G^{2}NetPL: Generic Game-Theoretic Network for Partial-Label Image Classification

Rabab Abdelfattah[1]
rabab@email.sc.edu

Xin Zhang[1]
xz8@email.sc.edu

Mostafa M. Fouda[2]
mfouda@ieee.org

Xiaofeng Wang[1]
wangxi@cec.sc.edu

Song Wang[3]
songwang@cec.sc.edu

[1] Department of Electrical Engineering
University of South Carolina
South Carolina, USA

[2] Department of Electrical and Computer
Engineering
Idaho State University
Idaho, USA

[3] Department of Computer Science
University of South Carolina
South Carolina, USA

Abstract

Multi-label image classification aims to predict all possible labels in an image. It is usually formulated as a partial-label learning problem, since it could be expensive in practice to annotate all the labels in every training image. Existing works on partial-label learning focus on the case where each training image is labeled with only a subset of its positive/negative labels. To effectively address partial-label classification, this paper proposes an end-to-end Generic Game-theoretic Network (G^{2}NetPL) for partial-label learning, which can be applied to most partial-label settings, including a very challenging, but annotation-efficient case where only a subset of the training images are labeled, each with only one positive label, while the rest of the training images remain unlabeled. In G^{2}NetPL, each unobserved label is associated with a soft pseudo label, which, together with the network, formulates a two-player non-zero-sum non-cooperative game. The objective of the network is to minimize the loss function with given pseudo labels, while the pseudo labels will seek convergence to 1 (positive) or 0 (negative) with a penalty of deviating from the predicted labels determined by the network. In addition, we introduce a confidence-aware scheduler into the loss of the network to adaptively perform easy-to-hard learning for different labels. Extensive experiments demonstrate that our proposed G^{2}NetPL outperforms many state-of-the-art multi-label classification methods under various partial-label settings on three different datasets.

1 Introduction

Based on deep-learning techniques, significant progress has been made on single-label image classification [10] where each image only has one label. However, in many real applications, one image may contain multiple objects and/or exhibit multiple attributes which cannot be well described by a single label. This leads to an important computer-vision task of multi-label image classification that aims to identify all the labels of an image. One main challenge in deep-learning based multi-label image classification comes from the requirement of large-scale labeled training images. In particular, many supervised-learning algorithms require all

the labels present in each training image to be accurately annotated and this full-label manual annotation can be very difficult and laborious [11].

To relieve the annotation burden of full labeling, several recent works on multi-label classification consider the use of training image data with partial labels [8, 12, 16, 17, 22, 32, 46], where only a limited number of labels are annotated on each image. A special case is the full-set-single-positive-label (FSPL) setting, which requires to annotate only one positive label for each training image [8]. While these partial-label settings can mitigate the annotation burden, one still needs to label all training images. To go one step further, the subset-single-positive-label (SSPL) setting only requires to label a subset of the training images and for each labeled image, only one positive label is annotated [1]. For example, each image in the iNaturalist dataset only has one positive label with other classes unlabeled [8]. Adding new unlabeled images to enrich this dataset leads to SSPL, which is considered a practical scenario. Actually, this applies to most datasets for multi-class classification. Adding new unlabeled images makes those datasets suitable for the multi-label problem under SSPL without annotation costs.

Clearly, this SSPL setting can further reduce the annotation burden for large-scale datasets. However, the inclusion of unlabeled images raises a great challenge in network design and training. On one hand, those existing partial-label learning methods based on label correlation [16] and label matrix completion [4] are not applicable to SSPL since they cannot handle single positive labels and unlabeled data simultaneously. On the other hand, although semi-supervised learning can deal with unlabeled images, it is usually designed based on the assumption of having a subset of fully-labeled training images, but not specifically for SSPL setting that does not offer such a subset. Therefore, the performance of the existing semi-supervised models on SSPL setting may be degraded.

This paper presents a new generic game-theoretic network (G^{2}NetPL) for end-to-end training of the multi-label classifier under SSPL setting. In G^{2}NetPL, each unobserved label is associated with a soft pseudo label, which acts as a player in a two-player non-zero-sum non-cooperative game with the objective of converging to 1 (positive) or 0 (negative), given a penalty of deviating from the predicted labels determined by the network. As the second player, the network is to minimize a weighted loss function with given pseudo labels. Our contributions are summarized as follows:

- We propose a novel G^{2}NetPL for partial-label image classification, focusing on SSPL setting with the understanding that G^{2}NetPL can be applied to most, if not all, partial-label settings. Within this game-theoretic framework, the two players, the network and the pseudo labels, are updated with different objectives. Accordingly, different loss functions are developed for players. Therefore, the network can, from a game-theoretic point of view, achieve robustness with respect to errors in pseudo label estimation. Meantime, the existence of Nash equilibrium will guarantee convergence in training.

- We introduce the confidence level of pseudo labels into our loss function, where less weights will be considered for those pseudo labels with low confidence levels. This is critical, especially at the early stage of training when the observed labels are lack and the confidence levels on the pseudo labels are low. Otherwise, premature convergence could be achieved with low confidence levels due to lack of observed labels.

- Extensive experimental results show that G^{2}NetPL outperforms the state-of-the-arts under different partial-label settings on three widely-used datasets. With fewer observed labels in SSPL setting, our method can still get comparable classification results as those methods using FSPL setting.

2 Related Work

This section briefly reviews previous works for partial multi-label image classification.

