Learning a Deep ConvNet for Multi-label Classification with Partial Labels - Supplementary

Thibaut Durand Nazanin Mehrasa Greg Mori
Borealis AI Simon Fraser University
{tdurand,nmehrasa}@sfu.ca mori@cs.sfu.ca

A. Supplementary

A.1. Multi-label classification with GNN

In this section, we give additional information about the Graph Neural Networks (GNN) used in our work. We first show the algorithm used to predict the classification scores with a GNN in Algorithm 1. The input $\mathbf{x} \in \mathbb{R}^C$ of the GNN is the ConvNet output, where C is the number of categories.

The $f_{\mathcal{M}}$ function in the message update function $\mathcal{M}$ is a fully connected layer followed by a ReLU. Because the graph is fully-connected, the message update function $\mathcal{M}$ averages on all the nodes of the graph excepts the current node v *i.e.* $\Omega_v = \mathcal{V} \setminus \{v\}$. Similarly to [11], the final prediction uses both first and last hidden states. We observe that using both first and last hidden states is better than using only the last hidden state. According to [11], we use $T = 3$ iterations in our experiments.

A.2. Experimental details

Datasets. We perform experiments on large publicly available multi-label datasets: Pascal VOC 2007 [4], MS COCO [9] and NUS-WIDE [1]. Pascal VOC 2007 dataset contains 5k/5k trainval/test images of 20 objects categories. MS COCO dataset contains 123k images of 80 objects categories. We use the 2014 data split with 83k train images and 41k val images. NUS-WIDE dataset contains 269,648 images downloaded from Flickr that have been manually annotated with 81 visual concepts. We follow the experimental protocol in [5] and use 150k randomly sampled images for training and the rest for testing. The results on NUS-WIDE cannot be directly comparable with the other works because the number of total images is different (209,347 in [5], 200,261 in [8]). The main reason is that some provided URLs are invalid or some images have been deleted from Flickr. For our experiments, we collected 216,450 images.

We also performs experiments on the largest publicly available multi-label dataset: Open Images [7]. This dataset is partially annotated with human labels and machine generated labels. For our experiments, we use only human labels on the 600 boxable classes. On the training set, only 0.9% of the labels are available.

Algorithm 1 Graph Neural Network (GNN)

Input: ConvNet output $\mathbf{x}$

1: Initialize the hidden state of each node $v \in \mathcal{V}$ with the output of the ConvNet.

$$\mathbf{h}_v^0 = [0, \dots, 0, x_v, 0, \dots, 0] \qquad \forall v \in \mathcal{V} \tag{1}$$

2: **for** t = 0 **to** T-1 **do**

3: Update message of each node $v \in \mathcal{V}$ based on the hidden states

$$\mathbf{m}_v^t = \mathcal{M}(\{\mathbf{h}_u^t | u \in \Omega_v\}) = \frac{1}{|\Omega_v|} \sum_{u \in \Omega_v} f_{\mathcal{M}}(\mathbf{h}_u^t) \tag{2}$$

4: Update hidden state of each node $v \in \mathcal{V}$ based on the messages

$$\mathbf{h}_v^{t+1} = \mathcal{F}(\mathbf{h}_v^t, \mathbf{m}_v^t) = GRU(\mathbf{h}_v^t, \mathbf{m}_v^t) \tag{3}$$

5: **end for**

6: Compute the output based on the first and last hidden states

$$\bar{\mathbf{y}} = s(\mathbf{h}_v^0, \mathbf{h}_v^T) = \mathbf{h}_v^0 + \mathbf{h}_v^T \tag{4}$$

Output: $\bar{\mathbf{y}}$

Implementation details. The hyperparameters of the WELDON pooling function [2, 3] are $k^+ = k^- = 0.1$. The models are implemented with PyTorch [10] and are trained with SGD during 20 epochs with a batch size of 16. The initial learning rate is 0.01 and it is divide by 10 after 10 epochs. During training, we only use random horizontal flip as data augmentation. Each image is resized to 448×448 with 3 color channels. On Open Images dataset, unlike [7] we do not train from scratch the network. We use a similar protocol that on the others datasets: we fine-tune a model

pre-train on ImageNet but stop the training when the validation performance does not increase. Because the training set has 1.7M images, the model converge in less than 5 epochs.

A.3. Multi-label metrics

In this section, we introduce the metrics used to evaluate the performances on multi-label datasets. We note $\mathbf{y}^{(i)} = [y_1^{(i)}, \dots, y_C^{(i)}] \in \mathcal{Y} \subseteq \{-1, 0, 1\}^C$ the ground truth label vector and $\hat{\mathbf{y}}^{(i)} = [\hat{y}_1^{(i)}, \dots, \hat{y}_C^{(i)}] \in \{-1, 1\}^C$ the predicted label vector of the i-th example.

Zero-one exact match accuracy (0-1). This metric considers a prediction correct only if all the labels are correctly predicted:

$$m_{0/1}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\mathbf{y}^{(i)} = \hat{\mathbf{y}}^{(i)}] \tag{5}$$

where $\mathbb{1}[.]$ is an indicator function.

