Multi-Label Image Recognition with Graph Convolutional Networks

Zhao-Min Chen[1,2] Xiu-Shen Wei[2] Peng Wang[3] Yanwen Guo[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Megvii Research Nanjing, Megvii Technology, China
[3]School of Computer Science, The University of Adelaide, Australia

Introduction

Multi-Label Learning

Person, Sports Ball, Tennis Racket

Person, Tie

Person, Ski

- Label Dependency

  Label co-occurance is often used to model the dependency.

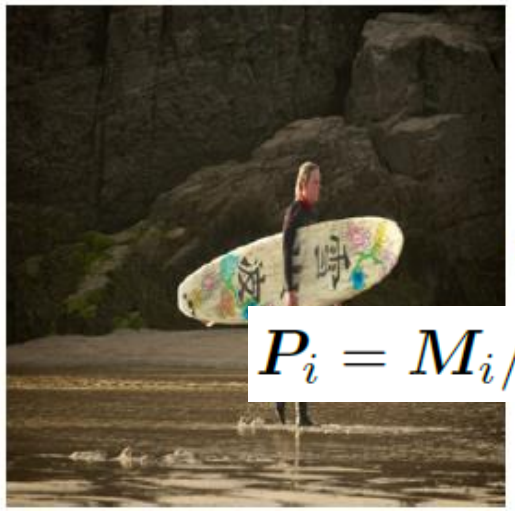
Method

Q1: How to effectively capture the correlation between object labels?

$P(L_j|L_i)$ → The probability of occurrence of label L_j when label L_i appears

Correlation Matrix

count the o
label pairs i

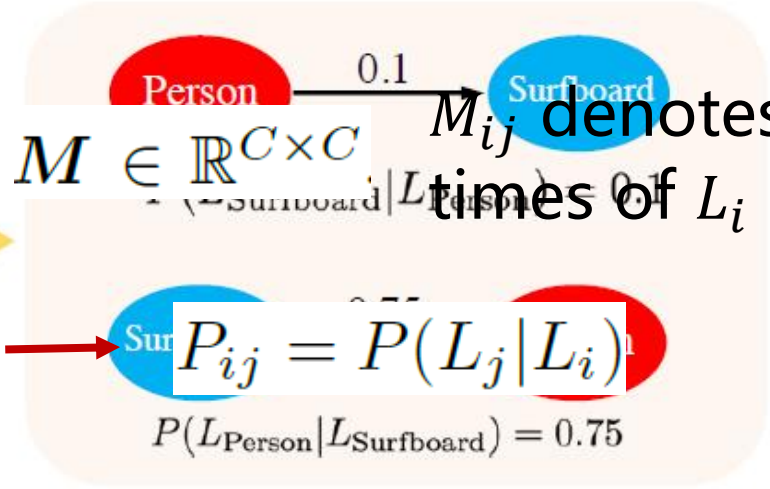


$\boldsymbol{M} \in \mathbb{R}^{C \times C}$

M_{ij} denotes the occurring times of L_i and L_j

$\boldsymbol{P}_i = \boldsymbol{M}_i / N_i$

$P_{ij} = P(L_j|L_i)$



The occurrence times of L_i in the training set

Method

Q1: How to effectively capture the correlation between object labels?

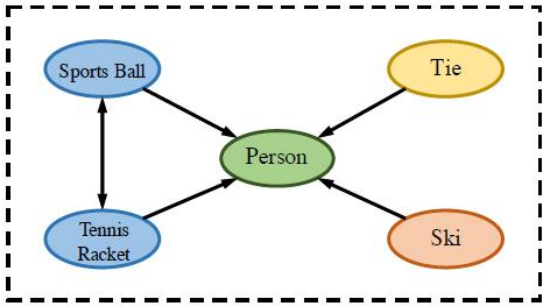
Binarize the correlation

$$\boldsymbol{A}_{ij} = \begin{cases} 0, & \text{if } \boldsymbol{P}_{ij} < \tau \\ 1, & \text{if } \boldsymbol{P}_{ij} \geq \tau \end{cases}$$

Re-weighted Scheme

$$\boldsymbol{A}'_{ij} = \begin{cases} p / \sum_{\substack{j=1 \\ i \neq j}}^{C} \boldsymbol{A}_{ij}, & \text{if } i \neq j \\ 1 - p, & \text{if } i = j \end{cases}$$

