

# DiCaP: Distribution-Calibrated Pseudo-labeling for Semi-Supervised Multi-Label Learning

Bo Han<sup>1</sup>, Zhuoming Li<sup>1</sup>, Xiaoyu Wang<sup>2</sup>, Yaxin Hou<sup>1</sup>, Hui Liu<sup>4</sup>, Junhui Hou<sup>5</sup>, Yuheng Jia<sup>1,3,4\*</sup>

<sup>1</sup>School of Computer Science and Engineering, Southeast University, Nanjing, China

<sup>2</sup>School of Software Engineering, Southeast University, Nanjing, China

<sup>3</sup>Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China

<sup>4</sup>School of Computing Information Sciences, Saint Francis University, Hong Kong, China

<sup>5</sup>Department of Computer Science, City University of Hong Kong, Hong Kong, China

{hanbo, lee Zhuoming, wang\_xy, yaxin, yhjia}@seu.edu.cn, hliu99-c@my.cityu.edu.hk, jh.hou@cityu.edu.hk

## Abstract

Semi-supervised multi-label learning (SSMLL) aims to address the challenge of limited labeled data in multi-label learning (MLL) by leveraging unlabeled data to improve the model performance. While pseudo-labeling has become a dominant strategy in SSMLL, most existing methods assign **equal weights** to all pseudo-labels regardless of their quality, which can amplify the impact of noisy or uncertain predictions and degrade the overall performance. In this paper, we **theoretically** verify that the optimal weight for a pseudo-label should reflect its correctness likelihood. **Empirically**, we observe that on the same dataset, the correctness likelihood distribution of unlabeled data remains stable, even as the number of labeled training samples varies. Building on this insight, we propose **Distribution-Calibrated Pseudo-labeling (DiCaP)**, a correctness-aware framework that estimates posterior precision to calibrate pseudo-label weights. We further introduce a dual-thresholding mechanism to separate confident and ambiguous regions: confident samples are pseudo-labeled and weighted accordingly, while ambiguous ones are explored by unsupervised contrastive learning. Experiments conducted on multiple benchmark datasets verify that our method achieves consistent improvements, surpassing state-of-the-art methods by up to **4.27%**.

**Code** — <https://github.com/hb-studying/DiCaP>

## 1 Introduction

Multi-label learning (MLL) addresses scenarios where each instance may be associated with multiple labels, enabling richer and more comprehensive predictions across complex tasks (Liu et al. 2021b). It has been widely adopted in various applications, such as image annotation (Chen et al. 2019a), text categorization (Shimura, Li, and Fukumoto 2018), and facial expression recognition (Li et al. 2022). However, a major challenge in real-world scenarios is the high cost associated with obtaining precise annotations for every instance. To alleviate this burden, semi-supervised

\*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

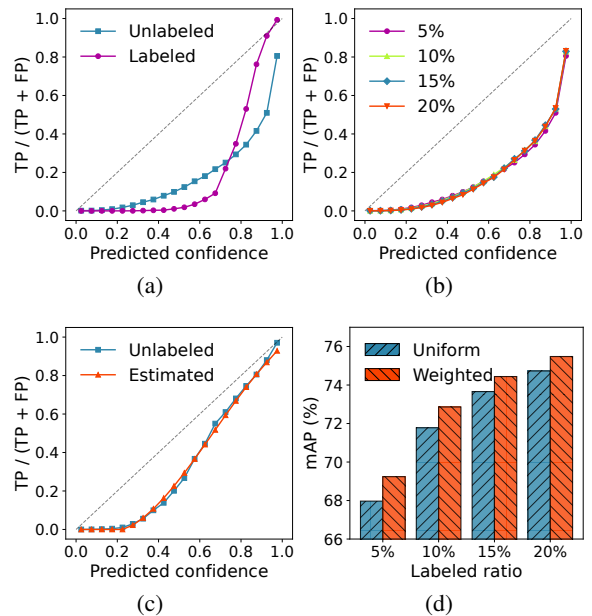


Figure 1: (a) Correctness likelihood distributions of labeled and unlabeled data after warm-up under 5% labeled setting on COCO. (b) Correctness likelihood distributions of unlabeled data across models trained with varying amounts of labeled data on COCO. (c) Estimated and true distributions for unlabeled data under 5% labeled setting on VOC. (d) Performance comparison between uniform and correctness-weighted pseudo-labeling on COCO.

multi-label learning (SSMLL) has gained considerable interest. Specifically, SSMLL aims to construct effective models by utilizing a small portion of fully labeled data together with a large pool of unlabeled samples. This approach reduces reliance on costly annotations, therefore enabling model to exploit the underlying structure of unlabeled data (Li et al. 2024; Liu et al. 2024; Jia et al. 2023; Hou and Jia 2025; Hou et al. 2025; Kou et al. 2025).

Most existing SSMLL approaches rely on fixed or

class-wise thresholds to convert prediction of the unlabeled samples into binary pseudo-labels, and use them to boost the performance. For example, CAP (Xie et al. 2023) estimates label-wise thresholds by analyzing the positive-to-negative label ratio within the labeled data. Similarly, D2L (Xiao et al. 2024) proposes a metric-adaptive thresholding strategy that considers both the quantity and quality of pseudo-labels. While these methods improve the decision boundaries, they commonly apply **uniform weights** to all pseudo-labels, regardless of their confidence levels. To be specific, the uniform weighting fails to account for the varying reliability of pseudo-labels, which ultimately limits the performance in semi-supervised multi-label learning. This motivates us to design a more adaptive weighting strategy to better exploit unlabeled data.

To this end, we propose a correctness-aware weighting strategy called **Distribution-Calibrated Pseudo-labeling (DiCaP)**. Our approach is grounded in a theoretically justified insight: the optimal weight assigned to a pseudo-label should reflect its correctness likelihood. Formally, for a pseudo-label identified as positive, its optimal weight should be the proportion of true positives among all predicted positives within the corresponding confidence bin, i.e.,  $TP/(TP + FP)$ , where TP and FP denote the number of true and false positive predictions, respectively. Please refer to Section 3 for the detailed theoretical analysis.

