TEL AVIV 2022

Paper Code

Acknowledging the Unknown for Multi-label Learning with Single Positive Labels

European Conference on Computer Vision (ECCV) 2022

Donghao Zhou[1,2], Pengfei Chen[3], Qiong Wang[1], Guangyong Chen[4*], Pheng-Ann Heng[1,5]

[1] SIAT, CAS [2] UCAS [3] Tencent [4] Zhejiang Lab [5] CUHK

Background

➢ **What is multi-label learning?**

Multi-class Classification

cat

VS

Multi-label Learning

person, *bus*, *bicycle*

Many worth-exploring variants:

- Extremely Multi-label Learning
- Partial Multi-Label Learning
- Multi-Label Active Learning
- Semi-supervised Multi-label Learning

⋮

Background

➢ **What is single positive multi-label learning (SPML)?**

	person	*dog*	*bus*	*bicycle*	*apple*	*boat*	*laptop*	*couch*	
(a)	✓	×	✓	✓	×	×	×	×	➡ **Multi-label Learning**
(b)	✓	×	?	✓	?	×	?	?	➡ **Multi-label Learning with Missing Labels**
(c)	✓	?	?	?	?	?	?	?	➡ **SPML**

✓: positive labels ×: negative labels ?: unknown Labels

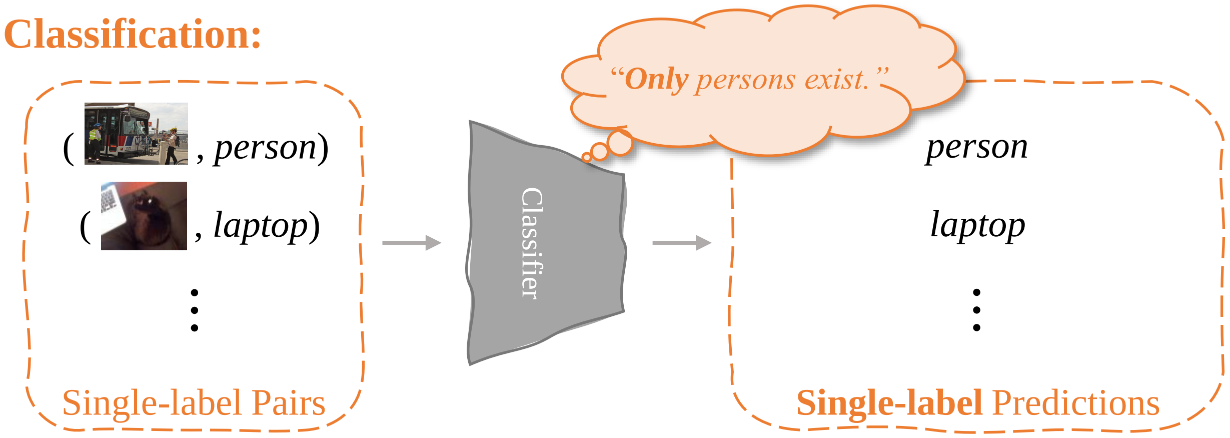
Only **one single positive label** is annotated for each training image.

Background

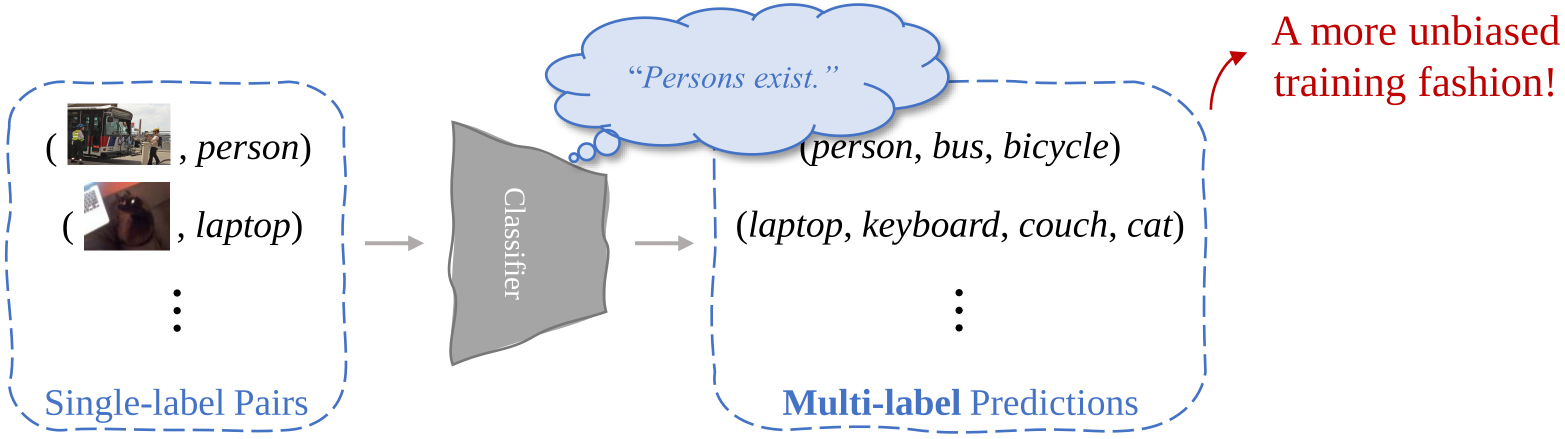
➢ **What is single positive multi-label learning (SPML)?**