Person, Sports Ball, Tennis Racket | Person, Tie | Person, Ski



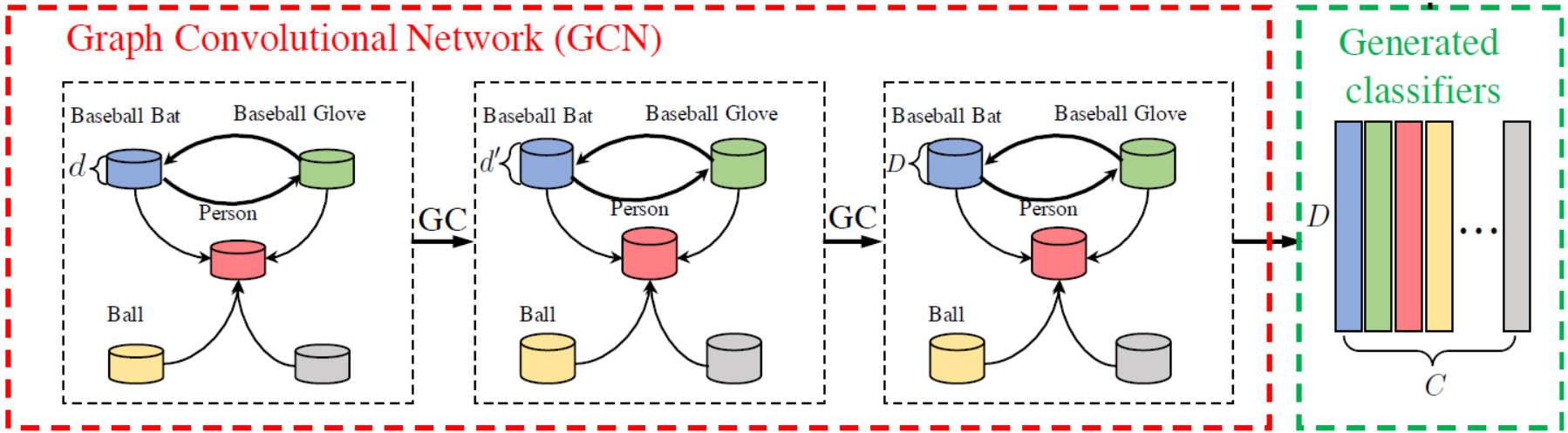
$$\text{Label}_\text{A} \rightarrow \text{Label}_\text{B}$$

means that when Lable_A appears, Lable_B is likely to appear, but the reverse may not be true

Method

Q2: How to effectively explore these label correlations to improve the classification performance?