However, accurately estimating the correctness likelihood is a challenging task. First, as shown in Figure 1(a), modern deep neural networks tend to produce **overconfident** predictions, leading to poor calibration between predicted confidence and actual correctness (Jia et al. 2025). Second, Figure 1(a) also illustrates **distinct correctness likelihood distributions** between labeled and unlabeled data due to the difference in supervision signals during training, making it unreliable to estimate the correctness likelihood based on labeled data. **Fortunately, we observe a consistent phenomenon:** on the same dataset, even when the number of labeled training samples varies, the distribution of pseudo-label correctness likelihood for the unlabeled data remains remarkably stable (see Figure 1(b)).

This finding motivates a simple yet effective solution: we can randomly split the labeled dataset, e.g., 80% for supervised training and the remaining 20% as an estimation set. This estimation subset is first treated as unlabeled and merged with the actual unlabeled data. Since the estimation set and the true unlabeled data are drawn from the same distribution (satisfying the i.i.d. assumption) and are trained under the same strategy, we can reliably estimate the correctness likelihood distribution of pseudo-labels for the unlabeled samples on this set, and dynamically update the weights during training. Despite its simplicity, our estimation strategy demonstrates high accuracy. As shown in Figure 1(c), our estimation method produces a near-perfect match to the true posterior distribution, even under severe label scarcity scenarios (e.g., using only 57 samples for estimation on VOC under 5% labeled setting). The accurate estimation allows the model to assign appropriate weights to pseudo-labels, significantly enhancing model performance, as shown in Figure 1(d).

To further mitigate the impact of ambiguous predictions in pseudo-labeling, we introduce a dual-thresholding strategy that separates class-wise predictions into confident region and highly uncertain (ambiguous) region. For predictions falling within the confident region, we assign pseudo-labels and apply the corresponding estimated weights. In contrast, for predictions that fall into the ambiguous uncertain region, we refrain from applying hard pseudo-labeling and instead apply a consistency-based contrastive loss to regularize their representations in an unsupervised manner. In addition, we repurpose the small estimation subset of labeled data and apply supervised loss during a fine-tuning stage. This final step enables the model to leverage all labeled annotations and preserving accurate pseudo-label calibration throughout the training.

Our main contributions are summarized as follows:

1. We theoretically establish that, the optimal weight assigned to any pseudo-label equals to its correctness likelihood in SSMLL.
2. We observe that on the same dataset, the correctness likelihood distribution of unlabeled samples remains consistent, even under different numbers of labeled training samples. Leveraging this insight, a simple data-splitting approach was proposed to accurately estimate the weight distribution of the unlabeled samples.
3. We propose a dual-thresholding strategy that separates predictions into confident and uncertain regions, and adopts different strategies to handle each region.
4. Extensive experimental results validate that our method outperforms the current state-of-the-art (SOTA) methods by a large margin, e.g., a **4.27%** improvement on VOC with limited labeled data.

## 2 Related Work

### 2.1 Multi-Label Learning

Multi-label learning addresses the problem where each instance can be associated with multiple labels. Modeling label correlations has become a mainstream research direction in multi-label learning. To effectively model label dependencies, neural networks with various architectures have been proposed from various perspectives, including recurrent neural networks (Wang et al. 2016), graph convolutional networks (Guo and Wang 2021), and Transformer architectures (Liu et al. 2021a). On the other hand, many methods (Ridnik et al. 2021; Chen et al. 2019b) focus on addressing label imbalance issues in multi-label learning, including both intra-class imbalance (i.e., imbalance between positive and negative instances within each class) and inter-class imbalance (i.e., long-tailed label distributions). For instance, Ridnik et al. proposed Asymmetric Loss (ASL), which dynamically down-weights easy negatives samples and introduces hard thresholding (Ridnik et al. 2021).

Given the high cost and effort required to obtain fully annotated labels for multi-label data, recent research has increasingly focused on weakly supervised variants of MLL. These include semi-supervised multi-label learning (Xiao et al. 2024), multi-label learning with missing labels (Kim

et al. 2023), and partial multi-label learning (Li et al. 2025; Yang et al. 2024), all aiming to reduce annotation burden while maintaining competitive performance.

## 2.2 Semi-Supervised Multi-Label Learning

SSMLL tackles the challenge of learning with limited labeled data by leveraging additional unlabeled samples. Various methods have been proposed to generate reliable pseudo-labels or effectively utilize unlabeled data in SSMLL. For instance, Xie et al. adopted a class-specific thresholding approach, where thresholds are dynamically estimated from class priors to binarize predictions for unlabeled instances (Xie et al. 2023). Liu et al. enhanced pseudo-label quality by introducing a causality-guided label prior, inferred via a Structural Causal Model, and incorporates this prior into a variational inference framework to guide label enhancement (Liu et al. 2024). Li et al. addressed the variance bias between positive and negative samples by extending the binary angular margin loss with Gaussian-based feature angle transformations, and introduces a prototype-aware negative sampling scheme to further stabilize training (Li et al. 2024). Xiao et al. adopted a dual-decoupling strategy to jointly model discriminative and correlative information, and introduces a metric-adaptive thresholding mechanism to dynamically refine pseudo-label assignment throughout training (Xiao et al. 2024).

However, these methods assign **uniform weights** to all pseudo-labels without considering their quality, causing low-quality pseudo-labels to harm performance. Therefore, studying how to obtain better pseudo-label weights is important. In the following section, we theoretically analyze the optimal weighting scheme.

## 3 Theoretical Analysis

To further justify the design of our correctness-aware weighting scheme, we theoretically characterize the impact of pseudo-label noise on the learning objective in SSMLL.

### 3.1 Problem Formulation

We begin by formally defining the problem setting and notation for SSMLL. Let  $x \in \mathcal{X}$  denote a feature vector and  $y \in \mathcal{Y}$  denote the corresponding multi-label annotation, where  $\mathcal{X} \subseteq \mathbb{R}^d$  is the input space and  $\mathcal{Y} = \{0, 1\}^C$  is the label space consisting of  $C$  possible categories. And,  $y_{ic} = 1$  indicates the relevance of the  $c$ -th label to the instance  $x_i$ , while  $y_c = 0$  signifies irrelevance.