Per-class precision/recall (PC-P/R).

$$m_{PC-P}(\mathcal{D}) = \frac{1}{C} \sum_{c=1}^{C} \frac{N_c^{correct}}{N_c^{predict}} \tag{6}$$

$$m_{PC-R}(\mathcal{D}) = \frac{1}{C} \sum_{c=1}^{C} \frac{N_c^{correct}}{N_c^{gt}} \tag{7}$$

where $N_c^{correct}$ is the number of correctly predicted images for the c-th label, $N_c^{predict}$ is the number of predicted images, N_c^{gt} is the number of ground-truth images. Note that the per-class measures treat all classes equal regardless of their sample size, so one can obtain a high performance by focusing on getting rare classes right.

Overall precision/recall (OV-P/R). Unlike per-class metrics, the overall metrics treat all samples equal regardless of their classes.

$$m_{OV-P}(\mathcal{D}) = \frac{\sum_{c=1}^{C} N_c^{correct}}{\sum_{c=1}^{C} N_c^{predict}} \tag{8}$$

$$m_{OV-R}(\mathcal{D}) = \frac{\sum_{c=1}^{C} N_c^{correct}}{\sum_{c=1}^{C} N_c^{gt}} \tag{9}$$

Macro-F1 (M-F1). The macro-F1 score [15] is the F1 score [12] averaged across all categories.

$$m_{MF1}(\mathcal{D}) = \frac{1}{C} \sum_{c=1}^{C} F_1^c \tag{10}$$

Given a category c, the F1 measure, defined as the harmonic mean of precision and recall, is computed as follows:

$$F_1^c = \frac{2P^cR^c}{P^c + R^c} \tag{11}$$

where the precision (P^c) and the recall (R^c) are calculated as follows:

$$P^c = \frac{\sum_{i=1}^{N} \mathbb{1}[y_c^{(i)} = \hat{y}_c^{(i)}]}{\sum_{i=1}^{N} \hat{y}_c^{(i)}} \tag{12}$$

$$R^c = \frac{\sum_{i=1}^{N} \mathbb{1}[y_c^{(i)} = \hat{y}_c^{(i)}]}{\sum_{i=1}^{N} y_c^{(i)}} \tag{13}$$

and $y_c^{(i)} \in \{0, 1\}$

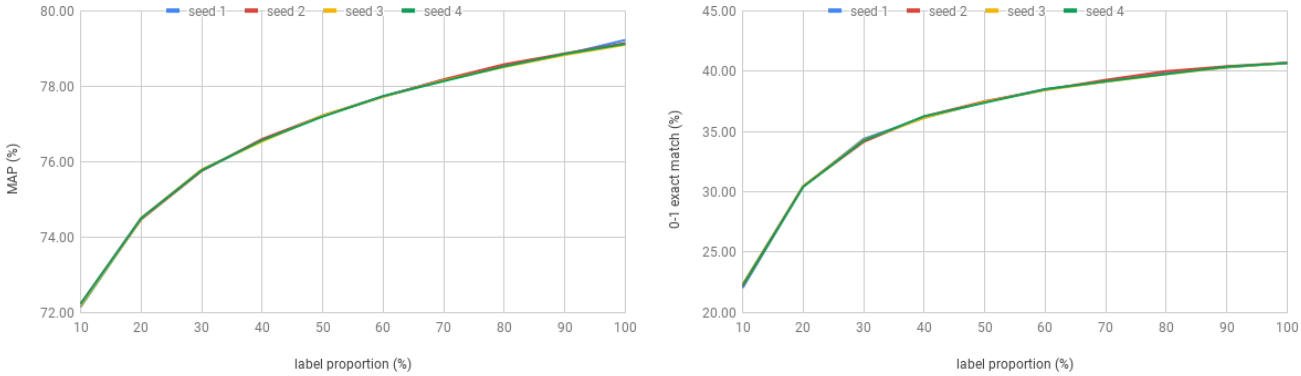
Micro-F1 (m-F1). The micro-F1 score [13] is computed using the equation of F_1^c and considering the predictions as a whole

$$m_{mF1}(\mathcal{D}) = \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} \mathbb{1}[y_c^{(i)} = \hat{y}_c^{(i)}]}{\sum_{c=1}^{C}\sum_{i=1}^{N} y_c^{(i)} + \sum_{c=1}^{C}\sum_{i=1}^{N} \hat{y}_c^{(i)}} \tag{14}$$

According to the definition, macro-F1 is more sensitive to the performance of rare categories while micro-F1 is affected more by the major categories.

