., Yao, J., Xue, H., Li, Y.: Weakly supervised lesion localization with probabilistic-cam pooling. arXiv preprint arXiv:2005.14480 (2020)
43. Yu, H.F., Jain, P., Kar, P., Dhillon, I.: Large-scale multi-label learning with missing labels. In: International Conference on Machine Learning, pp. 593–601. PMLR (2014)
44. Yuan, Z., Yan, Y., Sonka, M., Yang, T.: Large-scale robust deep AUC maximization: A new surrogate loss and empirical studies on medical image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3040–3049 (2021)