Label → Word Embedding



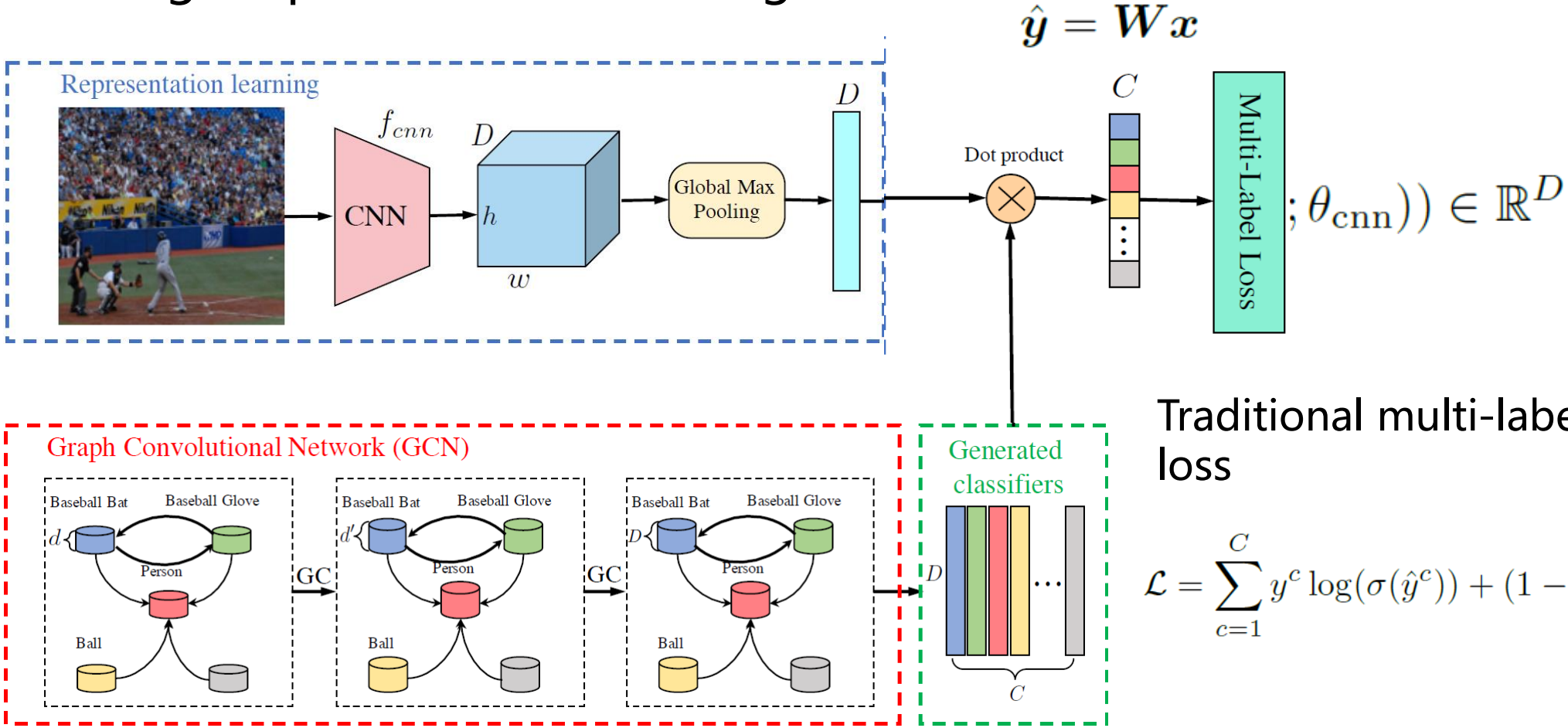
$$\boldsymbol{H}^{l+1} = f(\boldsymbol{H}^{l}, \boldsymbol{A}) \xrightarrow{\text{Conv}} \boldsymbol{H}^{l+1} = h(\widehat{\boldsymbol{A}}\boldsymbol{H}^{l}\boldsymbol{W}^{l})$$

Node

Transformation matrix to be learned

Method

Image representation learning



Traditional multi-label classification loss

$$\mathcal{L} = \sum_{c=1}^{C} y^c \log(\sigma(\hat{y}^c)) + (1 - y^c) \log(1 - \sigma(\hat{y}^c))$$