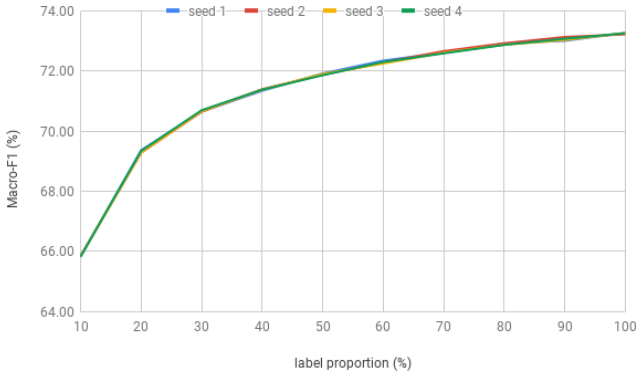
A.4. Analysis of the initial set of labels

In this section, we analyse the initial set of labels for the partial label scenario. We report the results for 4 random seeds to generate the initial set of partial labels. The experiments are performed on MS COCO val2014 with a ResNet-101 WELDON. The results are shown in Table 1 and Figure 1 for different label proportions and metrics. For every label proportion and every metric, we observe that the model is robust to the initial set of labels.

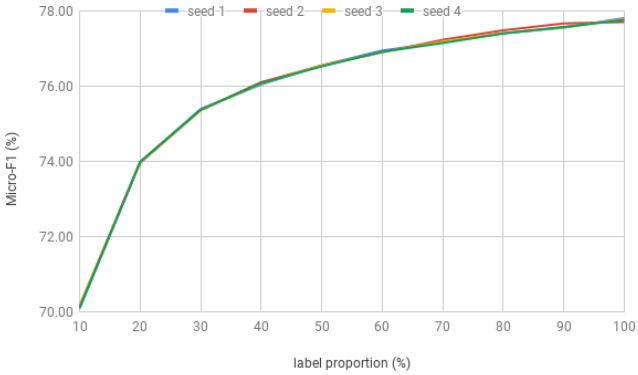


MAP

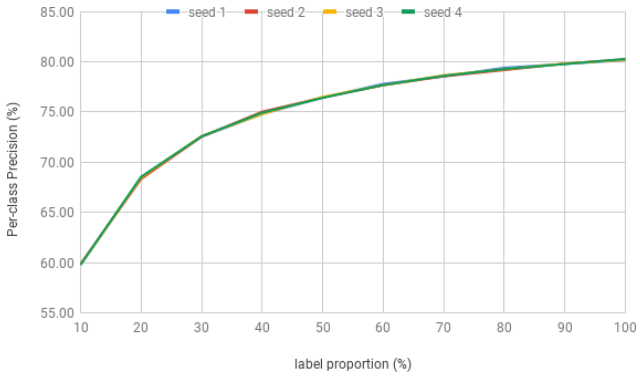
0-1 exact match



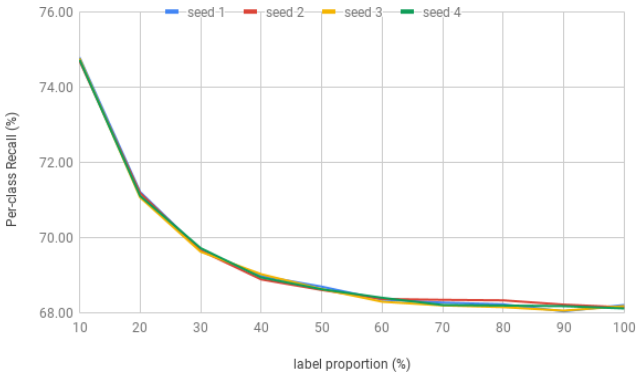
Macro-F1



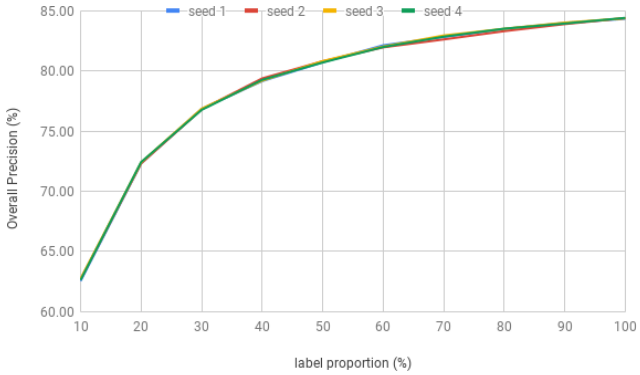
Micro-F1



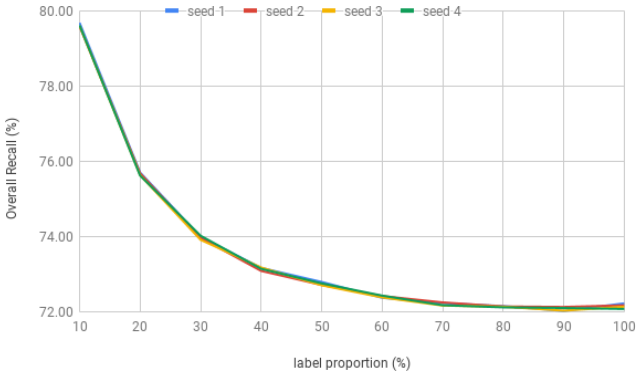
Per-class Precision



Per-class Recall



Overall Precision



Overall Recall

Figure 1. Results for differents metrics on MS COCO val2014 to analyze the sensibility of the initial label set.