In SSMLL, the training data consists of a labeled set with  $N_l$  training examples  $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$  and an unlabeled set with  $N_u$  training instances  $\mathcal{D}_u = \{x_j\}_{j=1}^{N_u}$ , where  $N_l \ll N_u$ . The primary objective is to train a multi-label classifier by leveraging both labeled and unlabeled data, where the unlabeled data are assigned pseudo-labels  $\hat{y}$ . The model, parameterized by  $\theta$  and denoted as  $f_\theta : \mathcal{X} \rightarrow [0, 1]^C$ , predicts a vector of class-wise relevance scores.

### 3.2 Derivation of the Optimal Weighting Strategy

In SSMLL, model training heavily depends on the pseudo-labels  $\hat{y}$  assigned to the unlabeled data. However, these

pseudo-labels are inevitably noisy. An effective strategy is to assign a weight to the loss of each pseudo-labeled sample, so that more reliable predictions are emphasized and less reliable ones are down-weighted. Intuitively, this weight  $w(p)$  should be a function of the model’s confidence  $p$ .

To derive the optimal weighting function, we assume the ground-truth labels of the unlabeled data are known. Let  $\mathbb{I}[\hat{y} = y]$  be a binary indicator denoting whether a pseudo-label is correct. The goal is to assign higher weights to correct pseudo-labels and lower weights to incorrect ones. This leads to formulating the optimization of  $w(p)$  as minimizing the binary cross-entropy (BCE) loss between the weight and the correctness indicator. For a pseudo-label  $\hat{y}_i$  with confidence  $p_i$  and ground-truth label  $y_i$ , the loss is defined as:

$$\ell_{\text{BCE}}(w(p_i), \mathbb{I}[\hat{y}_i = y_i]) = \underbrace{-\mathbb{I}[\hat{y}_i = y_i] \log(w(p_i))}_{\text{increase weight for correct samples}} - \underbrace{\mathbb{I}[\hat{y}_i \neq y_i] \log(1 - w(p_i))}_{\text{decrease weight for incorrect samples}}. \quad (1)$$

Since a single confidence value  $p_i$  typically corresponds to multiple samples, we aim to minimize the expected BCE loss over all samples with same confidence:

$$w^*(p_i) = \arg \min_{w(p_i)} \mathbb{E}_{y|p_i} [\ell_{\text{BCE}}(w(p_i), \mathbb{I}[\hat{y} = y])] = P(\hat{y} = y | p_i), \quad (2)$$

which shows that the optimal weighting function is exactly the posterior correctness likelihood of the pseudo-label given the confidence score  $p_i$ .

However, estimating correctness likelihood at each individual confidence value  $p_i$  is inherently unreliable due to data sparsity—especially under limited data, where only a few samples may share the same confidence score. To address this issue, we propose to approximate the optimal weighting function by aggregating samples within similar confidence ranges. The continuous confidence interval  $[0, 1]$  is divided into  $K$  non-overlapping bins, denoted as  $\{\mathcal{B}_1, \dots, \mathcal{B}_K\}$ . For each bin  $\mathcal{B}_k$ , we compute the empirical correctness rate of pseudo-labels whose predicted confidence scores fall within that bin. The optimal correctness likelihood on unlabeled data is then given by:

$$w_k^*(p) = P(\hat{y} = y | p \in \mathcal{B}_k), \quad (3)$$

which serves as a practical approximation of the optimal weight for any confidence value  $p \in \mathcal{B}_k$ .

## 4 Methods

### 4.1 Distribution-Calibrated Weight Estimation

**Estimation Set Construction.** As discussed earlier, the optimal weight for a pseudo-label should reflect its correctness likelihood. However, estimating this likelihood is inherently challenging without ground-truth labels for unlabeled data. Moreover, due to overconfidence issue and the different correctness distributions between labeled and unlabeled data, it is unreliable to infer correctness likelihood directly from the model’s predicted probabilities or based on the labeled set. Fortunately, we observe that within the

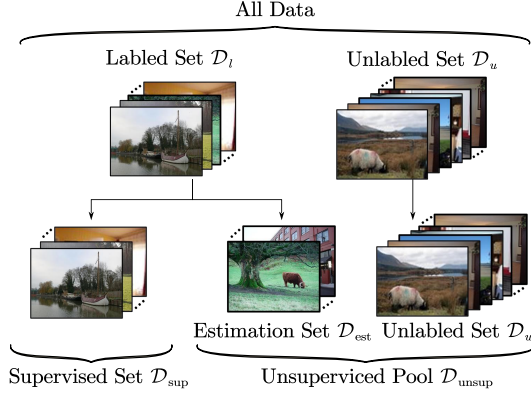


Figure 2: Illustration of estimation set construction and relationships among  $\mathcal{D}_l$ ,  $\mathcal{D}_u$ ,  $\mathcal{D}_{\text{sup}}$ ,  $\mathcal{D}_{\text{est}}$ , and  $\mathcal{D}_{\text{unsup}}$ .

same dataset, the distribution of correctness likelihood over unlabeled data remains stable, even when the number of labeled training samples varies (see Figure 1(b)).

Motivated by this observation, we divide the labeled dataset  $\mathcal{D}_l$  into two disjoint subsets: a supervised training set  $\mathcal{D}_{\text{sup}}$  for supervised training and an estimation set  $\mathcal{D}_{\text{est}}$  for estimating the weight, such that  $\mathcal{D}_l = \mathcal{D}_{\text{sup}} \cup \mathcal{D}_{\text{est}}$  and  $\mathcal{D}_{\text{sup}} \cap \mathcal{D}_{\text{est}} = \emptyset$ . The estimation set  $\mathcal{D}_{\text{est}}$  is treated as unlabeled and merged with the original unlabeled data  $\mathcal{D}_u$  to form a unified pool:

$$\mathcal{D}_{\text{unsup}} = \mathcal{D}_u \cup \{x_i \mid (x_i, y_i) \in \mathcal{D}_{\text{est}}\}. \quad (4)$$

As shown in Figure 2, samples from  $\mathcal{D}_{\text{est}}$  are handled identically to those in  $\mathcal{D}_u$ , ensuring distributional consistency across the entire unsupervised pool. This design allows us to reliably use  $\mathcal{D}_{\text{est}}$  to estimate the correctness likelihood distribution over the unlabeled data.