Experiments

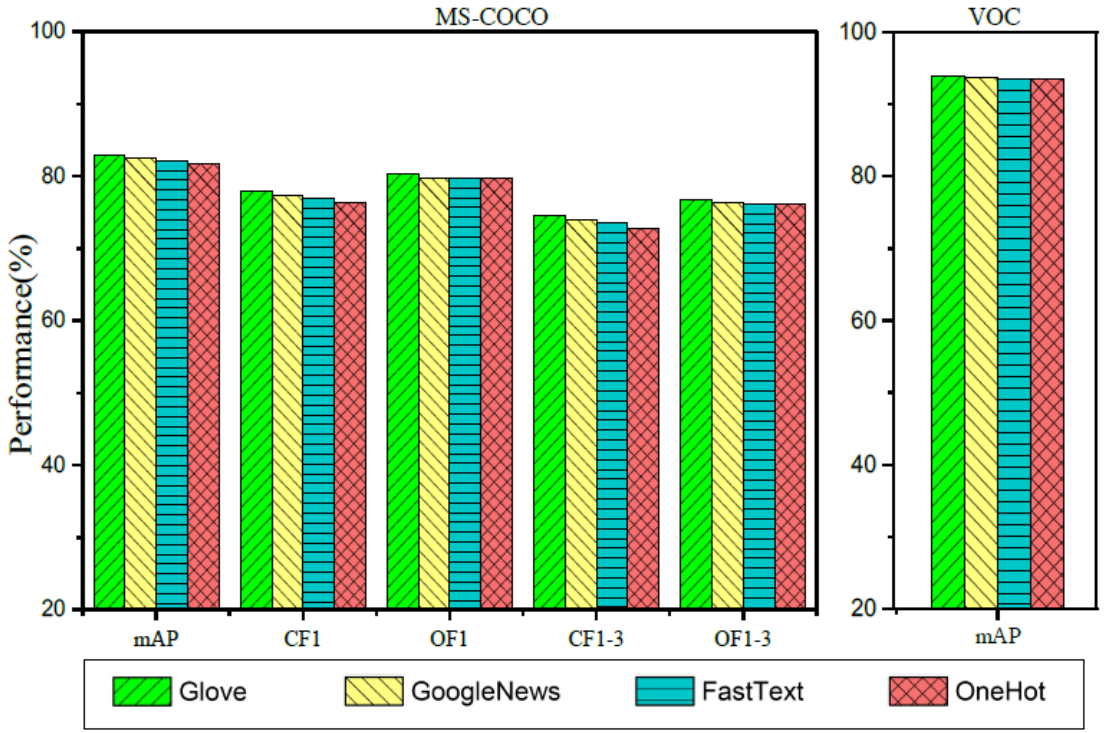
Methods	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN [28]	61.2	–	–	–	–	–	–	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention [29]	–	–	–	–	–	–	–	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN [1]	–	–	–	–	–	–	–	71.6	54.8	62.1	74.2	62.2	67.7
ML-ZSL [15]	–	–	–	–	–	–	–	74.1	**64.5**	69.0	–	–	–
SRN [36]	77.1	81.6	65.4	71.2	82.7	69.9	75.8	85.2	58.8	67.4	87.4	62.5	72.9
ResNet-101 [10]	77.3	80.2	66.7	72.8	83.9	70.8	76.8	84.1	59.4	69.7	89.1	62.8	73.6
Multi-Evidence [6]	–	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
ML-GCN (Binary)	80.3	81.1	70.1	75.2	83.8	74.2	78.7	84.9	61.3	71.2	88.8	65.2	75.2
ML-GCN (Re-weighted)	**83.0**	**85.1**	**72.0**	**78.0**	**85.8**	**75.4**	**80.3**	**89.2**	64.1	**74.6**	**90.5**	**66.5**	**76.7**