metric	label proportion									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
MAP	72.20±0.04	74.49±0.02	75.77±0.02	76.57±0.03	77.21±0.01	77.73±0.01	78.16±0.02	78.53±0.03	78.85±0.02	79.14±0.05
M-F1	65.84±0.01	69.32±0.04	70.66±0.02	71.37±0.02	71.88±0.03	72.29±0.04	72.61±0.03	72.89±0.03	73.05±0.06	73.24±0.02
m-F1	70.13±0.04	73.97±0.01	75.36±0.01	76.07±0.03	76.54±0.01	76.91±0.02	77.17±0.04	77.42±0.04	77.58±0.05	77.75±0.04
0-1	22.21±0.12	30.44±0.03	34.26±0.11	36.18±0.07	37.44±0.05	38.46±0.04	39.16±0.07	39.83±0.12	40.34±0.04	40.67±0.02
PC-P	59.82±0.05	68.45±0.10	72.56±0.03	74.88±0.11	76.45±0.04	77.70±0.07	78.59±0.05	79.28±0.10	79.80±0.02	80.22±0.05
PC-R	74.74±0.04	71.14±0.07	69.66±0.04	68.96±0.06	68.64±0.04	68.35±0.04	68.26±0.07	68.23±0.08	68.12±0.09	68.16±0.04
OV-P	62.66±0.09	72.36±0.06	76.81±0.04	79.24±0.10	80.75±0.06	82.01±0.08	82.79±0.14	83.44±0.10	83.94±0.06	84.36±0.04
OV-R	79.62±0.04	75.66±0.04	73.97±0.05	73.14±0.04	72.74±0.04	72.40±0.03	72.21±0.04	72.14±0.01	72.07±0.05	72.15±0.06

Table 1. Analysis of the initial set of labels for the partial label scenario. The results are averaged for 4 seeds on MS COCO val2014.

A.5. Analysis of the labeling strategies

In this section we analysis the labeling strategies for different network architectures. The results are shown in Table 2 and Figure 2 on MS COCO dataset. Overall, the results are very similar. For a given proportion of labels, we observe that the partial labels strategy is better that the complete image labels. The improvement increases when the label proportion decreases. The performance of a model learned with noisy labels drops significantly, even for large proportion of clean labels.

In Figure 3, we also show the results for different metrics. For MAP, Macro-F1 and Micro-F1, we observe a similar behaviour: the partial labels strategy has better performances than the complete image labels strategy. For the 0-1 exact match metric, we observe that the complete image labels strategy has better performances than the complete image labels strategy. For this metric, the predictions of all the categories must be corrected, so it advantages the complete image labels strategy because some training images have all the labels whereas in the partial labels strategy, none of the training images have all labels. For the precision and recall metrics, the behaviours are different for the complete image labels strategy and the partial labels strategy. We note that the complete image labels strategy has a better per-class/overall precision than the partial labels strategy but is has a lower per-class/overall recall than the partial labels strategy.

Comparison to noisy+ strategy. In Table 3, we show results for the noisy+ strategy on Pascal VOC 2007, MS COCO and NUS-WIDE for different metrics. For every dataset, we observe that the noisy+ strategy drops the performances of all the metrics with respect to the model learned with only 10% of clean labels.

architecture	labels	label proportion									
		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
ResNet-50	partial	61.26	63.78	65.21	66.22	66.97	67.60	68.16	68.58	69.01	69.33
	dense	54.29	59.67	62.50	64.28	65.60	66.68	67.55	68.26	68.80	69.32
	noisy	-	-	-	-	3.75	39.77	56.82	62.93	66.24	69.33
ResNet-50 WELDON	partial	69.91	72.37	73.74	74.53	75.25	75.77	76.25	76.66	77.02	77.28
	dense	62.16	68.04	71.14	73.01	74.17	75.14	75.83	76.42	76.88	77.28
	noisy	-	-	-	-	3.73	52.99	67.08	72.03	74.69	77.29
ResNet-101 WELDON	partial	72.15	74.49	75.76	76.56	77.22	77.73	78.17	78.53	78.84	79.22
	dense	65.22	71.00	73.80	75.44	76.59	77.44	78.08	78.61	78.90	79.24
	noisy	-	-	-	-	3.63	53.10	69.09	74.06	76.85	79.18
ResNeXt-101 WELDON	partial	75.74	77.80	78.95	79.64	80.22	80.61	80.94	81.24	81.48	81.69
	dense	69.03	74.58	77.13	78.50	79.38	80.15	80.65	81.05	81.40	81.71
	noisy	-	-	-	-	3.63	49.26	70.16	75.22	78.28	81.66

Table 2. Comparison of the labeling strategies for different label proportions and different architectures on MS COCO val2014.