Partial Labels. To address the unobserved labels, which can be positives or negatives, several works simply tend to regard unobserved labels as negative [3, 5, 27, 36]. This assumption usually leads to a significant performance drop by incorrectly initializing the positive labels as negatives [18]. Others tend to find the unobserved labels via label correlations [27, 45], label matrix completion [4, 41], low-rank learning [47, 48], and probabilistic models [6, 21]. Most of these methods do not allow end-to-end training. Recently, several end-to-end models are proposed for the partial-label setting [8, 12, 16]. In [16], image and label similarity graphs are built to estimate the unobserved labels. The construction of label graphs relies on the label co-occurrence information in the training data. This method is not applicable to single-positive-label (SPL) settings which has at most one observed label per image. Durand *et al*. [12] adopt both the graph neural network and curriculum learning to find the relations between the labels and complete the unobserved labels. This approach depends on a fixed threshold to add "easy-unobserved" labels to the loss function. Once the network is updated, those previously predicted unobserved labels will be discarded, which makes this approach heuristic without continuity in learning. Different from that, G^2NetPL can forecast convergence and optimality. In addition, our model will use every unobserved labels in training, while in [12] there is a possibility that some unobserved labels are never selected before the model converges, due to the threshold-based strategy.

Semi-supervised learning, on the other hand, is a widely used technique for leveraging a large unlabeled dataset alongside a small fully-labeled subset [2, 25, 30, 33, 35, 40]. As mentioned in [33], however, most semi-supervised models focus only on single-label classification problem. To fit into multi-label classification, modifications are necessary. Other related works exploit the label ranking [20] and label correlations [57] to learn from positive and unlabeled data. However, these methods require more than one positive label per image so that pairwise label dependencies can be utilized.

Full-set Single Positive Label. The FSPL setting was proposed by Cole *et al*. [8] most recently, where only a single positive label is observed on each training image. This model jointly trains the image classifier and the label estimator using the online label estimation. However, the online estimation requires to store the full labels of all the training data in the memory [8]. In addition, the label estimator is randomly initialized which leads to inferior model in end-to-end setting and the observed/unobserved labels are addressed with equal weights. Different from that, G^2NetPL defines a game between the network and the pseudo labels, each of which has its own goal to achieve. Moreover, the introduction of the confidence level places different weights for unobserved labels in the loss function of the network, which can gradually move the learning from observed labels to unobserved labels without being over-confident. Moreover, G^2NetPL learns pseudo labels for images in each mini-batch without requiring keeping all training data in the memory.

Subset Single Positive Label. To the best of our knowledge, there is no work explicitly exploring SSPL in-depth for multi-label image classification. As mentioned above, with necessary modifications, some semi-supervised learning algorithms, such as Fixmatch [35] and UPS [33], can work under this setting. In the experiments, we compare the performance of our proposed G^2NetPL with these semi-supervised methods under SSPL settings.
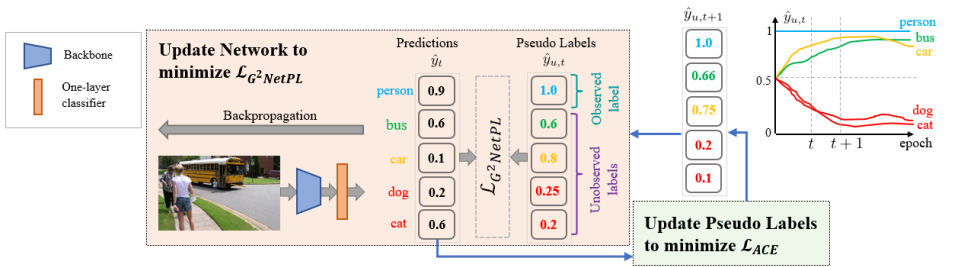
Figure 1: An overview of G^{2}NetPL that consists of two players: the network and the pseudo labels. The network includes backbone and one-layer classifier to generate a prediction for an input image. The loss function for the network is $\mathcal{L}_{G^2NetPL}$ that is a function of the network parameters θ_t with given pseudo labels. The pseudo labels $\hat{y}_{u,t}$ are iterated to minimize the cross-entropy loss function $\mathcal{L}_{ACE}$ between the pseudo labels and the predicted labels.

3   The G^{2}NetPL Framework

Notations. Let $\mathcal{I}$ be the set of all training images and $\mathcal{D}$ be the subset of $\mathcal{I}$, of which each element has a single positive label. The images in $\mathcal{I} \setminus \mathcal{D}$ remain unlabeled. Let y_o^i be the L-dimension label vector associated with image $i \in \mathcal{D}$, where L is the number of the total classes. The jth entry of y_o^i can be 1 or $\emptyset$, which means that the jth class is observed positive or unlabeled, respectively. Therefore, $\mathcal{D}$ provides the ground truth characterized in y_o^i. We denote the classifier by $f(i, \theta_t)$ that maps image i into a predicted label vector $\hat{y}_t^i \in [0,1]^L$, i.e., $\hat{y}_t^i = f(i, \theta_t)$. Here θ_t is the network parameters obtained at epoch t during training. To characterize the unobserved labels during training, we take advantage of soft pseudo labels [39] and use $\hat{y}_{u,t}^i \in [0,1]^L$ to denote the soft pseudo label vector of image $i \in \mathcal{I}$ at epoch t. In the following discussion, we will drop the index i for notational simplicity, if it is clear in context. The cross-entropy loss function between two scalars $p, q \in [0,1]$ is defined as $\mathcal{L}(p,q) = -p\log(q) - (1-p)\log(1-q)$. Given $v \in \mathbb{R}^L$, $[v]_j$ denotes the jth entry of v.