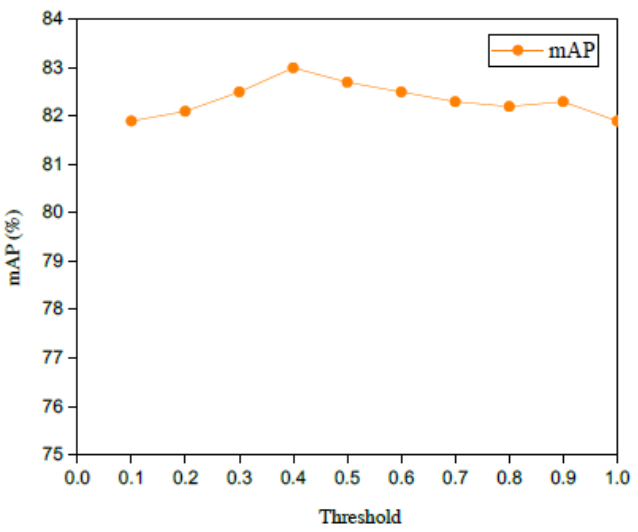
MS-COCO

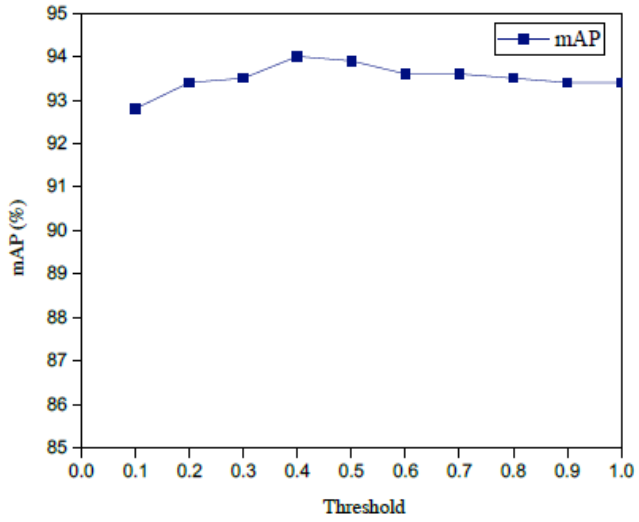
Methods	*aero*	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*motor*	*person*	*plant*	*sheep*	*sofa*	*train*	*tv*	mAP
CNN-RNN [28]	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	**99.7**	78.6	84.0
RLSD [34]	96.4	92.7	93.8	94.1	71.2	92.5	94.2	95.7	74.3	90.0	74.2	95.4	96.2	92.1	97.9	66.9	93.5	73.7	97.5	87.6	88.5
VeryDeep [26]	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	**87.8**	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
ResNet-101 [10]	99.5	97.7	97.8	96.4	65.7	91.8	96.1	97.6	74.2	80.9	85.0	**98.4**	96.5	95.9	98.4	70.1	88.3	80.2	98.9	89.2	89.9
FeV+LV [33]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	97.7	95.9	98.6	77.6	88.7	78.0	98.3	89.0	90.6
HCP [31]	98.6	97.1	98.0	95.6	75.3	**94.7**	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
RNN-Attention [29]	98.6	97.4	96.3	96.2	75.2	92.4	96.5	97.1	76.5	92.0	87.7	96.8	97.5	93.8	98.5	81.6	93.7	82.8	98.6	89.3	91.9
Atten-Reinforce [2]	98.6	97.1	97.1	95.5	75.6	92.8	96.8	97.3	78.3	92.2	87.6	96.9	96.5	93.6	98.5	81.6	93.1	83.2	98.5	89.3	92.0
VGG (Binary)	98.3	97.1	96.1	96.7	75.0	91.4	95.8	95.4	76.7	92.1	85.1	96.7	96.0	95.3	97.8	77.4	93.1	79.7	97.9	89.3	91.1
VGG (Re-weighted)	99.4	97.4	98.0	97.0	77.9	92.4	96.8	97.8	80.8	93.4	87.2	98.0	97.3	95.8	98.8	79.4	95.3	82.2	99.1	91.4	92.8
ML-GCN (Binary)	99.6	98.3	97.9	97.6	78.2	92.3	97.4	97.4	79.2	94.4	86.5	97.4	97.9	97.1	98.7	84.6	95.3	83.0	98.6	90.4	93.1
ML-GCN (Re-weighted)	99.5	**98.5**	**98.6**	**98.1**	80.8	94.6	**97.2**	98.2	**82.3**	**95.7**	86.4	98.2	**98.4**	**96.7**	**99.0**	**84.7**	**96.7**	**84.3**	98.9	**93.7**	**94.0**

VOC 2007

Experiments



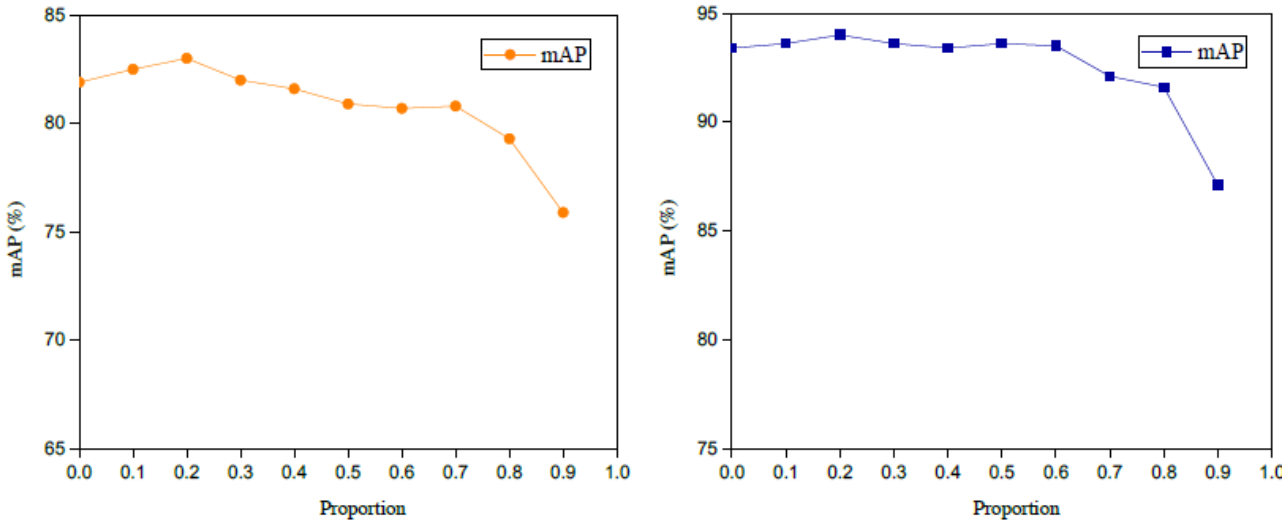




(a) Comparisons on MS-COCO. (b) Comparisons on VOC 2007.

