Semi-supervised Multi-label Learning with Balanced Binary Angular Margin Loss

Ximing Li[1,2], Silong Liang[1,2], Changchun Li[1,2,*], Pengfei Wang[3,4] and Fangming Gu[1,2]

[1]*College of Computer Science and Technology, Jilin University, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China*
[3]*Computer Network Information Center, Chinese Academy of Sciences, China*
[4]*University of Chinese Academy of Sciences, Chinese Academy of Sciences, China*

Presenter：Silong Liang

CONTENTS

NEURAL INFORMATION PROCESSING SYSTEMS

NEURAL INFORMATION PROCESSING SYSTEMS

CHAPTER

01

Semi-supervised Multi-label Learning

Semi-Supervised Multi-Label Learning

Multi-Label

Label：
people,
bicycle,
road

Multi-Class

Label：
people ×
bird ✓
road ×

In multi-label learning, each object is also represented by an instance and associated with a set of labels instead of a single label. The task of multi-label learning is to learn a function that can predict the correct set of labels for unseen instances.

Semi-Supervised Multi-Label Learning

In supervised learning, the class labels of the samples are known, and the goal of learning is to find the relationship between the features of the samples and their classes.

However, in many real-world scenarios, the cost of manually labeling data is high, which results in a scarcity of labeled samples. In contrast, unlabeled data can be easily collected, often in quantities that are hundreds of times greater than that of labeled data.

Therefore, semi-supervised learning seeks to train a classifier using a large number of unlabeled samples and a small number of labeled ones, addressing the challenge of insufficient labeled data.

NEURAL INFORMATION PROCESSING SYSTEMS

CHAPTER

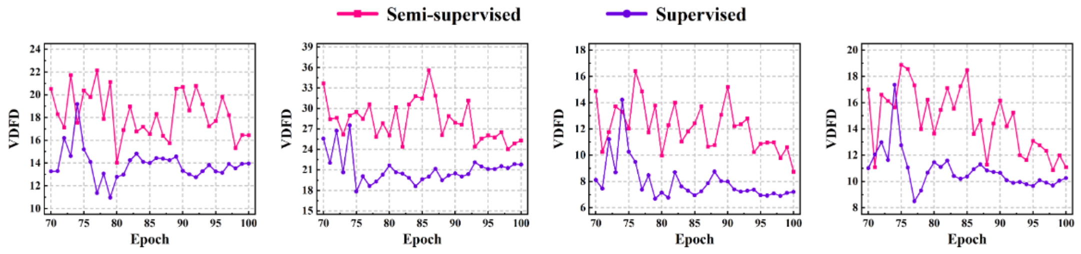
02

Variance Bias Problem

Variance Bias Problem

NEURAL INFORMATION PROCESSING SYSTEMS

In our preliminary experiments, we found self-training paradigms suffer from the **variance bias problem** by using the labeled and pseudo-labeled samples in the context of SSMLL, since it is **difficult to guarantee accurate enough pseudo-labels.**



The **variance difference between feature distributions (VDFD)** of positive and negative samples computed in semi-supervised and supervised manners in VOC2012.

NEURAL INFORMATION PROCESSING SYSTEMS

How Variance Bias Affects the Performance?

We set the ratio of positive and negative sample variances to **M**.

Given a SSBC dataset with pseudo-labels $\boldsymbol{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i^*)\} \cup \{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)\}$, the optimal linear classifier f_{ssl} minimizing the average standard classification error is given by:

$$f_{ssl} = \operatorname{argmin}_f \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}}[1(f(\mathbf{x}) \neq y)]$$

When $M > 1$, it has the intra-class standard classification errors for the two classes :

$$\mathcal{R}(f_{ssl}, +1) = \Phi(A - M\sqrt{A^2 + q(M, \alpha, \epsilon_-, \epsilon_+)}),$$
$$\mathcal{R}(f_{ssl}, -1) = \Phi(-M \cdot A + \sqrt{A^2 + q(M, \alpha, \epsilon_-, \epsilon_+)})$$

and when $M < 1$, they are given by:

$$\mathcal{R}(f_{ssl}, +1) = \Phi(A + M\sqrt{A^2 + q(M, \alpha, \epsilon_-, \epsilon_+)}),$$
$$\mathcal{R}(f_{ssl}, -1) = \Phi(-M \cdot A - \sqrt{A^2 + q(M, \alpha, \epsilon_-, \epsilon_+)})$$