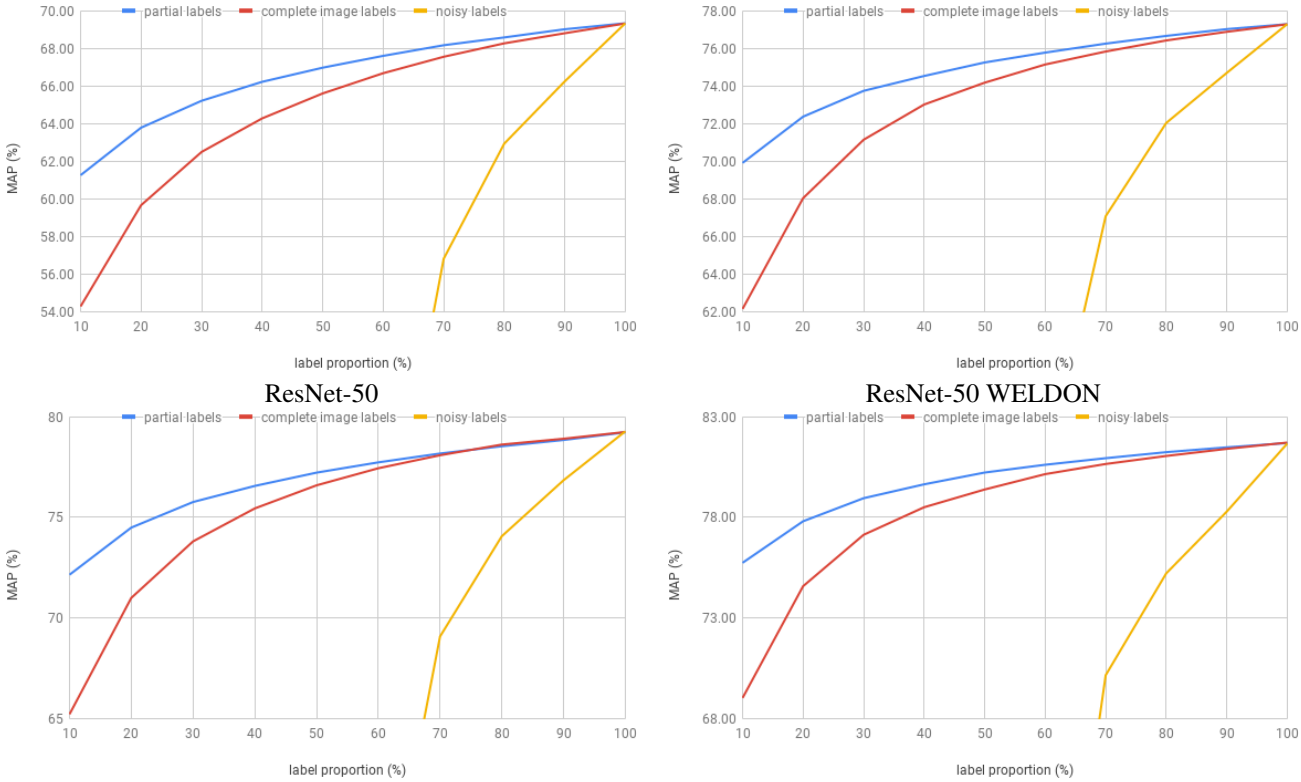


Figure 2. Comparison of the labeling strategies for different label proportions and different architectures on MS COCO val2014.