Figure 5. Accuracy comparisons with different values of τ.

Experiments



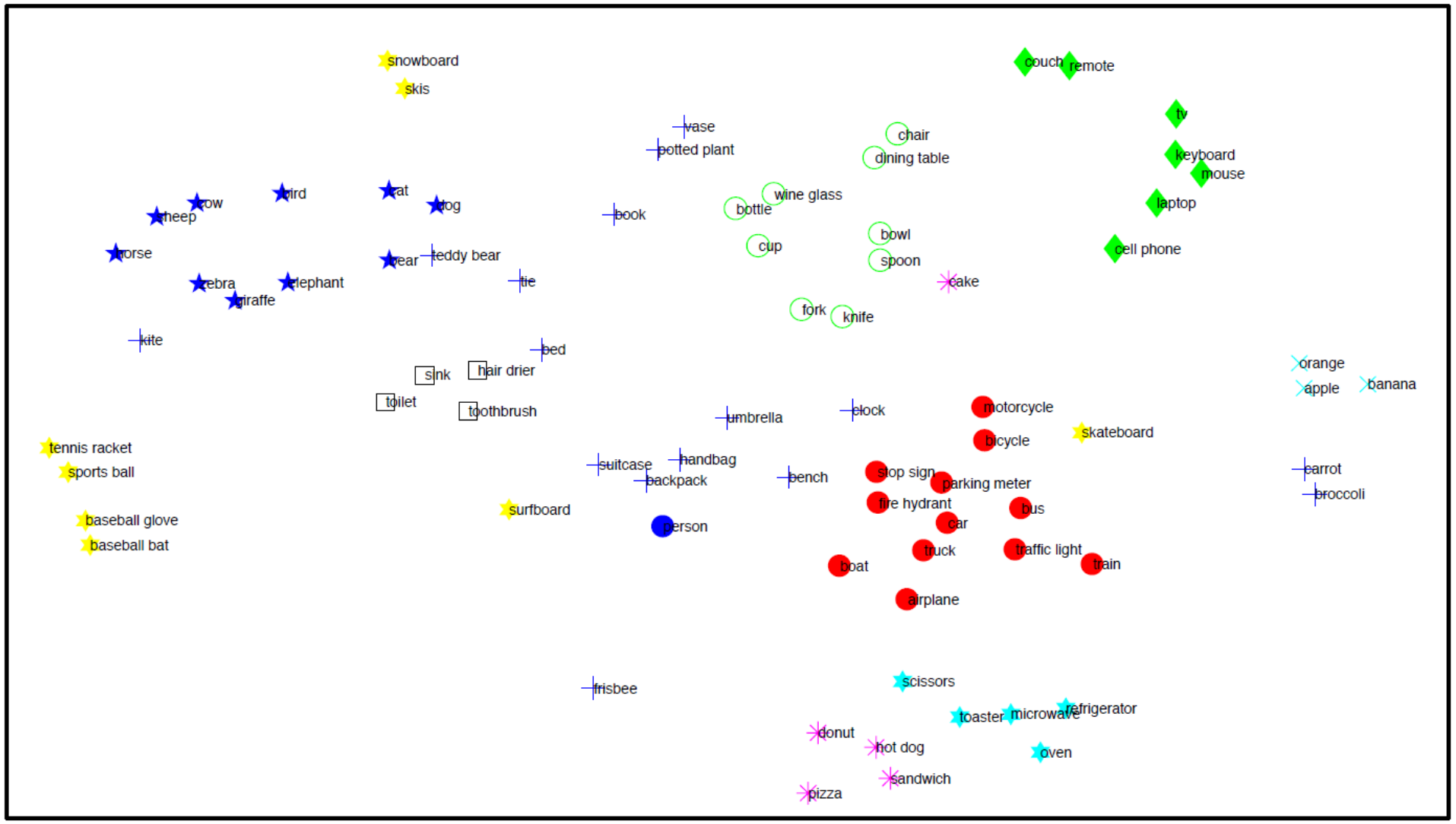
(a) Comparisons on MS-COCO. (b) Comparisons on VOC 2007.