**Weight Estimation.** As discussed above, we can empirically approximate the weight using the estimation set  $\mathcal{D}_{\text{est}}$ . Here, we introduce the specific implementation strategy for distribution-calibrated weight estimation for  $\mathcal{D}_{\text{unsup}}$ .

Following the theoretical analysis in Section 3, we uniformly partition the confidence interval  $[0, 1]$  into  $K = 20$  non-overlapping bins, denoted as  $\{\mathcal{B}_1, \dots, \mathcal{B}_K\}$ , where each bin corresponds to the range  $\mathcal{B}_k = [\frac{k-1}{K}, \frac{k}{K})$ . For each prediction score  $p$  and its associated ground-truth label  $y$  in  $\mathcal{D}_{\text{est}}$ , we count the number of true positives and true negatives that fall into each confidence bin:

$$n_k^{\text{pos}} = \sum_{(x,y) \in \mathcal{D}_{\text{est}}} \sum_{c=1}^C \mathbb{I}(p_c \in \mathcal{B}_k, y_c = 1), \quad (5)$$

$$n_k^{\text{neg}} = \sum_{(x,y) \in \mathcal{D}_{\text{est}}} \sum_{c=1}^C \mathbb{I}(p_c \in \mathcal{B}_k, y_c = 0).$$

We then compute the estimated positive and negative proportions as:

$$r_k^{\text{pos}} = \frac{n_k^{\text{pos}}}{n_k^{\text{pos}} + n_k^{\text{neg}} + \epsilon}, \quad r_k^{\text{neg}} = 1 - r_k^{\text{pos}}, \quad (6)$$

where  $\epsilon$  is a small constant to avoid division by zero.

During training, for any predicted confidence  $p$  and corresponding pseudo-label  $\hat{y} \in \{0, 1\}$ , we assign a soft weight based on linear interpolation between adjacent bins:

$$\begin{aligned} w(p, \hat{y} = 1) &= \left(\frac{k+1}{K} - p\right) r_k^{\text{pos}} + \left(p - \frac{k}{K}\right) r_{k+1}^{\text{pos}}, \\ w(p, \hat{y} = 0) &= \left(\frac{k+1}{K} - p\right) r_k^{\text{neg}} + \left(p - \frac{k}{K}\right) r_{k+1}^{\text{neg}}, \\ \text{s.t. } &\frac{k}{K} \leq p < \frac{k+1}{K}. \end{aligned} \quad (7)$$

This correctness-based weight reflects the estimated probability that the pseudo-label is correct, and is directly used to modulate the training loss.

## 4.2 Dual-Thresholding for Pseudo-Label

While correctness-based weighting alleviates the effect of noisy pseudo-labels, there still exist highly ambiguous regions where predictions are particularly unreliable, which can negatively affect model training. To further mitigate the impact of such uncertain predictions, we adopt a dual-thresholding strategy to explicitly partition pseudo-labeled data into confident and uncertain subsets.

Inspired by (Xu et al. 2025), we derive class-wise dynamic thresholds from the prediction statistics on the labeled training set. For each class  $c$ , we first collect the predicted confidence scores from positive and negative label in  $\mathcal{D}_{\text{sup}}$ . Let  $\mathcal{P}_c^{\text{pos}} = \{p_{ic} \mid (x_i, y_i) \in \mathcal{D}_{\text{sup}}, y_{ic} = 1\}$  and  $\mathcal{P}_c^{\text{neg}} = \{p_{ic} \mid (x_i, y_i) \in \mathcal{D}_{\text{sup}}, y_{ic} = 0\}$ . We define the thresholds as the mid-range of each group:

$$\begin{aligned} \tau_c^{\text{pos}} &= \frac{\max(\mathcal{P}_c^{\text{pos}}) + \min(\mathcal{P}_c^{\text{pos}})}{2}, \\ \tau_c^{\text{neg}} &= \frac{\max(\mathcal{P}_c^{\text{neg}}) + \min(\mathcal{P}_c^{\text{neg}})}{2}. \end{aligned} \quad (8)$$

Given an unlabeled sample  $x_u$  and its predicted confidence scores  $\{p_{uc}\}_{c=1}^C$ , we assign pseudo-labels by applying the derived dual thresholds for each class. The pseudo-label  $\hat{y}_{uc}$  is categorized into one of three types based on its confidence score:

$$\hat{y}_{uc} = \begin{cases} 1, & p_{uc} > \tau_c^{\text{pos}}, \\ 0, & p_{uc} < \tau_c^{\text{neg}}, \\ -1, & \tau_c^{\text{neg}} \leq p_{uc} \leq \tau_c^{\text{pos}}, \end{cases} \quad (9)$$

where the labels 1, 0, and  $-1$  correspond to confident positive, confident negative, and uncertain predictions, respectively. The samples assigned with 1 or 0 are aggregated into the confident set  $\mathcal{D}_{\text{conf}}$ , while those assigned with  $-1$  constitute the uncertainty set  $\mathcal{D}_{\text{uncer}}$ .

We retain only the confident samples  $(x_{ic}, \hat{y}_{ic}) \in \mathcal{D}_{\text{conf}}$  for supervised training. Their pseudo-label loss is computed as:

$$\mathcal{L}_{\text{pseudo}} = \frac{1}{|\mathcal{D}_{\text{conf}}|} \sum_{\hat{y}_{ic} \in \{1,0\}} w(p_{ic}, \hat{y}_{ic}) \cdot \ell(p_{ic}, \hat{y}_{ic}), \quad (10)$$

where  $w(\cdot)$  denotes the correctness-based weight defined in Eq. (7), and we use Asymmetric Loss (ASL) (Ridnik et al. 2021) for  $\ell(\cdot, \cdot)$ .

### 4.3 Robust Representation Learning for Uncertain Samples

Uncertain samples, whose predictions lie within the ambiguous confidence interval, are unsuitable for direct supervision. To exploit their representational value without introducing noisy gradients, we incorporate an unsupervised contrastive learning objective tailored to the multi-label setting.