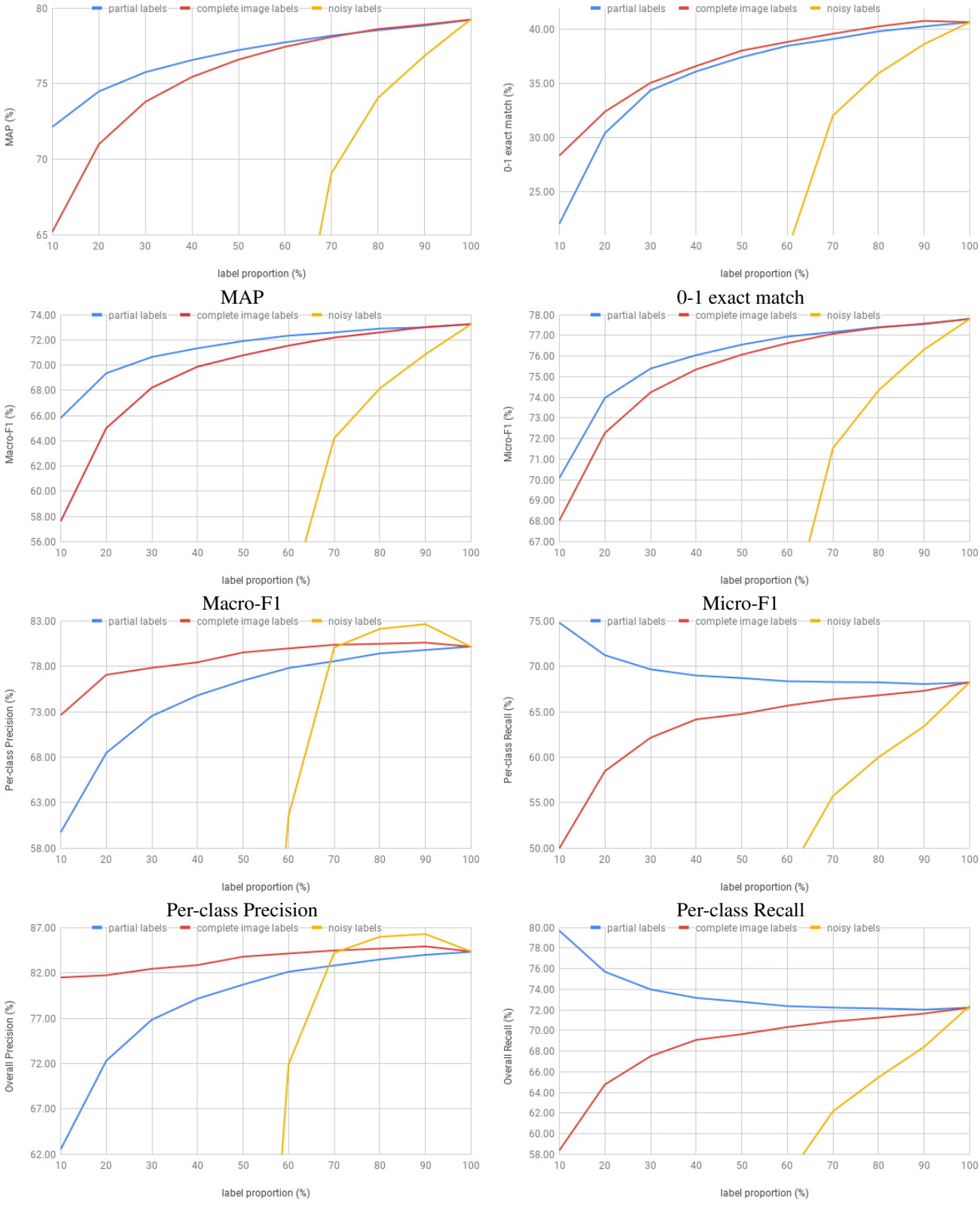


Figure 3. Comparison of the labeling strategies for different metrics on MS COCO val2014.

dataset	strategy	clean label	noisy label	MAP	0-1	M-F1	m-F1	PC-P	PC-R	OV-P	OV-R
VOC 2007	clean	100	0	93.93	79.16	88.90	91.12	90.72	87.34	93.40	88.95
	noisy+	97.1	2.9	90.94	62.21	78.11	78.62	95.41	68.64	97.20	66.00
	partial 10%	10	0	89.09	47.46	74.55	77.84	63.35	94.16	66.02	94.81
MS COCO	clean	100	0	79.22	40.69	73.26	77.80	80.16	68.21	84.31	72.23
	noisy+	97.6	2.4	71.60	20.28	38.62	33.72	91.76	28.17	97.34	20.39
	partial 10%	10	0	72.15	22.04	65.82	70.09	59.76	74.78	62.56	79.68
NUS-WIDE	clean	100	0	54.88	42.29	51.88	71.15	58.54	49.33	73.83	68.66
	noisy+	98.6	1.4	47.44	36.07	18.83	28.53	59.71	13.95	83.72	17.19
	partial 10%	10	0	51.14	25.98	51.36	65.52	41.80	69.23	53.62	84.19

Table 3. Comparison with a webly-supervised strategy (noisy+) on MS COCO. Clean (resp. noisy) means the percentage of clean (resp. noisy) labels in the training set. Noisy+ is a labeling strategy where there is only one positive label per image.

A.6. Comparison of the loss functions

In this section, we analyse the performances of the BCE and partial-BCE loss functions for different metrics. The results on MS COCO (resp. Pascal VOC 2007) are shown in Figure 5 (resp. Figure 7) and the improvement of the partial-BCE with respect to the BCE is shown in Figure 6 (resp. Figure 8). We observe that the partial-BCE significantly improves the performances for MAP, 0-1 exact match, Macro-F1 and Micro-F1 metrics. We note that the improvement is bigger when the label proportion is lower. The proposed loss also improves the (overall and per-class) recall for both datasets. On Pascal VOC 2007, it also improves the overall and per-class precision. However, we observe that the

We observe that decreasing the proportion of known labels can slightly improves the performances with respect to the model trained with all the annotations. This phenomenon is because of the tuning of the learning rate and the hyperparameter γ (Figure 6). Note that the BCE and the partial-BCE have the same results for the label proportion 100% because they are equivalent by definition. In the paper, we used the same training setting (learning rate, weight decay, *etc*.) as [3] for each model and dataset. In Figure 4, we observe that using a learning rate of 0.02 increases the performance and leads to a monotone increase of the performance with respect to the label proportion, but the optimal learning rate depends on the dataset. It is possible to improve the results by tuning carefully these hyperparameters, but we observe that the partial-BCE is still better than the BCE for a large range of LRs and for small label proportions which is the main focus of the paper.