We employ variance of class-wise accuracy **(VCA)** to quantitatively measure the model fairness. the variance of class-wise accuracy VCA(f_{ssl}) is increasing when $M \to \infty$ for $M > 1$ and $M \to 0$ for $M < 1$. Suppose $\log(\frac{\alpha(2-\epsilon_- - 2\epsilon_+)}{(1-\alpha)(2-2\epsilon_- - \epsilon_+)}) = 0$, then when $M = 1$, $R(f_{ssl}, +1) = R(f_{ssl}, -1)$ and VCA(f_{ssl}) = 0.

CHAPTER

03

$S^2ML^2 - BBAM$ Method

NEURAL INFORMATION PROCESSING SYSTEMS

Main Contributions

(1) We develop a novel SSMLL method, namely $S^2ML^2 - BBAM$, by balancing the variance bias between positive and negative samples from the perspective of the feature angle distribution for each label.

(2) We design a new BBAM loss by extending the traditional binary angular margin loss with feature angle distribution transformations under the Gaussian assumption.

(3) We suggest an efficient prototype-based negative sampling method to maintain high-quality negative samples for each label.

(4) We construct extensive experiments to evaluate $S^2ML^2 - BBAM$.

$S^2ML^2 - BBAM$ Method

We propose a novel Balanced Binary Angular Margin (BBAM) loss ℓBBAM(·, ·), aiming to balance the variance bias of positive and negative samples for each label from the feature angle distribution perspective with the Gaussian assumption.

$$\mathcal{L}(\mathbf{W}) = \frac{1}{B_l K} \sum_{i=1}^{B_l} \sum_{k=1}^{K} \beta_{ik} \ell_{\text{BBAM}}(p_{ik}^l, y_{ik}^l) + \frac{\lambda}{B_u K} \sum_{i=1}^{B_u} \sum_{k=1}^{K} \beta_{ik} \ell_{\text{BBAM}}(p_{ik}^u, y_{ik}^u)$$

$$\beta_{ik} = \begin{cases} 1 & \text{if } (\mathbf{x}_i, \mathbf{y}_i) \in \Omega_k; \\ 1 & \text{if } y_{ik} = 1; \\ 0 & \text{otherwise,} \end{cases} \quad \forall k \in [K], \ \forall i \in [N_l] \text{ or } [N_u]$$

Ωk denotes high-quality negative sample sets constructed by negative sampling.

Pseudo-labels of unlabeled data $\{y_i^u\}_{i=1}^{i=N_u}$ are produced by employing the Class-Aware Pesudo-labeling (CAP) trick.

$S^2ML^2 - BBAM$ Method

BBAM loss is extended from the Binary Angular Margin (BAM) loss, which measures the label-specific prediction risk by using the angle between the latent feature and boundary.

$$\ell_{\mathrm{BAM}}(p_{ik}, y_{ik}) = \begin{cases} -\log\left(\frac{1}{1+e^{-s*(p_{ik}-m)}}\right) & \text{if } y_{ik} = 1; \\ -\log\left(1 - \frac{1}{1+e^{-s*(p_{ik}-m)}}\right) & \text{if } y_{ik} = 0 \end{cases}$$

$$p_{ik} = \cos(\theta_{ik}) = \frac{\mathbf{z}_i^\top \mathbf{W}_k^c}{\|\mathbf{z}_i\|_2 \|\mathbf{W}_k^c\|_2}$$

$$\ell_{\mathrm{BBAM}}(p_{ik}, y_{ik}) = \begin{cases} -\log\left(\frac{1}{1+e^{-s*(\cos(\psi_k^{(p)}(\theta_{ik}))-m)}}\right) & \text{if } y_{ik} = 1; \\ -\log\left(1 - \frac{1}{1+e^{-s*(\cos(\psi_k^{(n)}(\theta_{ik}))-m)}}\right) & \text{if } y_{ik} = 0. \end{cases}$$

To address the previously mentioned variance bias, for each category k, we suppose that label angles of its positive samples and ones of its negative samples are drawn from a label-specific "positive" Gaussian distribution $\mathcal{N}\left(\mu_k^{(p)}, (\sigma_k^2)^{(p)}\right)$ and a label-specific "negative" one $\mathcal{N}\left(\mu_k^{(p)}, (\sigma_k^2)^{(p)}\right)$, respectively.