Specifically, we extend the standard instance-level InfoNCE loss (van den Oord, Li, and Vinyals 2019) into a **class-wise contrastive framework** to suit the multi-label setting. For each sample  $x_i$ , we generate a weakly augmented view  $x_i^w$  and a strongly augmented view  $x_i^s$ . A shared encoder is used to extract class-wise feature embeddings  $\{z_{ic}^w\}_{c=1}^C$  and  $\{z_{ic}^s\}_{c=1}^C$  for the two views, respectively. For all uncertain predictions (i.e., those with  $\hat{y}_{ic} = -1$ ), we collect the corresponding feature pairs  $(z_{ic}^w, z_{ic}^s)$  to construct the contrastive training set.

We treat each pair  $(z_{ic}^w, z_{ic}^s)$  as a positive pair and all other class-wise features within the mini-batch as negatives. The class-wise contrastive loss is defined as:

$$\mathcal{L}_{\text{uncer}} = -\frac{1}{2B} \sum_{i=1}^{2B} \log \frac{\exp(z_i \cdot z_i^+ / \tau)}{\sum_{j=1}^{2B} \mathbb{I}_{i \neq j} \exp(z_i \cdot z_j / \tau)}, \quad (11)$$

where  $B$  denotes the number of uncertain samples,  $z_i^+$  is the positive counterpart of  $z_i$ , and  $\tau$  is a temperature hyperparameter that controls the sharpness of the distribution. Inspired by recent self-supervised learning study (Chen et al. 2020), we can also employ it to utilize all data during the warm-up stage to improve representation generality.

Finally, the overall training objective during the pseudo-labeling stage is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{pseudo}} + \mathcal{L}_{\text{uncer}}, \quad (12)$$

where  $\mathcal{L}_{\text{sup}}$  denotes the supervised loss on  $\mathcal{D}_{\text{sup}}$ ,  $\mathcal{L}_{\text{pseudo}}$  is the weighted pseudo-label loss on confident samples, and  $\mathcal{L}_{\text{uncer}}$  represents the contrastive loss of uncertain samples.

### 4.4 Fine-Tuning on Estimation Set

To fully utilize all labeled data, we introduce a final fine-tuning stage on the estimation subset  $\mathcal{D}_{\text{est}}$ . This subset was previously treated as unlabeled to estimate pseudo-label reliability, and its ground-truth annotations were excluded from earlier supervised training. Leveraging its ground-truth annotations can thus further enhance the model’s generalization on all labeled data.

To mitigate overfitting and reduce computational cost, we freeze the backbone and fine-tune only the classification head using  $\mathcal{D}_{\text{est}}$ . The optimization objective is:

$$\mathcal{L}_{\text{ft}} = \frac{1}{|\mathcal{D}_{\text{est}}|} \sum_{(x,y) \in \mathcal{D}_{\text{est}}} \ell(f_{\theta}(x), y), \quad (13)$$

where  $\ell(\cdot, \cdot)$  denotes ASL. By incorporating previously unused annotations, this fine-tuning stage complements the training process and ensures complete utilization of labeled data.

For ease of understanding, we provide the algorithmic process in Algorithm 1.

---

Algorithm 1: Algorithm of DiCaP

---

**Input:** Labeled dataset  $\mathcal{D}_l$ , Unlabeled dataset  $\mathcal{D}_u$

**Output:** Trained multi-label classifier  $f_{\theta}$

- 1: Split  $\mathcal{D}_l$  into  $\mathcal{D}_{\text{sup}}$  and  $\mathcal{D}_{\text{est}}$ ,  $\mathcal{D}_{\text{unsup}} = \mathcal{D}_u \cup \mathcal{D}_{\text{est}}$ .
  - 2: Warm up on  $\mathcal{D}_{\text{sup}} \cup \mathcal{D}_{\text{unsup}}$  to initialize  $f_{\theta}$ .
  - 3: **for** each epoch **do**
  - 4:   Estimate pseudo-label weights  $w(p)$  using  $\mathcal{D}_{\text{est}}$ .
  - 5:   Derive thresholds  $\{\tau_c^{\text{pos}}, \tau_c^{\text{neg}}\}^C$  from  $\mathcal{D}_{\text{sup}}$ .
  - 6:   Generate pseudo-labels for  $\mathcal{D}_{\text{unsup}}$  using Eq. (9).
  - 7:   Update model by minimizing the total loss Eq. (13).
  - 8: **end for**
  - 9: Fine-tune the classification head on  $\mathcal{D}_{\text{est}}$ .
- 

## 5 Experiments

### 5.1 Settings

**Datasets.** To validate our proposed approach, we conducted experiments on four commonly used real-world multi-label image datasets, including MS-COCO 2014 (COCO) (Lin et al. 2014), Pascal VOC 2007 (VOC) (Everingham et al. 2010), NUS-WIDE (NUS) (Chua et al. 2009), and Animals with Attributes2 (AWA) (Lampert, Nickisch, and Harmeling 2014). Following (Xie et al. 2023), we transform these datasets into SSL versions. For each dataset, we randomly select a proportion  $\rho$  of training samples as labeled ones, and the remaining as unlabeled ones. We set  $\rho \in \{5\%, 10\%, 15\%, 20\%\}$ , to explore the performance of our method under different data proportions.

**Comparison Methods.** We compare our model with 11 baseline methods, which are categorized into three groups. **BCE**, **ASL** (Ridnik et al. 2021), and **MLD** (Ridnik et al. 2023) are classic multi-label learning methods. These methods only use the labeled data for training. **Top-1**, **Top-k**, and **IAT** (Xie et al. 2023) are instance-based pseudo-labeling methods. **DRML** (Wang et al. 2020), **CAP** (Xie et al. 2023), **PCLP** (Liu et al. 2024), **BBAM** (Li et al. 2024), and **D2L** (Xiao et al. 2024) represent the SOTA SSMLL methods based on deep models.