3.1   Overview

The G^{2}NetPL is shown in Fig. 1, which introduces a two-player non-zero-sum non-cooperative game between the network and the pseudo labels. The 2-player game we employ contains three basic elements of a game: a set of players (pseudo labels and network), a set of strategies or actions for players (y_u for pseudo labels and θ for the network), and the associated costs ($\mathcal{L}_{ACE}$ and $\mathcal{L}_{G^2NetPL}$). In our game, the payoff of each player is determined by the strategies of both players. The objective of the network is to minimize the loss function $\mathcal{L}_{G^2NetPL}$ (y_u affects $\mathcal{L}_{G^2NetPL}$ through the scheduler ξ), while the goal of the pseudo labels is to converge to 1 or 0 with a penalty of deviating from the predicted labels generated by the network (θ affects $\mathcal{L}_{CE}$ through the predicted labels). The Nash equilibrium makes both players have no incentive to change its current strategy. During training, the pseudo label vector $\hat{y}_{u,0}$ of each image in $\mathcal{I}$ will be initialized with 1 for the observed labels and the unbiased probability 0.5 for the unobserved labels, which means equal distance to 1 (positive) and 0 (negative). Based on received $\hat{y}_{u,t}$, the network updates its parameter θ_t which minimizes the loss function $\mathcal{L}_{G^2NetPL}$. Once the θ_t is determined, the pseudo label $\hat{y}_{u,t+1}$ will be iterated to minimize cross-entropy loss function $\mathcal{L}_{ACE}$, given $\hat{y}_t$ predicted by the network with θ_t. This process repeats until both network parameters and the pseudo labels converge to Nash equilibrium, which provides robustness in learning. During testing, only the backbone and the classifier will be used to predict the labels.

3.2 The Loss of Pseudo Labels

As mentioned in context, the objective of the pseudo labels is to converge to the true 1 or 0 with the predicted labels from the network as a reference. A penalty will be placed when the pseudo label $\hat{y}_{u,t}$ deviates from the predicted label $\hat{y}_t$. To present the loss function $\mathcal{L}_{\text{ACE}}$, we first introduce the pseudo label latent parameter $y_{u,t} \in \mathbb{R}^L$ and a mapping function $F : \mathbb{R} \to [0,1]$ such that

$$[\hat{y}_{u,t}]_j = F([y_{u,t}]_j), \; j = 1,2,\cdots,L. \tag{1}$$

The pseudo label latent parameter $y_{u,t}$ is actually a project of $\hat{y}_{u,t}$ on $\mathbb{R}^L$. Inversely, the function $F(\cdot)$ regulates $[y_{u,t}]_j$ into $[0,1]$. We define the augmented cross-entropy loss $\mathcal{L}_{\text{ACE}}$ based on $y_{u,t}$, adding a penalty on $[\hat{y}_{u,t}]_j$ when deviating from 0 and 1. For instance, the loss function to be minimized can be formulated as

$$\mathcal{L}_{\text{ACE}}(\hat{y}_t, y_{u,t}) = \sum_{j=1}^{L} [\mathcal{L}([\hat{y}_t]_j, F([y_{u,t}]_j)) + \lambda_j F([y_{u,t}]_j)(1 - F([y_{u,t}]_j))] \tag{2}$$

where λ_j is a positive weighting constant. Notice that the function $F(\cdot)$ must be a function which outputs 1 when $[y_{u,t}]_j$ approaches $+\infty$ and 0 when $[y_{u,t}]_j$ approaches $-\infty$. If F is smooth, then based on the chain rule in derivative,

$$\nabla\mathcal{L}_{\text{ACE}[y_{u,t}]_j}(\hat{y}_t, y_{u,t}) = \left(\frac{[\hat{y}_{u,t}]_j - [\hat{y}_t]_j}{[\hat{y}_{u,t}]_j(1 - [\hat{y}_{u,t}]_j)} + \lambda_j - 2\lambda_j[\hat{y}_{u,t}]_j \right) \cdot F'([y_{u,t}]_j). \tag{3}$$

Letting $\nabla\mathcal{L}_{\text{ACE}[y_{u,t}]_j}(\hat{y}_t, y_{u,t}) = 0$ indicates that the optimal $[y_{u,t}]_j$ will satisfy either $F'([y_{u,t}]_j) = 0$ or $\frac{[\hat{y}_{u,t}]_j - [\hat{y}_t]_j}{[\hat{y}_{u,t}]_j(1 - [\hat{y}_{u,t}]_j)} + \lambda_j - 2\lambda_j[\hat{y}_{u,t}]_j = 0$. If we can obtain an explicit solution on $[\hat{y}_{u,t}]_j$ in the former equation, then the computation cost can be dramatically saved. Meantime, the choice of λ_j can be time-varying depending on $[\hat{y}_t]_j$, i.e. $\lambda_j = \lambda([\hat{y}_t]_j)$. If $[\hat{y}_t]_j$ is close to 0.5, which means unreliable, λ_j can be large; otherwise, it can be small, such as the curves indicating in Fig. 2(a). The choice of $F(\cdot)$ is not unique and will affect the performance of the classifier. For instance, one candidate is the sigmoid function. Another interesting option is the cumulative distribution function (CDF) of a Gaussian distribution with the mean at 0.5, where the associated probability density function is

$$F'([y_{u,t}]_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{[y_{u,t}]_j - 0.5}{\sigma}\right)^2}, \tag{4}$$