🤔 Learn a **multi-label** classifier from a **single-label** dataset!

- **Multi-class Classification:**

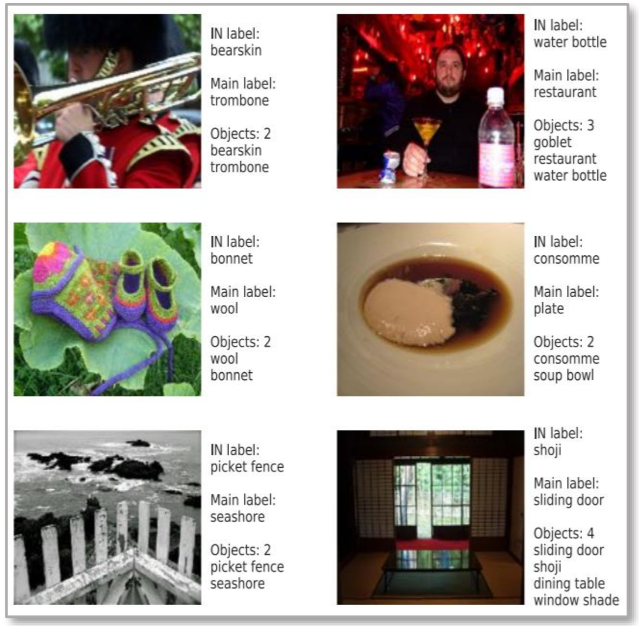


- **SPML:**



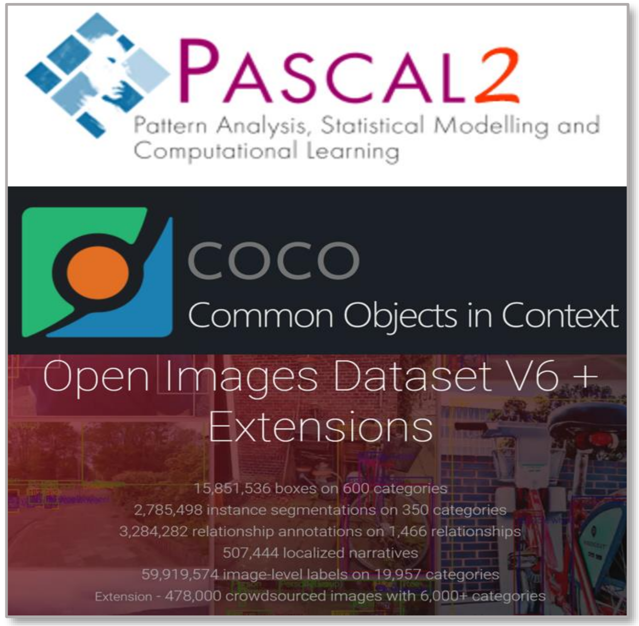
Background

➢ **Why study single positive multi-label learning (SPML)?**



Some multi-class datasets like ImageNet are found to being multi-label. †

Applies to many real-world scenarios (*e.g.* medical diagnosis).



Helps to relax the annotation requirements for multi-label datasets.

† Dimitris Tsipras, et al., "From ImageNet to Image Classification: Contextualizing Progress on Benchmarks", ICML, 2020.