According to the properties of Gaussian distribution, we can easily transfer their variance to consistency.

$$\psi_k^{(p)}(\theta_{ik}) = a_k^{(p)}\theta_{ik} + b_k^{(p)}, \quad \psi_k^{(n)}(\theta_{ik}) = a_k^{(n)}\theta_{ik} + b_k^{(n)},$$

$$a_k^{(p)} = \frac{\widehat{\sigma}_k}{\sigma_k^{(p)}}, \quad b_k^{(p)} = (1 - a_k^{(p)})\mu_k^{(p)},$$

$$a_k^{(n)} = \frac{\widehat{\sigma}_k}{\sigma_k^{(n)}}, \quad b_k^{(n)} = (1 - a_k^{(n)})\mu_k^{(n)}, \quad \forall k \in [K].$$

$$\widehat{\sigma}_k^2 = \frac{(\sigma_k^2)^{(p)} + (\sigma_k^2)^{(n)}}{2}$$

$S^2ML^2 - BBAM$ Method

We approximate $\{(\mu_k^{(p)}, (\sigma_k^2)^{(p)})\}_{k=1}^{k=K}$ and $\{(\mu_k^{(n)}, (\sigma_k^2)^{(n)})\}_{k=1}^{k=K}$ with labeled and pseudo-labeled samples per-epoch.

The label angles between label prototypes and latent features of samples are given by:

$$\phi_{ik} = \arccos(\frac{\mathbf{z}_i{}^\top \mathbf{c}_k}{\|\mathbf{z}_i\|_2 \|\mathbf{c}_k\|_2}),\ \forall k \in [K],\ \forall i \in [N_l + N_u]$$

c_k is label prototype and z_i is latent feature.

Accordingly, the estimations of above can be formulated as:

$$\mu_k^{(p)} = \frac{\sum_{i=1}^{N_l+N_u} \mathbb{I}(y_{ik}=1)\phi_{ik}}{\sum_{i=1}^{N_l+N_u} \mathbb{I}(y_{ik}=1)}, \qquad \mu_k^{(n)} = \frac{\sum_{i=1}^{N_l+N_u} \beta_{ik}\mathbb{I}(y_{ik}=0)\phi_{ik}}{\sum_{i=1}^{N_l+N_u} \beta_{ik}\mathbb{I}(y_{ik}=0)},$$

$$(\sigma_k^2)^{(p)} = \frac{\sum_{i=1}^{N_l+N_u} \mathbb{I}(y_{ik}=1)(\phi_{ik}-\mu_k^{(p)})^2}{\sum_{i=1}^{N_l+N_u} \mathbb{I}(y_{ik}=1)-1} \qquad (\sigma_k^2)^{(n)} = \frac{\sum_{i=1}^{N_l+N_u} \beta_{ik}\mathbb{I}(y_{ik}=0)(\phi_{ik}-\mu_k^{(n)})^2}{\sum_{i=1}^{N_l+N_u} \beta_{ik}\mathbb{I}(y_{ik}=0)-1}$$

NEURAL INFORMATION PROCESSING SYSTEMS

To further alleviate the imbalance issue of positive and negative samples for each category, we also introduce negative sampling techniques.