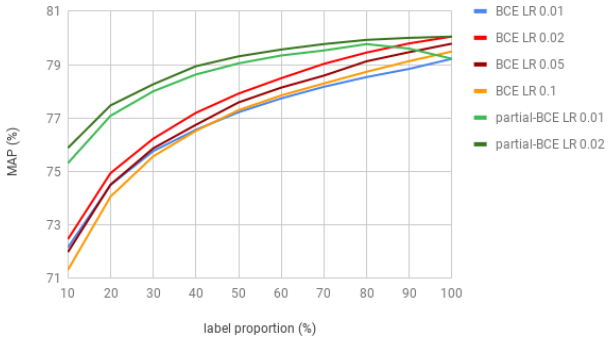
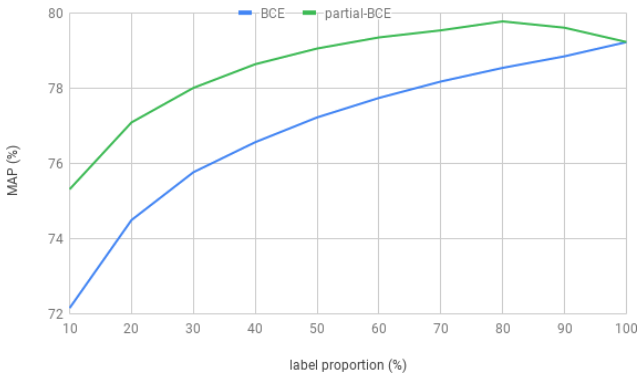
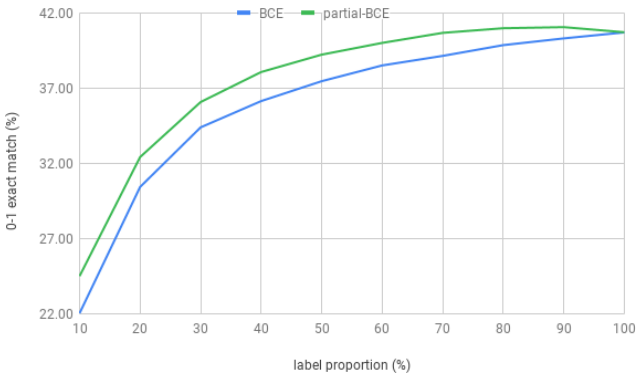


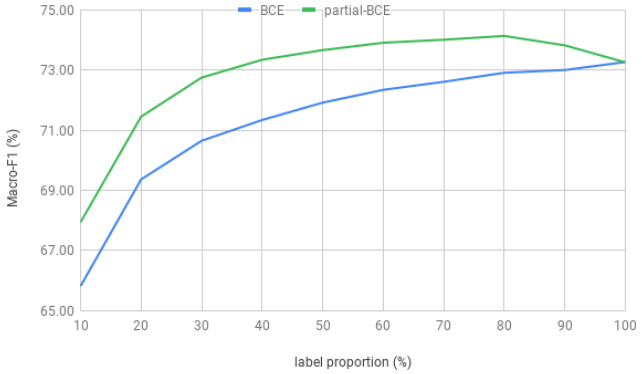
Figure 4. Analysis of the learning rate on MS COCO dataset.