Naive Solutions

Labels

$y_c^{(n)}=1$: positive label

$y_c^{(n)}=-1$: negative label

$y_c^{(n)}=0$: unannotated label

- **Trained with only positive labels (Infeasible!)**

$$\mathcal{L}\left(\underbrace{\mathbf{f}^{(n)}}_{\text{predicted probabilities}}, \underbrace{\mathbf{y}^{(n)}}_{\text{“single positive” labels}}\right) = -\underbrace{\frac{1}{C}}_{\text{the number of classes}}\sum_{c=1}^{C}[\underbrace{\mathbb{1}_{[y_c^{(n)}=1]}}_{\text{the indicator function}} \log(f_c^{(n)})]$$

It would collapse to a trivial solution.

- **Trained with positive labels and assumed negative labels** †

$$\mathcal{L}_{\mathrm{AN}}(\mathbf{f}^{(n)}, \mathbf{y}^{(n)}) = -\frac{1}{C}\sum_{c=1}^{C}[\mathbb{1}_{[y_c^{(n)}=1]} \log(f_c^{(n)}) + \mathbb{1}_{[y_c^{(n)}=0]} \log(1 - f_c^{(n)})]$$

Good intuition! Because Negative labels are the **overwhelming majority** of multi-label Annotations. It can serve as a **baseline** of SPML.

† Elijah Cole, et al., “Multi-Label Learning from Single Positive Labels”, CVPR, 2021.

Take a Deep Look

Notations

p : predicted probability

g : output logit

➢ Assuming-Negative (AN) Loss

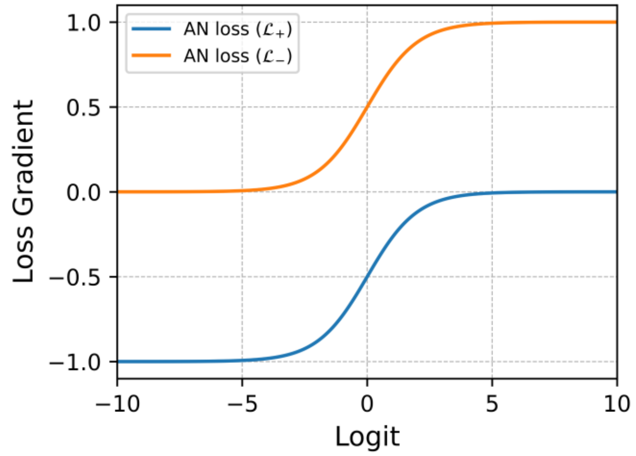
$$\mathcal{L}_{\mathrm{AN}}(\mathbf{f}^{(n)}, \mathbf{y}^{(n)}) = -\frac{1}{C}\sum_{c=1}^{C}[\mathbb{1}_{[y_c^{(n)}=1]}\log(f_c^{(n)}) + \mathbb{1}_{[y_c^{(n)}=0]}\log(1-f_c^{(n)})]$$

➢ Gradient Regime of AN Loss

$$\mathcal{L}_+ = -\log(p)$$
$$\mathcal{L}_- = -\log(1-p)$$

$$\Rightarrow \begin{cases} \dfrac{\partial \mathcal{L}_+}{\partial g} = \dfrac{\partial \mathcal{L}_+}{\partial p}\dfrac{\partial p}{\partial g} = \dfrac{-e^{-g}}{1+e^{-g}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_-}{\partial g} = \dfrac{\partial \mathcal{L}_-}{\partial p}\dfrac{\partial p}{\partial g} = \dfrac{1}{1+e^{-g}}, & y_c^{(n)} = 0 \end{cases}$$

the same gradient regimes



➢ What's wrong?

1. Dominance of Assumed Negative Labels
2. Introduced Label Noise
3. Over-Suppression for Confident Positive Predictions

Unannotated labels need to be properly treated during training, or more specifically, be treated with a **better gradient regime.**

Acknowledging the Unknown

Making any unrealistic assumptions would confuse the model. How about **acknowledging the fact that these unannotated labels are unknown**?

- **Entropy-Maximization (EM) Loss**

$$\mathcal{L}_{\mathrm{EM}}(\mathbf{f}^{(n)}, \mathbf{y}^{(n)}) = -\frac{1}{C}\sum_{c=1}^{C}[\mathbb{1}_{[y_c^{(n)}=1]} \log(f_c^{(n)}) + \mathbb{1}_{[y_c^{(n)}=0]} \alpha H(f_c^{(n)})]$$

$$H(f_c^{(n)}) = -[f_c^{(n)} \log(f_c^{(n)}) + (1 - f_c^{(n)}) \log(1 - f_c^{(n)})]$$

We maximize the entropy of predicted probabilities for unannotated labels.