Figure 6. Accuracy comparisons with different values of p. Note that, when $p = 1$, the model does not converge.

Table 3. Comparisons with different depths of GCN in our model.

♯ Layer	MS-COCO					VOC
	All			Top-3		All
	mAP	CF1	OF1	CF1	OF1	mAP
2-layer	**83.0**	**78.0**	**80.3**	**74.6**	**76.7**	**94.0**
3-layer	82.1	76.9	79.7	73.7	76.2	93.6
4-layer	81.1	76.4	79.4	72.5	75.8	93.0

Experiments



(a) t-SNE on the learned inter-dependent classifiers by our model.

Experiments

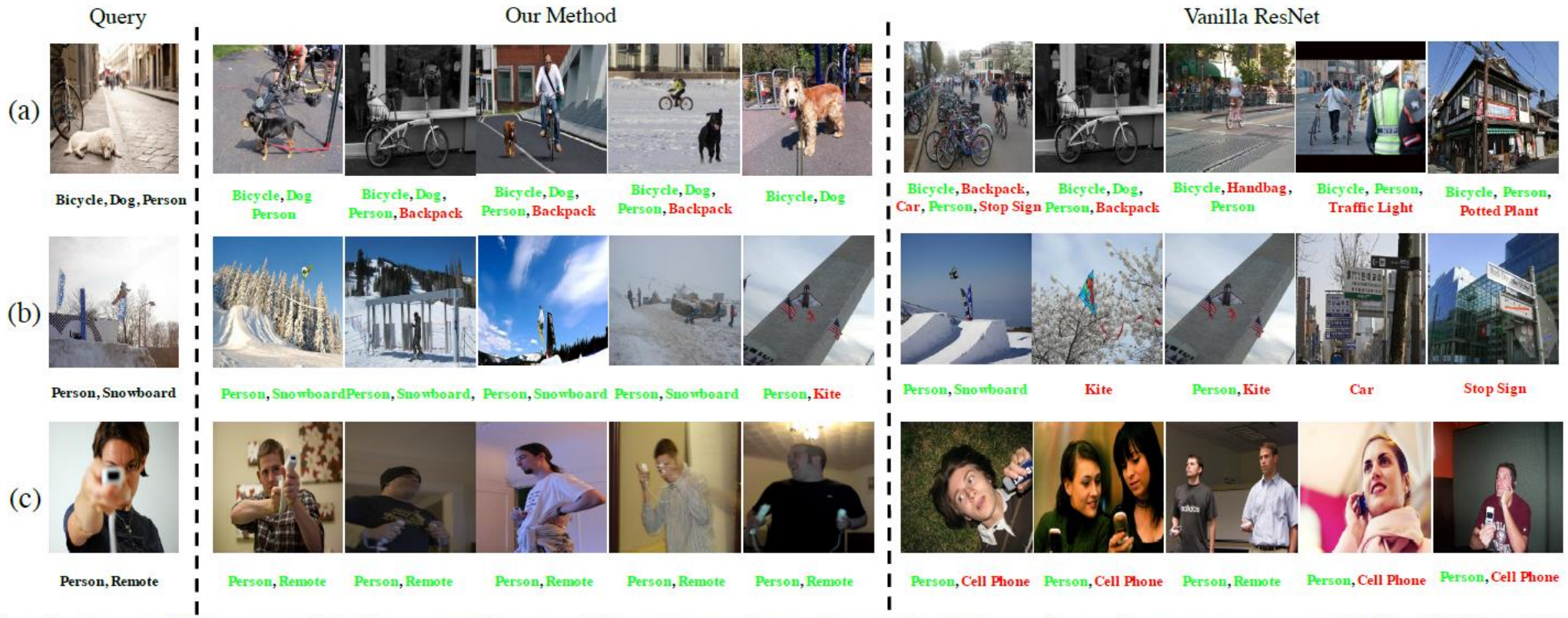


Figure 7. Top-5 returned images with the query image. The returned results on the left are based on our proposed ML-GCN, while the results on the right are vanilla ResNet. All results are sorted in the ascending order according to the distance from the query image.

Thanks