For each category, we measure similarity scores of negative samples based on label prototypes, and construct a combination based on selecting the nearest negative samples.

$$\widetilde{\Omega}_k = \{(\mathbf{x}_i, \mathbf{y}_i) | d(\mathbf{z}_i, \mathbf{c}_k) \in \text{Rank}(\{d(\mathbf{z}_i, \mathbf{c}_k)\}_{(\mathbf{x}_i, \mathbf{y}_i) \in \widehat{\Omega}_k}) \quad (\mathbf{x}_i, \mathbf{y}_i) \in \widehat{\Omega}_k\} \quad \forall k \in [K],$$

$$\widehat{\Omega}_k = \{(\mathbf{x}_i^l, \mathbf{y}_i^l) | (\mathbf{x}_i^l, \mathbf{y}_i^l) \in \mathcal{D}_l, y_{ik}^l = 0\} \cup \{(\mathbf{x}_i^u, \mathbf{y}_i^u) | \mathbf{x}_i^u \in \mathcal{D}_u, y_{ik}^u = 0\}$$

The final negative sample sets are generated by :

$$\Omega_k = \{(\mathbf{x}_i, \mathbf{y}_i) | (\mathbf{x}_i, \mathbf{y}_i) \sim \text{Uniform}(\widetilde{\Omega}_k)\} \quad \forall k \in [K],$$

with size: $\{|\Omega_k| = \eta N_k\}_{k=1}^{k=K}$, where $N_k = \sum_{i=1}^{N_l} \mathbb{I}(y_{ik}^l = 1) + \sum_{i=1}^{N_u} \mathbb{I}(y_{ik}^u = 1)$

η controls the proportion of positive and negative samples of each category.

NEURAL INFORMATION PROCESSING SYSTEMS

CHAPTER

04

Experimental Result

Experimental Result

Table 2: Experimental results on images datasets. The best results are highlighted in boldface.

Method	VOC																			
	Micro-F1↑				Macro-F1↑				mAP↑				Hamming Loss↓				One Loss↓			
	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$
SoftMatch	0.6542	0.7187	0.7461	0.7484	0.5868	0.6630	0.6931	0.6876	0.6295	0.7235	0.7721	0.7867	0.0594	0.0368	0.0319	0.0294	0.4398	0.1655	0.1308	0.1148
FlatMatch	0.6493	0.7038	0.7420	0.7465	0.5344	0.6313	0.6666	0.6597	0.6468	0.7430	0.7923	0.8022	0.0386	0.0322	0.0313	0.0290	0.1983	0.1366	0.1238	0.1097
MIME	0.3650	0.6607	0.7013	0.7021	0.2439	0.5442	0.6425	0.5898	0.6653	0.7553	0.8090	0.8137	0.0546	0.0407	0.0336	0.0333	0.2099	0.1218	0.0835	0.0949
DRML	0.6450	0.6525	0.7274	0.7525	0.5660	0.5339	0.6864	0.7495	0.6058	0.6852	0.7131	0.7272	0.0564	0.0518	0.0377	0.0381	0.3542	0.2888	0.1720	0.1512
CAP	0.6162	0.6573	0.6798	0.7073	0.5822	0.6308	0.6536	0.6636	0.7616	0.8216	0.8348	**0.8460**	0.0801	0.0675	0.0622	0.0591	0.1303	0.0918	0.0827	**0.0755**
S^{2}ML^2-BBAM	**0.7897**	**0.8401**	**0.8443**	**0.8458**	**0.7306**	**0.8015**	**0.8124**	**0.8141**	**0.7866**	**0.8345**	**0.8454**	0.8458	**0.0310**	**0.0259**	**0.0243**	**0.0233**	**0.1087**	**0.0867**	**0.0817**	0.0795

Method	COCO																			
	Micro-F1↑				Macro-F1↑				mAP↑				Hamming Loss↓				One Loss↓			
	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$
SoftMatch	0.5763	0.6273	0.6487	0.6676	0.4283	0.5265	0.5493	0.5830	0.5624	0.6194	0.6395	0.6622	0.0235	0.0218	0.0211	0.0205	0.1293	0.0948	0.0844	0.0879
FlatMatch	0.5960	0.6389	0.6590	0.6720	0.4794	0.5341	0.5710	0.5870	0.5827	0.6335	0.6542	0.6654	**0.0227**	0.0213	0.0208	0.0203	0.1215	0.1002	0.0933	0.0878
MIME	0.2982	0.4378	0.4906	0.5323	0.2557	0.3731	0.4096	0.4545	0.5372	0.5991	0.6379	0.6633	0.0302	0.0265	0.0250	0.0236	0.1495	0.1110	0.0883	0.0799
DRML	0.6071	0.6226	0.6492	0.6486	0.5345	0.5604	0.5779	0.5867	0.5118	0.5461	0.6026	0.6177	0.0242	0.0240	0.0230	0.0223	0.1438	0.1288	0.1243	0.1039
CAP	0.5629	0.5657	0.5724	0.5696	0.5230	0.5306	0.5402	0.5416	0.6243	0.6736	**0.6911**	**0.7041**	0.0523	0.0512	0.0499	0.0558	0.1004	**0.0841**	**0.0788**	**0.0726**
S^{2}ML^2-BBAM	**0.6830**	**0.7074**	**0.7150**	**0.7246**	**0.6144**	**0.6480**	**0.6594**	**0.6726**	**0.6354**	**0.6741**	0.6886	0.7023	0.0230	**0.0212**	**0.0206**	**0.0201**	**0.1000**	0.0878	0.0824	0.0799