**Implementation Details.** Following the experimental setup in (Xie et al. 2023), we adopt a ResNet-50 (He et al. 2016) backbone pre-trained on ImageNet (Deng et al. 2009) for feature extraction. For the decoder, we employ the ML-Decoder framework (Ridnik et al. 2023) to produce class-wise embeddings. All input images are resized to  $224 \times 224$ , and data augmentation is applied using RandAugment (Cubuk et al. 2020) and Cutout (DeVries and Taylor 2017). The optimizer is AdamW (Loshchilov and Hutter 2019) with the weight decay of  $1e-4$ , and we employ a one-cycle learning rate scheduler with a maximum learning rate of  $1e-4$ . To stabilize training, we apply an exponential moving average (EMA) to the model parameters with a decay rate of 0.9997. All experiments are conducted on NVIDIA GeForce RTX 3090 GPUs. To ensure reproducibility, the random seed is fixed to 1 throughout all experiments. Consistent with prior works (Xie et al. 2023), mean average precision (mAP) is applied to evaluate models’ performance.

Method	VOC				COCO				NUS				AWA			
	5%	10%	15%	20%	5%	10%	15%	20%	5%	10%	15%	20%	5%	10%	15%	20%
BCE	65.40	75.48	77.87	79.00	57.09	62.34	65.55	67.31	40.12	45.04	47.04	48.29	61.33	63.35	62.51	63.00
ASL	71.41	77.81	79.12	79.84	57.87	62.95	65.73	67.43	42.04	46.07	48.04	49.55	60.40	63.06	62.88	63.31
MLD	74.18	81.10	82.74	84.29	62.45	67.27	69.95	71.22	39.78	44.31	47.02	49.06	62.28	63.57	63.17	63.94
Top-1	75.34	80.80	82.93	83.67	59.42	63.52	65.13	66.88	39.27	46.05	47.02	46.69	63.39	64.63	64.60	63.86
Top-k	73.62	80.20	82.17	83.03	59.83	64.02	65.21	67.45	39.18	46.15	46.98	46.70	63.87	64.41	64.81	64.49
IAT	71.88	80.18	82.87	83.99	60.76	65.60	65.31	69.29	40.10	46.45	47.39	47.15	62.93	64.17	63.57	63.57
DRML	61.75	70.97	72.97	74.44	53.59	57.02	58.62	59.18	30.57	35.03	37.93	40.01	61.60	62.46	63.77	63.38
CAP	75.90	81.83	83.10	84.32	62.88	67.18	68.99	70.43	44.98	47.81	49.04	51.37	63.90	64.15	64.40	64.51
PCLP	77.25	82.21	83.72	84.59	64.43	69.02	70.86	71.52	46.39	48.83	50.57	52.45	64.30	65.38	64.47	64.49
BBAM	78.66	83.45	84.54	84.58	63.54	67.41	68.86	70.23	33.15	39.12	41.26	42.52	64.19	64.63	64.16	64.76
D2L	79.26	<u>84.06</u>	<u>86.25</u>	<u>87.16</u>	69.30	73.06	74.63	75.70	<u>46.86</u>	<u>50.25</u>	<u>51.61</u>	<u>52.64</u>	<u>64.66</u>	<u>65.57</u>	<u>64.95</u>	<u>65.06</u>
Ours	<b>83.53</b>	<b>87.92</b>	<b>88.40</b>	<b>88.48</b>	<b>70.07</b>	<b>73.55</b>	<b>75.06</b>	<b>75.90</b>	<b>48.37</b>	<b>51.24</b>	<b>52.30</b>	<b>53.61</b>	<b>66.32</b>	<b>66.48</b>	<b>66.49</b>	<b>66.78</b>
$\Delta$	+4.27	+3.86	+2.15	+1.32	+0.77	+0.49	+0.43	+0.20	+1.51	+0.99	+0.69	+0.97	+1.66	+0.91	+1.54	+1.72

Table 1: Comparison of our model with other methods, with labeled rate 5%, 10%, 15% and 20%. mAP(%) is adopted as the evaluation metric. Optimal values are denoted in bold, and the second-best values are underlined.  $\Delta$  denotes the performance gap between our method and the best compared method.

For all compared methods, we use their officially recommended optimal hyperparameter settings. During the warm-up phase of our method, we utilize 80% of the labeled data for supervised learning. In contrast, all baseline methods follow their original protocols and leverage the full set of labeled data during training. In the fine-tuning phase, the backbone is frozen, and only the classification head is updated. We train for 20 epochs with a batch size of 32.

## 5.2 Comparison with SOTA Methods

As shown in Table 1, we conduct a comprehensive comparison against SOTA methods across four benchmark datasets under four different labeled data ratios. The results clearly demonstrate that our proposed method consistently achieves the best performance across all datasets and label-sparsity settings. Specifically, our method surpasses the second-best approach by an average margin of 2.9% on the VOC dataset. Our method also remains robust in highly challenging scenarios where only 5% of the training data is labeled. Under this limited supervision setting, we outperform the second-best method by notable margins with 4.27%, 1.51%, and 1.67% on VOC, NUS-WIDE, and AWA, respectively, highlighting the effectiveness of our method.

Among the baselines, DRML, a two-stage semi-supervised method, performs poorly on large-scale image datasets. This is mainly due to its decoupled training scheme, which prevents end-to-end feature learning. As a result, DRML performs worse than baselines BCE and ASL. On the other hand, D2L achieves consistently competitive results across datasets, securing the second-best performance overall. This demonstrates the benefit of refining pseudo-labels using model predictions. However, D2L applies uniform weights to all pseudo-labels, which limits its performance. By comparison, our method explicitly models the *correctness likelihood* of each pseudo-label and dynamically adjusts their weights during training. This weighting mechanism enables more effective use of pseudo-labels and

Ablations	COCO		NUS		Avg.
	5%	10%	5%	10%	
Baseline	65.13	69.53	42.69	45.91	55.82
+ DCW	69.18	72.46	45.20	48.08	58.73 +2.91
+ DTH	69.24	72.86	47.73	49.82	59.91 +1.18
+ URRL	69.79	72.97	47.86	50.38	60.25 +0.34
+ WCL	69.98	73.14	47.96	50.51	60.40 +0.15
+ FTE	<b>70.07</b>	<b>73.55</b>	<b>48.37</b>	<b>51.24</b>	<b>60.81</b> +0.41

Table 2: Ablation study of different components on COCO and NUS. **Baseline** corresponds to training only on labeled data. **DCW** denotes using the estimated correctness weights as soft pseudo-labels. **DTH** refers to the dual-threshold confidence partitioning strategy. **URRL** represents robust representation learning for uncertain samples. **WCL** indicates incorporating all data into the warm-up stage via class-wise contrastive loss. **FTE** denotes fine-tuning on  $\mathcal{D}_{est}$ . mAP(%) is adopted as metric.

leads to consistent performance improvements.