Figure 2: (a) The probability density function of different Gaussian distributions; (b) The scheduler $\xi([\hat{y}_{u,t}]_j, \varphi_t)$.

as shown in Fig. 2(a). In this case, $F'([y_{u,t}]_j) = 0$ indicates that $[y_{u,t}]_j = \pm\infty$, which are the generalized roots to $\nabla\mathcal{L}_{\text{ACE}y_{u,t}}$, corresponding to $[\hat{y}_{u,t}]_j = 1$ (positive) or 0 (negative). The parameter σ is the standard deviation of this Gaussian distribution. It determines how fast the probability increases from 0 to 1. For instance, when $[y_{u,t}]_j = 0.5$, $F([y_{u,t}]_j)$ achieves its maximal increasing rate, $F'(0.5) = \frac{1}{\sigma\sqrt{2\pi}}$. The smaller σ is, the faster $F([y_{u,t}]_j)$ changes from 0 to 1. In an extreme case when $\sigma = 0$, $F'(0.5) = +\infty$, then F will be degraded to a step function with 0.5 delay, which implies hard pseudo labels, i.e. $[\hat{y}_{u,t}]_j = 1$ if $[y_{u,t}]_j \geq 0.5$ and $[\hat{y}_{u,t}]_j = 0$ if $[y_{u,t}]_j < 0.5$. Since using hard pseudo labels may lose smoothness of $\mathcal{L}_{\text{ACE}}$
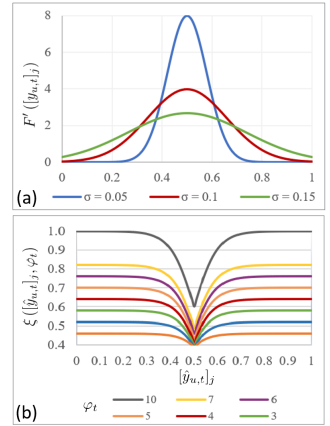
and make the results over-confident [38], it suggests that σ cannot be extremely small. On the other hand, σ cannot be too large. Otherwise, $F'([y_{u,t}]_j)$ will be small. It means that $F([y_{u,t}]_j)$ is not anxious to push $[y_{u,t}]_j$ to 1 or 0 at all, which could be contracted with the objective of the pseudo labels that seek convergence to 1 or 0. Finally, the choice of the augmented loss function is not unique either. For instance, another candidate can be

$$\mathcal{L}_{\text{ACE}}(\hat{y}_t, y_{u,t}) = \sum_{j=1}^{L} e^{\lambda_j F([y_{u,t}]_j)(1-F([y_{u,t}]_j))} \mathcal{L}\left([\hat{y}_t]_j, F([y_{u,t}]_j)\right), \tag{5}$$

which will amplify the cross-entropy loss when $[y_{u,t}]_j$ is far away from 0 or 1. When the loss function becomes complicated, numerical methods are necessary for the solutions.

3.3   Scheduled Loss of the Network

To learn the parameters of the primary multi-label CNN, the following loss function must be minimized

$$\mathcal{L}_{\text{G}^2\text{NetPL}} = \mathcal{L}_{\text{obs}} + \mathcal{L}_{\text{unobs}}. \tag{6}$$

$\mathcal{L}_{\text{obs}}$ is the binary cross-entropy classification loss between the observed labels and the associated predicted labels plus the regularizer that implicitly penalizes false positives (more details on the regularizer can be found in [8]). $\mathcal{L}_{\text{unobs}}$ is the weighted cross-entropy classification loss between the pseudo labels for the unobserved labels and their associated predicted labels. Let $\mathcal{U}_i$ be the set of the unobserved labels in image i. Then $\mathcal{L}_{\text{unobs}}$ is defined as

$$\mathcal{L}_{\text{unobs}} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{U}_i} \xi([\hat{y}_{u,t}]_j, \varphi_t) \mathcal{L}([\hat{y}_t]_j, [\hat{y}_{u,t}]_j) \tag{7}$$

where $\xi([\hat{y}_{u,t}]_j, \varphi_t)$ is the confidence-aware scheduler that controls the metric learning loss and φ_t is the current epoch t divided by the total number of epochs (representing a linear scheduler [43]). In $\mathcal{L}_{\text{unobs}}$, we exploit the soft pseudo labels instead of sharp 1 or 0 since the latter may push the predictions to be over-confident [38]. In contrast, using the pseudo labels as the distribution scores may inherit the idea of label-smoothing regularization [38]. The purpose of introducing the scheduler is to weight the unobserved loss. Different timed schedulers have been developed such as dropout training [28], transfer learning [44], and self-correction [23], in which cases the weighted score only relies on the iteration steps (or epochs). Practically, if the scheduler depends on both the confidence and the iteration steps, it may lead to improved performance. For instance, when the pseudo label $[\hat{y}_{u,t}]_j$ has low confidence, the related loss $\mathcal{L}([\hat{y}_t]_j, [\hat{y}_{u,t}]_j)$ may deviate far away from the actual loss between the predicted label and the true label (although unknown). Therefore, the scheduler $\xi([\hat{y}_{u,t}]_j, \varphi_t)$ must be small to reduce the impact of such deviation on the total loss $\mathcal{L}_{\text{G}^2\text{NetPL}}$. On the contrary, if $[\hat{y}_{u,t}]_j$ has high confidence, $\xi([\hat{y}_{u,t}]_j, \varphi_t)$ should be large. The pseudo labels will gradually build up their confidence during iterations. Following this idea, we introduce a new confidence-aware scheduler as follows, inspired by [42]:

$$\xi([\hat{y}_{u,t}]_j, \varphi_t) = \beta \frac{1 - \gamma e^{-10|2[\hat{y}_{u,t}]_j - 1|}}{1 + \gamma e^{-10|2[\hat{y}_{u,t}]_j - 1|}} + (1 - \beta)\varphi_t, \tag{8}$$

where β, γ are the positive hyper-parameters. Notice that the confidence of a pseudo label is reflected by the term $|2\hat{y}_{u,t} - 1|$. For instance, if $[\hat{y}_{u,t}]_j$ is 0.5, the related ξ will achieve its minimum, which means that the network is not confident about the pseudo label and the related loss should contribute less in $\mathcal{L}_{\text{G}^2\text{NetPL}}$. Otherwise, if $[\hat{y}_{u,t}]_j = 0$ or $[\hat{y}_{u,t}]_j = 1$, the related

ξ reaches its maximum, which indicates high confidence on the current pseudo label and the associated loss should contribute more. In Fig. 2, the scheduler has lower weighted scores over the interval $0.3 \leq [\hat{y}_{u,t}]_j \leq 0.7$, because it represents the region with lower confidence and the results generated over this region may potentially degrade the performance.

Algorithm 1 G^{2}NetPL Training

Require: input image $i \in \mathcal{I}$;
Require: observed label y_o;
Require: neural network $f(i, \theta)$ with parameters θ and input i.
Require: $[\hat{y}_u]_j = 1$ or 0, if labeled; $[\hat{y}_u]_j = 0.5$, if unlabeled
 1: **Repeat**
 2:    $\xi \longleftarrow \beta \frac{1 - \gamma e^{-10|2[\hat{y}_u]_j - 1|}}{1 + \gamma e^{-10|2[\hat{y}_u]_j - 1|}} + (1 - \beta)\varphi$;
 3:    $\hat{y} \longleftarrow f(i, \theta)$;
 4:    $\theta \longleftarrow \arg\min_\theta \mathcal{L}_{\text{G}^2\text{NetPL}}$ (through back-propagation);
 5:    $y_u \longleftarrow \arg\min_{y_u} \mathcal{L}_{\text{ACE}}(\hat{y}, y_u)$;
 6:    $[\hat{y}_u]_j \longleftarrow F([y_u]_j)$;
 7: **until** the max iteration or convergence;
 8: **Output:** θ, pseudo labels $\hat{y}_u$, and predicted labels $\hat{y}$.

Existence and Convergence of Nash equilibrium. According to the Debreu-Glicksberg-Fan results [9, 14], if every player's strategy set is compact and convex, and the payoff functions are all continuous and concave (or convex if minimizing the cost), then a pure strategy Nash equilibrium exists. In our case, the pseudo label $\hat{y}_u$ stays inside $[0, 1]$ which is compact and convex. The related loss function $\mathcal{L}_{\text{ACE}}$ is convex w.r.t. $\hat{y}_u$. Meantime, ResNet has been shown to be (near-)convex in [31]. Therefore, as long as the network parameters stay in a compact and convex set (although this set could be very large), Nash equilibrium will exist and convergence can be achieved using our proposed algorithm based on fixed-point theorem [19].

Advantages of our pseudo labels compared to existing methods. *(i)* Our approach can ensure smoothness and convergence of pseudo labels during training, which are important quality indices, while most existing methods (e.g., FixMatch [55] and UPS [53]) use threshold-based strategies that lack the continuity and guarantee of convergence in pseudo label updating. *(ii)* Our pseudo labels, as a player in the game, have their own goal, independent of the network loss, which provide a second opinion on the quality of the network training to avoid over-fitting, while in most existing work, pseudo labels either do not have a clear goal or share the same loss as the network. *(iii)* All pseudo labels in our model will contribute to the network loss except weight differences, while in threshold-based strategies, there is a chance that some pseudo labels are never selected for network training, which may result in information loss.

4   Experiments

Datasets. The **PASCAL VOC** [13] consists of 20 classes in 5,717 training images and the results are reported on the official validation set (5,823 images). **MS-COCO** [24] consists of 80 classes, including 82,081 training images and 40,137 testing images. The **NUS-WIDE** [2] contains 81 classes and it is split into 150K for training and 60.2K for testing.

Implementation Details. We use the same backbone architecture and the same classifier, ResNet-50 [15] pre-trained on ImageNet [54], for all comparisons. The training is performed in the end-to-end setting, where the parameters of the backbone and the classifier are updated

and trained for 10 epochs [8]. The learning rate is used, varying from 10^{-2} to 10^{-5}, with the batch size in 8, 16. The best mean average precision (mAP) on the validation set is reported.

4.1 Full-set Single Positive Label (FSPL)

We first conduct experiments with FSPL setting which has 100% of the training images annotated with only one single positive label. To do so, we follow [8] by assigning randomly only one positive label for each image in the training set with the guarantee that each class has at least one image labeled.