Method	AWA																			
	Micro-F1↑				Macro-F1↑				mAP↑				Hamming Loss↓				One Loss↓			
	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$
SoftMatch	0.6992	0.6973	0.7024	0.7024	0.5476	0.5284	0.5524	0.5457	0.6368	0.6524	0.6494	0.6518	0.2160	0.2155	0.2132	0.2126	0.1580	0.08876	0.1494	0.1549
FlatMatch	0.6918	0.6977	0.6989	0.7013	0.5221	0.5487	0.5507	0.5636	0.6393	0.6459	0.6565	0.6577	0.2190	0.2167	0.2165	0.2164	0.1029	0.0936	0.1116	0.1162
MIME	0.1470	0.3889	0.4893	0.4090	0.0705	0.1830	0.2659	0.2327	0.3992	0.3803	0.4762	0.5265	0.3570	0.3290	0.3064	0.3012	0.1850	0.2091	0.1664	0.2004
DRML	0.6827	0.6856	0.6942	0.6893	0.5399	0.5541	0.5727	0.5618	0.6160	0.6246	0.6377	0.6338	0.2285	0.2270	0.2226	0.2258	0.1360	0.1801	0.2609	0.1839
CAP	0.6868	0.7065	0.7091	0.7099	0.5742	0.5864	0.5905	0.5914	0.6390	0.6415	0.6440	0.6451	0.3120	0.2727	0.2589	0.2617	0.1146	0.0933	0.1045	0.1199
S^{2}ML^2-BBAM	**0.7213**	**0.7255**	**0.7215**	**0.7279**	**0.5853**	**0.5914**	**0.5905**	**0.5944**	**0.6419**	0.6463	0.6416	0.6476	**0.2091**	**0.2060**	**0.2109**	**0.2042**	0.1206	0.1103	0.1149	0.1188

Table 3: Experimental results on text datasets. The best results are highlighted in boldface.

Method	Ohsumed																			
	Micro-F1↑				Macro-F1↑				mAP↑				Hamming Loss↓				One Loss↓			
	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$
SoftMatch	0.4769	0.4478	0.4462	0.4449	0.3056	0.2366	0.2348	0.2229	0.4664	0.5106	0.5218	0.5392	0.0756	0.0798	0.0801	0.0803	0.4213	0.5036	0.5274	0.5140
FlatMatch	0.5161	0.4836	0.4254	0.4472	0.3073	0.2262	0.1904	0.1775	0.4187	0.4751	0.4993	0.5139	0.0699	0.0747	0.0831	0.0799	0.3943	0.4416	0.5824	0.5008
DRML	0.3975	0.4015	0.4185	0.4055	0.1903	0.1972	0.1996	0.2070	0.1833	0.1931	0.2083	0.2140	0.0939	0.0868	0.0873	0.0851	0.6020	0.5677	0.5760	0.5496
CAP	0.5562	0.5776	0.5819	0.5455	0.4743	0.5144	0.5285	0.5214	0.4722	0.5370	0.5740	0.5995	0.0678	0.0840	0.0752	0.0967	0.3237	0.2746	0.2541	0.2493
S^{2}ML^2-BBAM	**0.6671**	**0.7100**	**0.7196**	**0.7550**	**0.6058**	**0.6515**	**0.6719**	**0.7120**	**0.5537**	**0.6345**	**0.6604**	**0.6884**	**0.0467**	**0.0409**	**0.0243**	**0.0346**	**0.2417**	**0.2186**	**0.2068**	**0.1710**