## 5.3 Further Analysis

**All components contribute, with the distribution-calibrated weighting providing the largest gain.** In Table 2, we evaluate the contribution of each component in DiCaP through ablation experiments on COCO and NUS under 5% and 10% labeled data. The results clearly demonstrate that all modules contribute positively to the overall performance. For instance, on COCO with 5% labeled data, incorporating the distribution-calibrated weighting (DCW) module brings a notable improvement of 4.05%. On NUS, introducing the dual-thresholding strategy (DTH) yields an additional 2.14% gain. The benefit of the final fine-tuning step becomes more evident as the proportion of labeled data increases, providing a 0.73% improvement on NUS with 10%

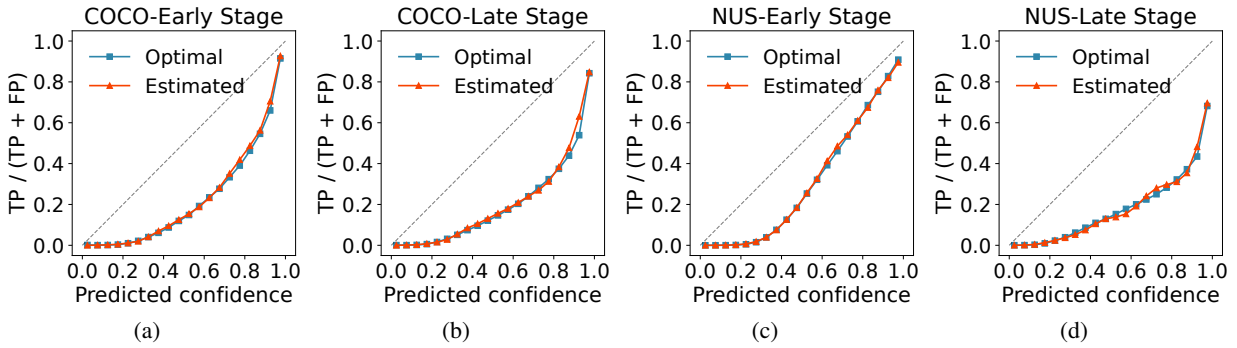


Figure 3: Comparison between estimated and optimal correctness likelihood distributions at early and late training stages on COCO and NUS under 5% labeled setting.

Weighting policy	Labeled rate				Avg.
	5%	10%	15%	20%	
Uniform	67.97	71.78	73.66	74.73	72.04
Confidence	67.83	71.63	73.50	74.42	71.84
Labeled	68.47	72.26	74.09	74.88	72.42
Ours	69.24	72.86	74.44	75.48	73.01
Optimal	69.32	72.93	74.57	75.54	73.09

Table 3: Effectiveness of weighting policies on COCO with various labeled rate. mAP(%) is adopted as metric.

labeled data. Among all components, the DCW module delivers the most significant overall contribution, with an average gain of 2.92%. These consistent improvements highlight the importance of distribution-calibrated weighting strategy.

**Our estimated correctness likelihoods are highly accurate.** To further validate the accuracy of our estimated correctness-likelihood distributions, we visualize both the estimated and optimal distributions at different training stages on COCO and NUS datasets under 5% labeled setting, as shown in Figure 3. Here, the optimal distributions are computed using the true labels of the unlabeled data. The comparison on VOC is already presented in Figure 1(c). As illustrated, our estimated distributions closely match the optimal correctness curves across datasets and training stages, demonstrating the reliability of our estimation strategy.

**Our weighting strategy is highly effective.** In Table 3, we further evaluate the effectiveness of our estimated weights by comparing with several weighting strategies: (1) **Uniform**: assigns all pseudo-labels a constant weight of 1; (2) **Confidence**: uses the predicted probability  $p$  as the weight; (3) **Labeled**: estimates weights from the labeled data; (4) **Optimal**: computes ideal weights from ground-truth correctness on the unlabeled data. As shown in Table 3, our proposed method (denoted as **Ours**) consistently outperforms all other weighting strategies. Notably, the **Confidence** strategy, which directly uses predicted probabilities, performs worse than the **Uniform** baseline due to the unreliability of raw confidence scores. Meanwhile, our method achieves performance close to the **Optimal** setting, validat-

Metrics	Method	VOC	COCO	NUS	AWA
Time (min)	D2L	1.85	21.42	37.74	8.38
	Ours	<b>1.47</b>	<b>18.29</b>	<b>29.38</b>	<b>6.61</b>
Memory (GB)	D2L	7.34	14.17	14.13	14.12
	Ours	<b>2.98</b>	<b>4.44</b>	<b>4.43</b>	<b>4.44</b>

Table 4: Time/memory comparison between D2L and our method on single 24GB NVIDIA GeForce RTX 3090 GPU. “Time” is the training time per epoch, including threshold updating and weight estimating, “Memory” is the max GPU memory allocated during training phase.

ing the accuracy of our estimated correctness likelihoods.

**Our method is more efficient.** As shown in Table 4, we compare the time and GPU memory usage of our method with the best compared method D2L under the 5% labeled ratio setting. From results demonstrated in Table 4, D2L requires significantly more GPU memory and has slower training speed due to its patch-wise image processing strategy. In contrast, our method achieves much lower GPU memory usage and faster training speed. Considering both the efficiency gains and the superior classification performance reported in Table 1, our method demonstrates clear advantages in both effectiveness and computational cost.