Table 1: Quantitative results (mAP) of multi-label image classification on four different datasets. Bold represents the highest mAP and underline represents the second-best among FSPL setting (Single positive and No negative).

Losses	Observed		End-to-End Setting		
	Positive	Negative	VOC	COCO	NUS
$\mathcal{L}_{BCE}$[29]	All	All	89.1	75.5	52.6
$\mathcal{L}_{BCE-LS}$	All	All	90.0	76.8	53.5
$\mathcal{L}_{IU}$ [1]	Single	Single	83.2	59.7	42.9
$\mathcal{L}_{AN}$ [22]	Single	No	85.1	64.1	42.0
$\mathcal{L}_{AN-LS}$ [8]	Single	No	86.7	66.9	44.9
$\mathcal{L}_{WAN}$ [26]	Single	No	86.5	64.8	46.3
$\mathcal{L}_{EPR}$ [8]	Single	No	85.5	63.3	46.0
$\mathcal{L}_{ROLE}$ [8]	Single	No	87.9	66.3	43.1
$\mathcal{L}_{G^2NetPL}$ (ours)	Single	No	88.8	72.4	49.7

Baseline. Several existing multi-label approaches are discussed in [8], based on the variants of the binary cross-entropy (BCE) loss which is widely used in multi-label classification. The compared approaches in our experiments include: ignoring all unobserved labels $\mathcal{L}_{IU}$, assuming the unobserved labels are negatives $\mathcal{L}_{AN}$ [22], smoothing the former loss function $\mathcal{L}_{AN-LS}$, down-weighting the terms in the loss function related to negative labels $\mathcal{L}_{WAN}$ [26], expected positive regularization $\mathcal{L}_{EPR}$ [8], and online estimation of unobserved labels $\mathcal{L}_{ROLE}$ [8]. The evaluation under the FSPL setting is divided into three categories in Table 1. In the first category, the results are evaluated in the case where all positive and all negative labels are available ("All") with the binary cross-entropy loss function $\mathcal{L}_{BCE}$ and smoothing $\mathcal{L}_{BCE-LS}$. In the second category, the results are evaluated in the case of assigning single observed label, positive and negative ("Single"). The results in these two categories provide a reference of the learning performance when all labels, or more labels, can be observed. The third category employs a fair comparison of approaches under FSPL setting. G²NetPL runs under this category.

Discussion. As reported in Table 1, G²NetPL is trained under FSPL setting, but still achieves comparable results to $\mathcal{L}_{BCE}$ and $\mathcal{L}_{BCE-LS}$, which rely on fully observed labels on VOC, COCO, and NUS datasets. It means that G²NetPL requires much fewer labels without significantly sacrificing the precision. For instance, on the COCO dataset, G²NetPL only uses 1.25% of the total labels to train the model, while the performance is only dropped by 4.4%, compared with the baseline (second row in Table 1) that are trained with 100% of the labels. In the third category of FSPL setting, G²NetPL exceeds the highest scores from the existing methods by 0.9%, 5.5% and 3.4% over VOC, COCO, and NUS datasets, respectively.

Quality of Pseudo Labels. The quality can be evaluated by two metrics. (i) The closer a pseudo label is to 0 or 1, the more indicative it is, which means higher quality. This principle is carried by the scheduler in G²NetPL during training. (ii) If a pseudo label oscillates a lot over epochs, it may be unreliable. So the quality can be evaluated by smoothness and convergence, while this can be ensured by our model. Fig. 3 shows that our pseudo labels converge without much oscillation. We evaluate the final pseudo labels against the ground truth (GT) on Pascal under FSPL. The mAP is 93.6% for ours and 89.8% for ROLE [8], which shows our pseudo labels are closer to GT.

Table 2: Quantitative results (mAP) of multi-label image classification on different subsets of the single observed label (SSPL) setting on three different datasets with End-to-End learning settings. Bold represents the highest mAP and underline represents the second-best.

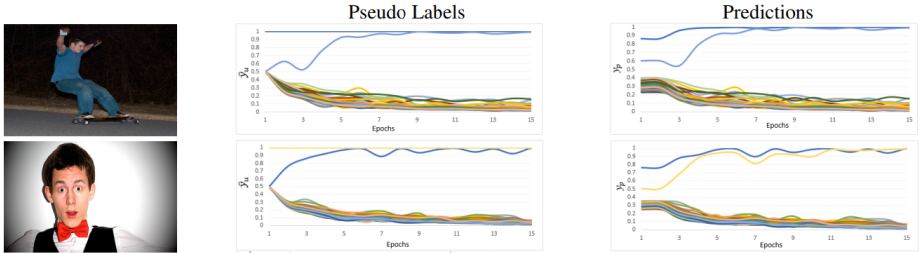| Losses	COCO				VOC				NUS			
	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%
$\mathcal{L}_{AN}$ [22]	46.3	53.8	59.5	62.4	51.3	71.7	80.2	82.8	28.5	35.2	38.9	40.8
$\mathcal{L}_{AN-LS}$	48.9	57.9	62.3	65.5	70.0	79.0	85.0	86.1	27.8	36.0	39.1	41.4
$\mathcal{L}_{WAN}$ [14]	57.0	60.9	63.5	64.5	76.4	82.5	85.1	85.6	37.6	41.3	43.7	44.8
$\mathcal{L}_{EPR}$ [8]	52.7	58.4	61.5	62.6	75.5	81.0	83.9	84.5	34.9	39.5	42.3	44.2
$\mathcal{L}_{ROLE}$ [8]	47.3	57.6	62.7	65.2	66.9	80.9	85.9	86.8	27.0	32.2	37.1	39.7
$\mathcal{L}_{G^2NetPL}$ (ours)	62.2	65.8	69.7	71.2	79.6	85.2	87.6	88.2	38.4	42.9	46.9	48.5

Figure 3: Convergence for pseudo labels and the predictions in COCO dataset for FSPL.