- **Gradient Regime of EM Loss**

the same gradient regime as AN loss

$$\mathcal{L}_{+} = -\log(p)$$
$$\mathcal{L}_{\varnothing} = \alpha[p \log p + (1-p)\log(1-p)]$$

$$\Rightarrow \begin{cases} \dfrac{\partial \mathcal{L}_{+}}{\partial g} = \dfrac{\partial \mathcal{L}_{+}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-e^{-g}}{1+e^{-g}}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial g} = \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-\alpha e^{-g}\log e^{-g}}{(1+e^{-g})^2}}, & y_c^{(n)} = 0 \end{cases}$$

a quite different one

Acknowledging the Unknown

➢ Gradient Regime of EM Loss

$$\mathcal{L}_+ = -\log(p)$$
$$\mathcal{L}_\varnothing = \alpha[p\log p + (1-p)\log(1-p)]$$

$$\Rightarrow \begin{cases} \dfrac{\partial \mathcal{L}_+}{\partial g} = \dfrac{\partial \mathcal{L}_+}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-e^{-g}}{1+e^{-g}}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_\varnothing}{\partial g} = \dfrac{\partial \mathcal{L}_\varnothing}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-\alpha e^{-g}\log e^{-g}}{(1+e^{-g})^2}}, & y_c^{(n)} = 0 \end{cases}$$

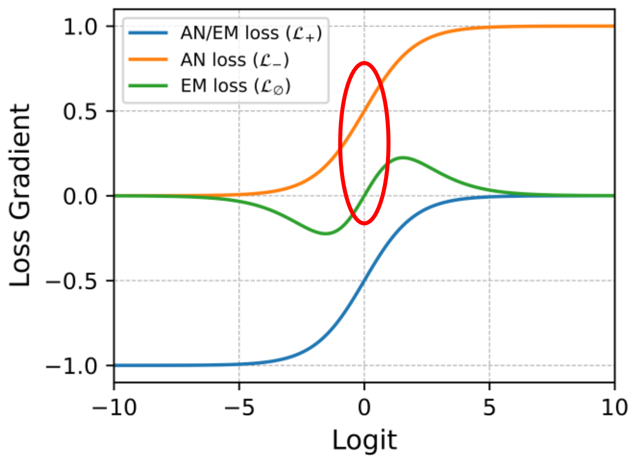
the same gradient regime as AN loss

a quite different one

➢ What can EM loss help?

1. Learning from Annotated Labels Preferentially

In early training, EM loss can provide small gradients for the **ambiguous predictions** of unannotated labels. EM loss tends to keep these ambiguous predictions, and thus is capable of providing small gradients for them **throughout training**.



Acknowledging the Unknown

➢ Gradient Regime of EM Loss

the same gradient regime as AN loss

$$\mathcal{L}_{+} = -\log(p)$$
$$\mathcal{L}_{\varnothing} = \alpha[p\log p + (1-p)\log(1-p)]$$

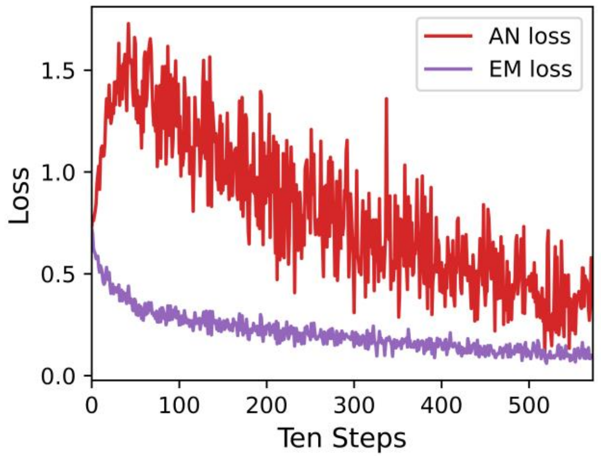
$$\Rightarrow \begin{cases} \dfrac{\partial \mathcal{L}_{+}}{\partial g} = \dfrac{\partial \mathcal{L}_{+}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-e^{-g}}{1+e^{-g}}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial g} = \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-\alpha e^{-g}\log e^{-g}}{(1+e^{-g})^2}}, & y_c^{(n)} = 0 \end{cases}$$

a quite different one

➢ What can EM loss help?