## 6 Conclusion

In this paper, we propose DiCaP, a Distribution-Calibrated Pseudo-labeling framework for SSMLL. We theoretically derive the optimal pseudo-label weight as the posterior correctness likelihood conditioned on prediction confidence. Motivated by the observation that correctness distributions remain stable across varying labeled data, we introduce a bin-based estimation scheme by partitioning the labeled data to approximate this optimal weight. To further reduce the impact of ambiguous predictions, we propose a dual-thresholding strategy that separates pseudo-labels into confident and uncertain regions. Finally, we introduce a fine-tuning stage to fully exploit all annotated data. Extensive experiments on multiple benchmarks validate the effectiveness of our approach under different label sparsity conditions.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U24A20322, 62576094 and 62422118. This work is also supported by Hong Kong UGC under grants UGC/FDS11/E03/24, UGC/FDS11/E03/25, and Hong Kong Research Grants Council under Grant 11219324. This research work is also supported by the Big Data Computing Center of Southeast University.

## References

- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, volume 119, 1597–1607.
- Chen, Z.; Wei, X.; Wang, P.; and Guo, Y. 2019a. Multi-Label Image Recognition With Graph Convolutional Networks. In *CVPR*, 5177–5186.
- Chen, Z.; Wei, X.; Wang, P.; and Guo, Y. 2019b. Multi-Label Image Recognition With Graph Convolutional Networks. In *CVPR*, 5177–5186.
- Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. 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*, 1–9.
- Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.
- Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.
- DeVries, T.; and Taylor, G. W. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv:1708.04552*.
- Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.
- Guo, H.; and Wang, S. 2021. Long-Tailed Multi-Label Visual Recognition by Collaborative Training on Uniform and Re-Balanced Samplings. In *CVPR*, 15089–15098.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.
- Hou, Y.; Han, B.; Jia, Y.; Liu, H.; and Hou, J. 2025. Keep It on a Leash: Controllable Pseudo-label Generation Towards Realistic Long-Tailed Semi-Supervised Learning. In *NeurIPS*.
- Hou, Y.; and Jia, Y. 2025. A Square Peg in a Square Hole: Meta-Expert for Long-Tailed Semi-Supervised Learning. In *ICML*.
- Jia, Y.; Cheng, J.; Liu, H.; and Hou, J. 2025. Towards Calibrated Deep Clustering Network. In *ICLR*.
- Jia, Y.; Lu, G.; Liu, H.; and Hou, J. 2023. Semi-Supervised Subspace Clustering via Tensor Low-Rank Representation. *IEEE Trans. Circuits Syst. Video Technol.*, 33(7): 3455–3461.
- Kim, Y.; Kim, J.; Jeong, J.; Schmid, C.; Akata, Z.; and Lee, J. 2023. Bridging the Gap Between Model Explanations in Partially Annotated Multi-Label Classification. In *CVPR*, 3408–3417.
- Kou, Z.; Xie, Y.; Wang, H.; Chen, J.; Wang, J.; Xie, M.-K.; Chen, S.; Jia, Y.; Liu, T.; and Geng, X. 2025. RankMatch: A Novel Approach to Semi-Supervised Label Distribution Learning Leveraging Rank Correlation between Labels. In *NeurIPS*.
- Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3): 453–465.
- Li, X.; Liang, S.; Li, C.; Wang, P.; and Gu, F. 2024. Semi-supervised Multi-label Learning with Balanced Binary Angular Margin Loss. In *NeurIPS*.
- Li, Y.; Gao, Y.; Chen, B.; Zhang, Z.; Lu, G.; and Zhang, D. 2022. Self-Supervised Exclusive-Inclusive Interactive Learning for Multi-Label Facial Expression Recognition in the Wild. *IEEE Trans. Circuits Syst. Video Technol.*, 32(5): 3190–3202.
- Li, Z.; Jia, Y.; Yu, M.; and Miao, Z. 2025. Calibrated Disambiguation for Partial Multi-label Learning. In *AAAI*, 18620–18628.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.
- Liu, B.; Xu, N.; Fang, X.; and Geng, X. 2024. Correlation-Induced Label Prior for Semi-Supervised Multi-Label Learning. In *ICML*.
- Liu, S.; Zhang, L.; Yang, X.; Su, H.; and Zhu, J. 2021a. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*.
- Liu, W.; Wang, H.; Shen, X.; and Tsang, I. W. 2021b. The emerging trends of multi-label learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11): 7955–7974.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.
- Ridnik, T.; Baruch, E. B.; Zamir, N.; Noy, A.; Friedman, I.; Protter, M.; and Zelnik-Manor, L. 2021. Asymmetric Loss For Multi-Label Classification. In *ICCV*, 82–91.
- Ridnik, T.; Sharir, G.; Ben-Cohen, A.; Baruch, E. B.; and Noy, A. 2023. ML-Decoder: Scalable and Versatile Classification Head. In *WACV*, 32–41.
- Shimura, K.; Li, J.; and Fukumoto, F. 2018. HFT-CNN: Learning Hierarchical Category Structure for Multi-label Short Text Categorization. In *EMNLP*, 811–816.
- van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748*.
- Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *CVPR*, 2285–2294.

Wang, L.; Liu, Y.; Qin, C.; Sun, G.; and Fu, Y. 2020. Dual Relation Semi-Supervised Multi-Label Learning. In *AAAI*, 6227–6234.

Xiao, J.; Xie, M.; Fan, H.; Niu, G.; Sugiyama, M.; and Huang, S. 2024. Dual-Decoupling Learning and Metric-Adaptive Thresholding for Semi-supervised Multi-label Learning. In *ECCV*.

Xie, M.; Xiao, J.; Liu, H.; Niu, G.; Sugiyama, M.; and Huang, S. 2023. Class-Distribution-Aware Pseudo-Labeling for Semi-Supervised Multi-Label Learning. In *NeurIPS*.

Xu, Y.; Li, H.; Zhu, M.; Wang, N.; and Gao, X. 2025. Boosting Semi-Supervised Facial Attribute Recognition With Dynamic Threshold Pairs. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(7): 6926–6935.

Yang, F.; Jia, Y.; Liu, H.; Dong, Y.; and Hou, J. 2024. Noisy Label Removal for Partial Multi-Label Learning. In *KDD*, 3724–3735. ACM.