4.2 Subset Single Positive Label (SSPL)

The comparison results of SSPL setting are reported in Table 2, where 20% in the table means that only 20% of the total images have a single positive label and the rest of images are totally unlabeled [8].

Discussion. According to Table 2, G^{2}NetPL outperforms all the compared methods with different percentage of observed labels over all datasets. Among these compared approaches, $\mathcal{L}_{AN}$ is considered the simplest which is widely explored [18, 22, 27]. However, its performance is degraded by assuming the unobserved labels as negatives since those false negatives will become noisy labels. Using label smoothing with $\mathcal{L}_{AN}$ gives $\mathcal{L}_{AN-LS}$, which can overcome the negative impact of noisy labels in the multi-class setting and achieve reasonable performance as compared to $\mathcal{L}_{AN}$. As suggested in [8], we use $\mathcal{L}_{AN-LS}$ as a baseline for our work. We also observe that increasing the number of unobserved labels did not significantly affect the performance of $\mathcal{L}_{WAN}$ because down-weighing the terms related to the negative labels may help reduce the effect of noisy labels. $\mathcal{L}_{EPR}$ follows the same conclusion and performs results very close to $\mathcal{L}_{WAN}$, since the former ignores the unobserved terms and focuses on finding the positive labels according to its regularizer. Finally, we notice that $\mathcal{L}_{ROLE}$ shows a significant drop when the subset of the labeled images is small (i.e. 20%) in the end-to-end setting, due to random initialization of the estimated labels [8] as well as equally weighting the observed and unobserved labels. The G^{2}NetPL still outperforms the other methods since our game-theoretic framework not only avoids the negative factors mentioned before, but augments the label confidence into the weights as well. In fact, in 60% and 80% SSPL settings, the scores of our method (reported in Table 2) are even comparable to that under the FSPL setting (reported in Table 1). According to **SSPL versus FSPL,** we observed that G^{2}NetPL under 60% and 80% SSPL settings (Table 2) achieves comparable results to ROLE under FSPL setting (Table 1).

4.3 Compared to Semi-Supervised Models.

This subsection compares the performance of G^{2}NetPL with the state-of-the-art semi-supervised models that exploit the pseudo labels to train the network, such as FixMatch [35] and
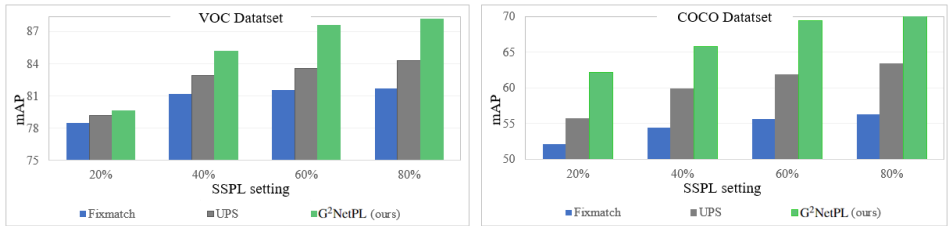
Figure 4: Comparison between our proposed model and semi-supervised models on SSPL.

UPS [53], on two datasets VoC and COCO under different SSPL settings. As shown in Fig. 4, G²NetPL outperforms FixMatch and UPS in both datasets. This is because both Fix-Match and UPS models suffer from the same issue of using threshold-based strategies as [12] (more details can be found in Section 2. There is a lack of continuity in pseudo label learning. In addition, a considerable number of unobserved data may be ignored during the entire training. On the contrary, G²NetPL (game-theoretic framework) enables continuous pseudo label training and therefore is able to capture the temporal information more effectively.

4.4   Ablation Study

G²NetPL Modules. The results are reported in Table 3 for SSPL (20%) and FSPL with linear-init setting. Linear-init learning trains the classifier linearly for 25 epochs with fixed parameters of the backbone and then fine-tunes for another 10 epochs to update the parameters of both the classifier and the backbone [8]. We start with the basic variation of the model which uses the cross-entropy loss on the observed labels. In this case, we assume that all the unobserved labels are negative, since ignoring the unobserved labels may lead to overfitting. When adding components into the model, this assumption can be removed since we use the pseudo labels to deal with the unobserved labels. We first use the sigmoid function as $F(\cdot)$ to update the soft pseudo labels which improve the mAP under different settings and datasets. Adding the confidence-aware scheduler and applying the cumulative distribution function of Guassian distribution as $F(\cdot)$ further enhances the overall performance, since they work together to make the training adaptive with respect to the confidence of the pseudo labels.

Table 3: Ablation study for all proposed model components on SSPL (20%) and FSPL.