1. Learning from Annotated Labels Preferentially

In early training, EM loss can provide small gradients for the **ambiguous predictions** of unannotated labels. EM loss tends to keep these ambiguous predictions, and thus is capable of providing small gradients for them **throughout training**.



(*Training losses of annotated labels on PASCAL VOC*)

Acknowledging the Unknown

➢ Gradient Regime of EM Loss

$$\mathcal{L}_+ = -\log(p)$$
$$\mathcal{L}_\varnothing = \alpha[p\log p + (1-p)\log(1-p)]$$

the same gradient regime as AN loss

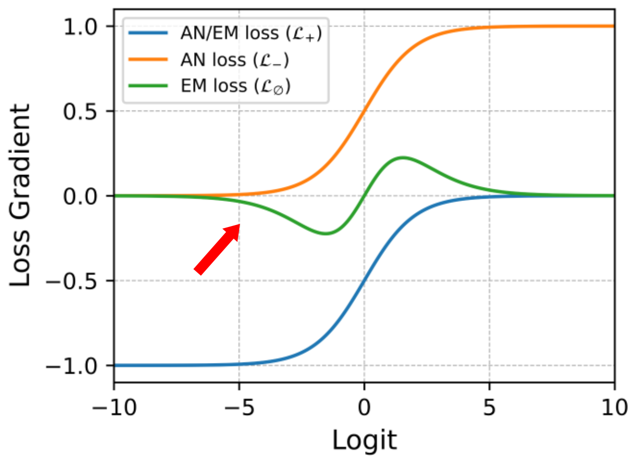
$$\begin{cases} \dfrac{\partial \mathcal{L}_+}{\partial g} = \dfrac{\partial \mathcal{L}_+}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-e^{-g}}{1+e^{-g}}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_\varnothing}{\partial g} = \dfrac{\partial \mathcal{L}_\varnothing}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-\alpha e^{-g}\log e^{-g}}{(1+e^{-g})^2}}, & y_c^{(n)} = 0 \end{cases}$$

a quite different one

➢ What can EM loss help?

2. Mitigating the Effect of Label Noise

There are **no false negative labels**, which prevents the model from producing incorrect negative predictions. Though unannotated positive labels still exist, the model trained with EM loss would **mainly focus on the annotated ones**.



Acknowledging the Unknown

➢ Gradient Regime of EM Loss

$$\mathcal{L}_{+} = -\log(p)$$
$$\mathcal{L}_{\varnothing} = \alpha[p\log p + (1-p)\log(1-p)]$$

$$\Rightarrow \begin{cases} \dfrac{\partial \mathcal{L}_{+}}{\partial g} = \dfrac{\partial \mathcal{L}_{+}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-e^{-g}}{1+e^{-g}}}, & y_c^{(n)} = 1 \\ \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial g} = \dfrac{\partial \mathcal{L}_{\varnothing}}{\partial p}\dfrac{\partial p}{\partial g} = \boxed{\dfrac{-\alpha e^{-g}\log e^{-g}}{(1+e^{-g})^2}}, & y_c^{(n)} = 0 \end{cases}$$

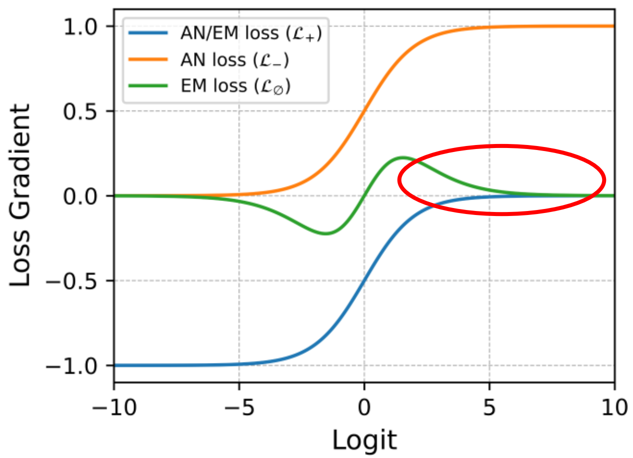
the same gradient regime as AN loss

a quite different one

➢ What can EM loss help?