Method	AAPD																			
	Micro-F1↑				Macro-F1↑				mAP↑				Hamming Loss↓				One Loss↓			
	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 5\%$	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$
SoftMatch	0.3345	0.3325	0.3325	0.3279	0.0612	0.0514	0.0520	0.0481	0.3753	0.3949	0.4084	0.3990	0.0596	0.0598	0.0598	0.0602	0.6630	0.6630	0.6630	0.6627
FlatMatch	0.3221	0.3147	0.3155	0.3155	0.0519	0.0439	0.0437	0.0437	0.3571	0.3706	0.3570	0.3621	0.0607	0.0614	0.0613	0.0613	0.6629	0.6631	0.6635	0.6634
DRML	0.4160	0.4101	0.4027	0.4130	0.1024	0.1005	0.0998	0.1052	0.1465	0.1538	0.1579	0.1591	0.0545	0.0578	0.0521	0.0542	0.5450	0.5910	0.5280	0.5430
CAP	0.5722	0.5726	0.5504	0.5026	0.3917	0.4310	0.4257	0.4051	0.4095	0.4696	0.4899	0.4932	0.0432	0.0498	0.0571	0.0742	0.3010	0.2461	0.2523	0.2384
S^{2}ML^2-BBAM	**0.7057**	**0.7279**	**0.7312**	**0.7316**	**0.5091**	**0.5825**	**0.5706**	**0.5823**	**0.5153**	**0.5903**	**0.5804**	**0.5930**	**0.0262**	**0.0241**	**0.0238**	**0.0238**	**0.1821**	**0.1500**	**0.1550**	**0.1590**

Overall, our method achieved good performance on all five metrics.

Our model ranks first on average on five datasets and has a significant advantage over other methods.

The comparison between macro-f1 and micro-f1 show that the fairness has been improved after apply our method.

Experimental Result

Table 4: Results of the ablative study on *VOC2012* and *COCO*.

Metric	VOC							
	$\pi = 5\%$		$\pi = 10\%$		$\pi = 15\%$		$\pi = 20\%$	
	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM
Micro-F1	**0.7897**	0.7845	**0.8401**	0.8206	**0.8443**	0.8301	**0.8458**	0.8318
Macro-F1	**0.7306**	0.7247	**0.8015**	0.7789	**0.8124**	0.7988	**0.8141**	0.7967
mAP	0.7866	**0.7881**	**0.8345**	0.8204	**0.8454**	0.8274	**0.8458**	0.8282
Metric	COCO							
	$\pi = 5\%$		$\pi = 10\%$		$\pi = 15\%$		$\pi = 20\%$	
	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM	S^{2}ML^2-BBAM	w/o BBAM
Micro-F1	**0.6830**	0.6691	**0.7074**	0.6952	**0.7150**	0.7052	**0.7246**	0.7143
Macro-F1	**0.6144**	0.5885	**0.6480**	0.6264	**0.6594**	0.6424	**0.6726**	0.6530
mAP	**0.6354**	0.5894	**0.6741**	0.6316	**0.6886**	0.6520	**0.7023**	0.6628

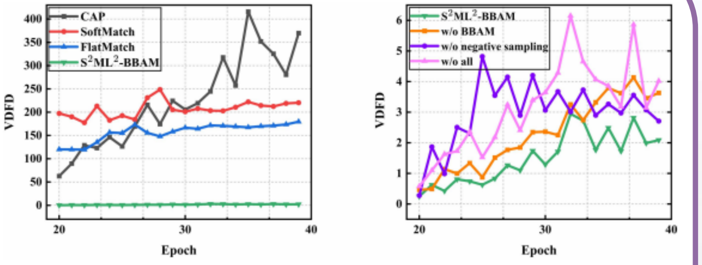


Figure 2: Comparison of VDFD on *VOC2012*.

The ablation study result indicate that our method can significantly reduce variance differences. And constraining the angle variance between positive and negative samples can effectively improve the accuracy.

Thank you for listening!