	COCO dataset		NUS dataset	
G²NetPL Modules	SSPL (mAP)	FSPL (mAP)	SSPL (mAP)	FSPL (mAP)
Baseline	56.6	67.3	36.0	46.8
Use sigmoid as $F(\cdot)$	62.4	71.1	44.5	49.9
Add the scheduler ξ	64.4	71.4	44.8	50.4
Use CDF in (4) as $F(\cdot)$	65.0	72.7	45.3	50.9

5   Conclusions

This paper presented a novel G²NetPL for multi-label image classification under partial-label settings. In G²NetPL, two players, the network and the pseudo labels, formulate a non-zero sum game, where the network minimizes the network loss $\mathcal{L}_{G²NetPL}$ and the pseudo labels minimize $\mathcal{L}_{ACE}$ that reflects the idea of converging to 1 or 0 but penalizing deviation from the predicted labels. Through extensive experiments on different partial-label settings, we demonstrated that our G²NetPL outperforms several state-of-the-art methods by training on fewer observed labels. G²NetPL is generic to all kinds of partial-label settings.

References

[1] Rabab Abdelfattah, Xin Zhang, Zhenyao Wu, Xinyi Wu, Xiaofeng Wang, and Song Wang. Plmcl: Partial-label momentum curriculum learning for multi-label image classification. In *European Conference on Computer Vision workshops (ECCVw)*, 2022.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Serhat Selcuk Bucak, Rong Jin, and Anil K Jain. Multi-label learning with incomplete class assignments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2801–2808. IEEE, 2011.

[4] Ricardo S Cabral, Fernando Torre, Joao P Costeira, and Alexandre Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, pages 190–198, 2011.

[5] Minmin Chen, Alice Zheng, and Kilian Weinberger. Fast image tagging. In *International Conference on Machine Learning (ICML)*, pages 1274–1282. PMLR, 2013.

[6] Hong-Min Chu, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Deep generative models for weakly-supervised multi-label classification. In *European Conference on Computer Vision (ECCV)*, pages 400–415, 2018.

[7] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *ACM International Conference on Image and Video Retrieval*, pages 1–9, 2009.

[8] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 933–942, 2021.

[9] Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10):886–893, 1952.

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

[11] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Fei-Fei Li. Scalable multi-label annotation. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014.

[12] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 647–657, 2019.

[13] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8:5, 2011.

[14] Ky Fan. Fixed-point and minimax theorems in locally convex topological linear spaces. *Proceedings of the National Academy of Sciences*, 38(2):121–126, 1952.

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

[16] Dat Huynh and Ehsan Elhamifar. Interactive multi-label cnn learning with partial labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9423–9432, 2020.

[17] Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. *arXiv preprint arXiv:1705.07541*, 2017.

[18] Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision (ECCV)*, pages 67–84. Springer, 2016.

[19] Shizuo Kakutani. A generalization of brouwer's fixed point theorem. *Duke mathematical journal*, 8(3):457–459, 1941.

[20] Atsushi Kanehira and Tatsuya Harada. Multi-label ranking from positive and unlabeled data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5138–5146, 2016.

[21] Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. Multilabel classification using bayesian compressed sensing. *Advances in Neural Information Processing Systems*, 25:2645–2653, 2012.

[22] Kaustav Kundu and Joseph Tighe. Exploiting weakly supervised visual patterns to learn from partial annotations. *Advances in Neural Information Processing Systems*, 33:561–572, 2020.

[23] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[24] 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 *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[25] Yi Liu, Rong Jin, and Liu Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *AAAI*, volume 6, pages 421–426, 2006.

[26] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9596–9606, 2019.

[27] 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 *European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[28] Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, René Vidal, and Vittorio Murino. Curriculum dropout. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3552, 2017.

[29] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452. Springer, 2014.

[30] Xuesong Niu, Hu Han, Shiguang Shan, and Xilin Chen. Multi-label co-regularization for semi-supervised facial action unit recognition. *arXiv preprint arXiv:1910.11012*, 2019.

[31] Namuk Park and Songkuk Kim. How do vision transformers work? *ICLR*, 2022.

[32] Luis Pineda, Amaia Salvador, Michal Drozdzal, and Adriana Romero. Elucidating image-to-set prediction: An analysis of models, losses and datasets. *CoRR*, 2019.

[33] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

[35] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[36] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, 2017.

[37] Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. Multi-label learning with weak label. In *Twenty-fourth AAAI Conference on Artificial Intelligence*, 2010.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[39] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5552–5560, 2018.

[40] Lichen Wang, Zhengming Ding, and Yun Fu. Adaptive graph guided embedding for multi-label annotation. In *IJCAI*, 2018.

[41] Qifan Wang, Bin Shen, Shumiao Wang, Liang Li, and Luo Si. Binary codes embedding for fast image tagging with incomplete labels. In *European Conference on Computer Vision (ECCV)*, pages 425–439. Springer, 2014.

[42] Sinan Wang, Xinyang Chen, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Progressive adversarial networks for fine-grained domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9213–9222, 2020.

[43] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5017–5026, 2019.

[44] Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning (ICML)*, pages 5238–5246. PMLR, 2018.

[45] Baoyuan Wu, Siwei Lyu, and Bernard Ghanem. Ml-mg: Multi-label learning with missing labels using a mixed graph. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4157–4165, 2015.

[46] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems*, pages 2301–2309, 2013.

[47] Hao Yang, Joey Tianyi Zhou, and Jianfei Cai. Improving multi-label learning with missing labels by structured semantic correlations. In *European Conference on Computer Vision (ECCV)*, pages 835–851. Springer, 2016.

[48] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning (ICML)*, pages 593–601. PMLR, 2014.