3. Maintaining Confident Positive Predictions

When the logit is large enough, the gradients of unannotated labels would **decline and even approach zero** as the logit goes larger, which helps to maintain these confident positive predictions.



One More Step Forward

🤔 Taking full advantage of EM loss, can we provide **more precise supervision** for the model and then further improve its performance?

➢ "Tolerance" of Pseudo-Labeling

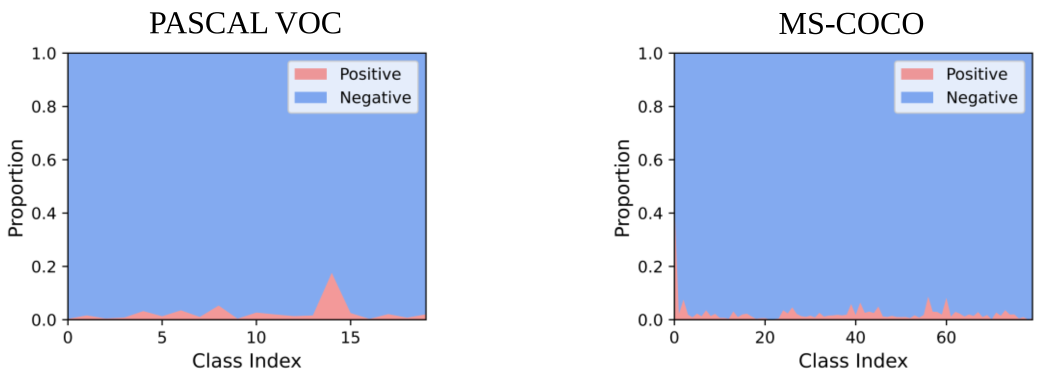
Low-Tolerance Strategy
(high score threshold or low sample proportion)

VS

High-Tolerance Strategy
(low score threshold or high sample proportion)

There is a nature trade-off between the provided supervision and the introduced noise.

➢ Issue: Positive-Negative Label Imbalance



(Proportions of unannotated positive and negative labels)

One More Step Forward

➢ Asymmetric Pseudo-Labeling (APL)

Low-Tolerance Strategy
(high score threshold or low sample proportion) → For positives (**do not generate any pseudo-labels**)

High-Tolerance Strategy
(low score threshold or high sample proportion) → For negatives (**adopt a 90% sample proportion**)

➢ Additional Tricks for Pseudo-Labeling

- Self-paced Procedure
- Soft Labels
- Reweighting

⋮

Algorithm 1 Asymmetric Pseudo-Labeling

Input: Training set $\mathcal{D}$ and model f_{T_w} trained with Eq. 3 for T_w epochs
Parameter: Total training epoch T_t, sample proportion $\theta\%$ and loss weight β
Output: Well-trained model f_i

$i \leftarrow T_\mathrm{w}$, $\theta'\% \leftarrow \theta\%/(T_\mathrm{t} - T_\mathrm{w})$
repeat
  Generate pseudo-labels using f_i by following Eq. 6
  Train f_{i+1} from f_i with Eq. 8
  $i \leftarrow i + 1$
until early stopping **or** $i = T_\mathrm{t}$
return f_i

Experiments

➢ Benchmark Results

Ann. Labels	Methods	VOC	COCO	NUS	CUB
All P. & All N.	BCE loss	89.42±0.27	76.78±0.13	52.08±0.20	30.90±0.64
1 P. & All N.	BCE loss	87.60±0.31	71.39±0.19	46.45±0.27	20.65±1.11
1 P. & 0 N.	AN loss	85.89±0.38	64.92±0.19	42.27±0.56	18.31±0.47
	DW	86.98±0.36	67.59±0.11	45.71±0.23	19.15±0.56
	L1R	85.97±0.31	64.44±0.20	42.15±0.46	17.59±1.82
	L2R	85.96±0.36	64.41±0.24	42.72±0.12	17.71±1.79
	LS	87.90±0.21	67.15±0.13	43.77±0.29	16.26±0.45
	N-LS	88.12±0.32	67.15±0.10	43.86±0.54	16.82±0.42
	EntMin	53.16±2.81	32.52±5.55	19.38±3.64	13.08±0.15
	Focal loss	87.59±0.58	68.79±0.14	47.00±0.14	19.80±0.30
	ASL	87.76±0.51	68.78±0.32	46.93±0.30	18.81±0.48
	ROLE	87.77±0.22	67.04±0.19	41.63±0.35	13.66±0.24
	ROLE+LI	88.26±0.21	69.12±0.13	45.98±0.26	14.86±0.72
1 P. & 0 N.	EM loss	*89.09±0.17*	*70.70±0.31*	*47.15±0.11*	*20.85±0.42*
	EM loss+APL	**89.19±0.31**	**70.87±0.23**	**47.59±0.22**	**21.84±0.34**

Oracles

AN Loss and Improved AN Loss

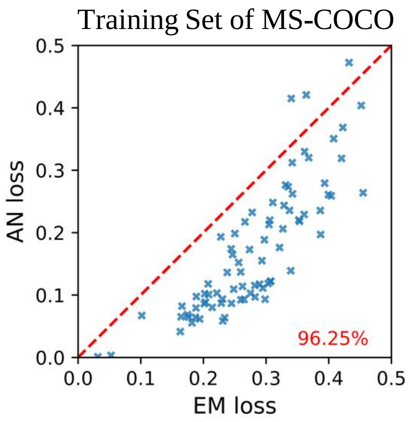
Other Comprising Methods

Ours

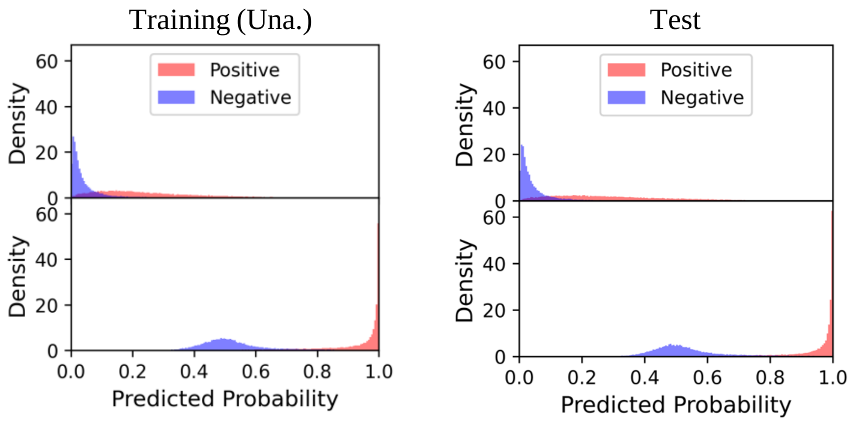
(*Experimental results with mAP on four SPML benchmarks*)

Further Analysis

➢ Distinguishability of Model Predictions

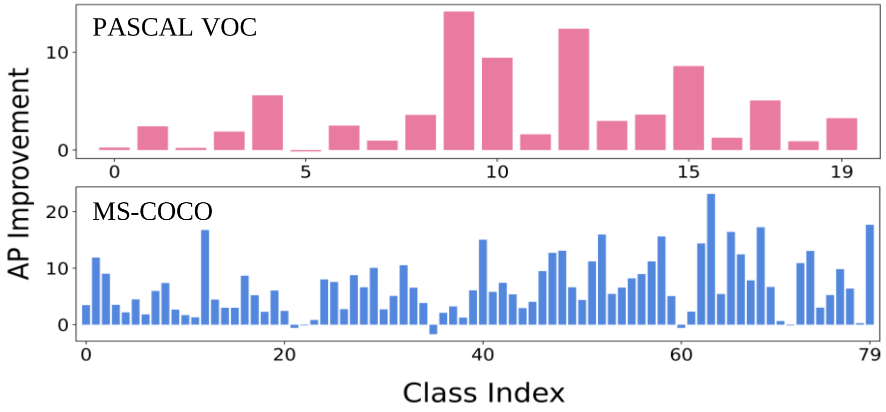


(*Wasserstein distances between the distributions of the predicted probabilities for unannotated positive and negative labels*)



(*Densities of predicted probabilities on the "person" class of MS-COCO*)

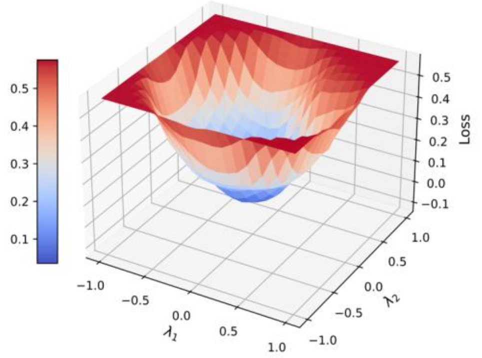
➢ Class-wise Performance Improvement



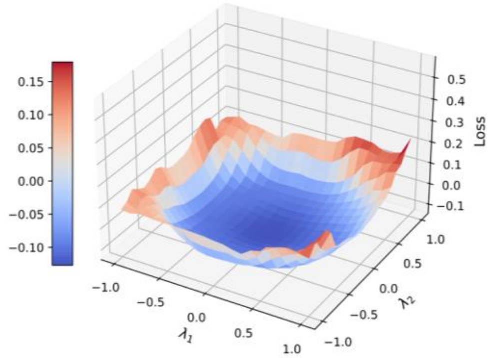
Further Analysis

➢ Generalization Evaluation by Loss Landscapes †

AN Loss

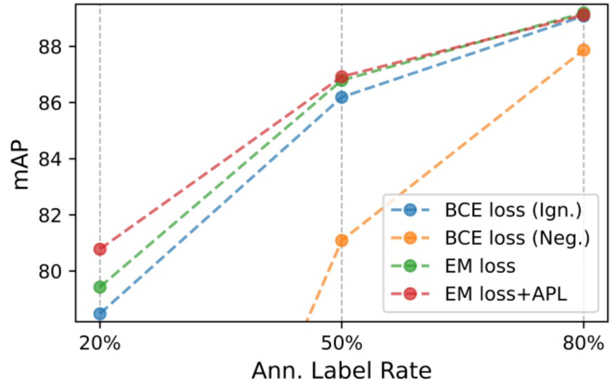


EM Loss



The model trained EM loss would converge to a flatter minimum, which contributes to better generalization.

➢ Performance in a More General Scenario (MLML)



Our method can be generalized to other similar tasks.

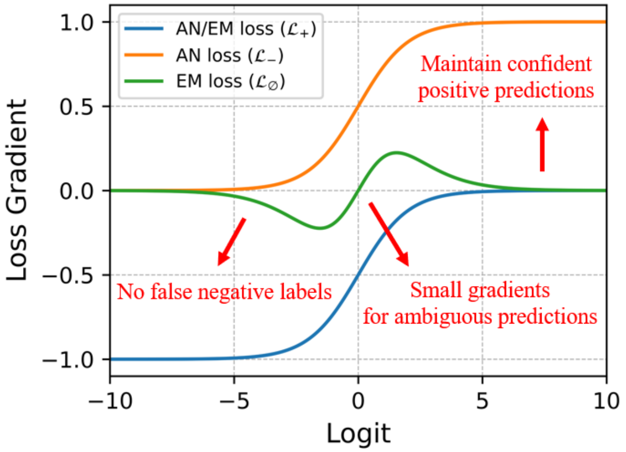
† Hao Li, et al., "Visualizing the Loss Landscape of Neural Nets", NeurIPS, 2018.

Further Analysis

➢ **Qualitative Results**

Highlights

EM loss



APL

Algorithm 1 Asymmetric Pseudo-Labeling

Input: Training set $\mathcal{D}$ and model f_{T_w} trained with Eq. 3 for T_w epochs
Parameter: Total training epoch T_t, sample proportion $\theta\%$ and loss weight β
Output: Well-trained model f_i

$i \leftarrow T_\mathrm{w}$, $\theta'\% \leftarrow \theta\%/(T_\mathrm{t} - T_\mathrm{w})$
repeat
  Generate pseudo-labels using f_i by following Eq. 6
  Train f_{i+1} from f_i with Eq. 8
  $i \leftarrow i + 1$
until early stopping **or** $i = T_\mathrm{t}$
return f_i

- This work focuses on **single positive multi-label learning**, an extreme of weakly supervised learning problem.
- we choose to treat all unannotated labels from a novel perspective, and hence propose our **entropy-maximization loss** (with a special gradient regime) and **asymmetric pseudo-labeling** (with asymmetric-tolerance strategies).
- Our method achieves **SOTA results on all four SPML benchmarks** and various analyses are provided to verify its effectiveness and rationality.

ECCV TEL AVIV 2022

Paper

Code

Thanks for Listening!

Q&A