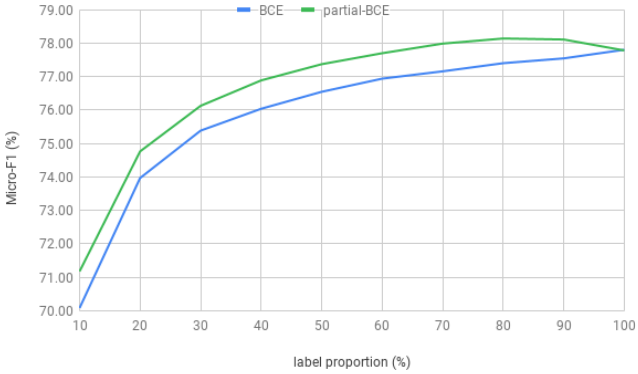
MAP



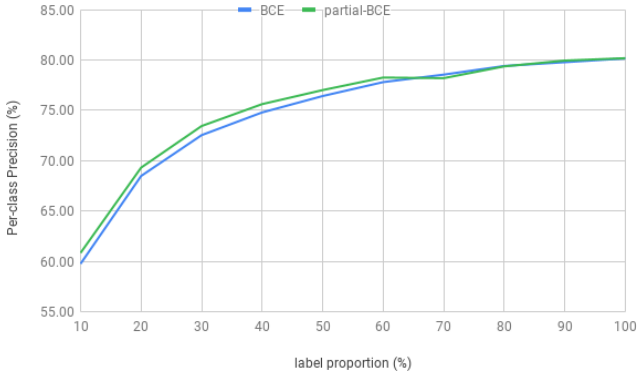
0-1 exact match



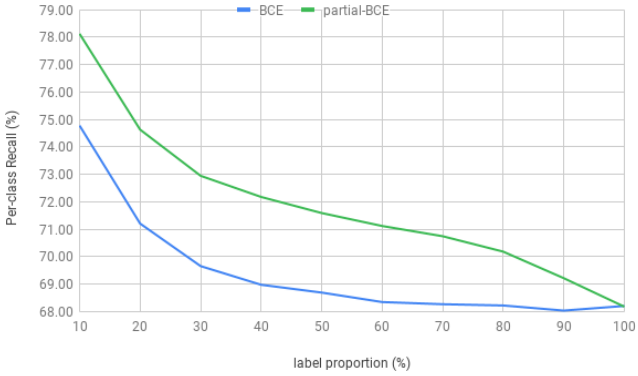
Macro-F1



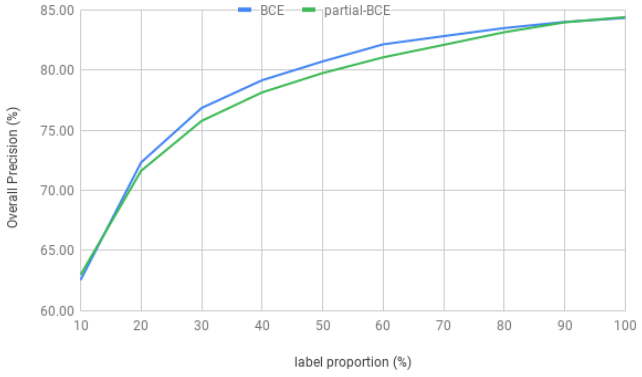
Micro-F1



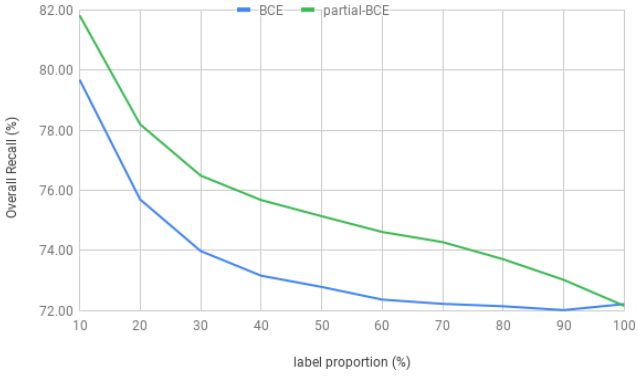
Per-class Precision



Per-class Recall

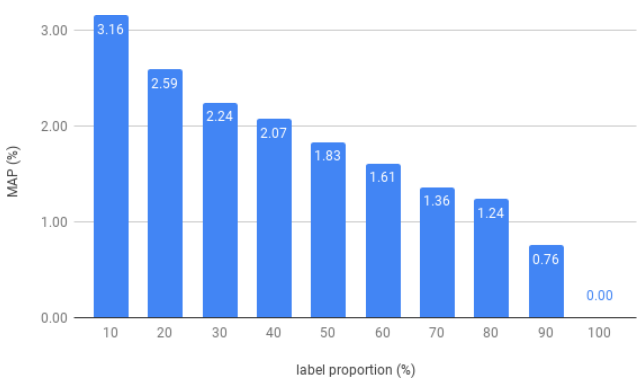


Overall Precision

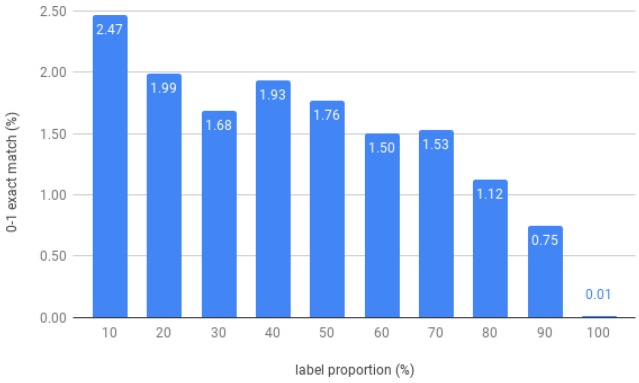


Overall Recall

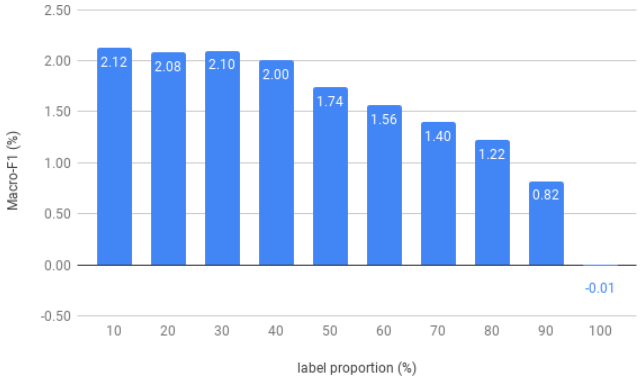
Figure 5. Results for different metrics on MS COCO val2014.



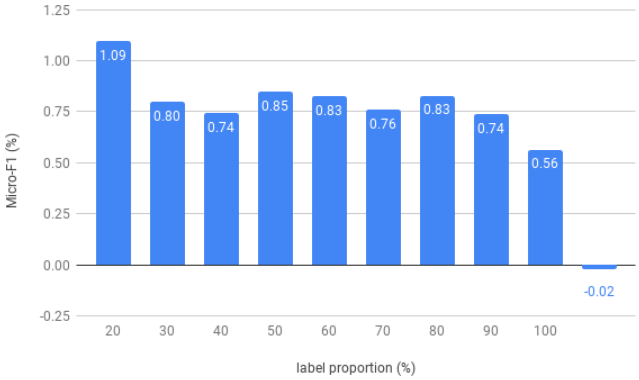
MAP



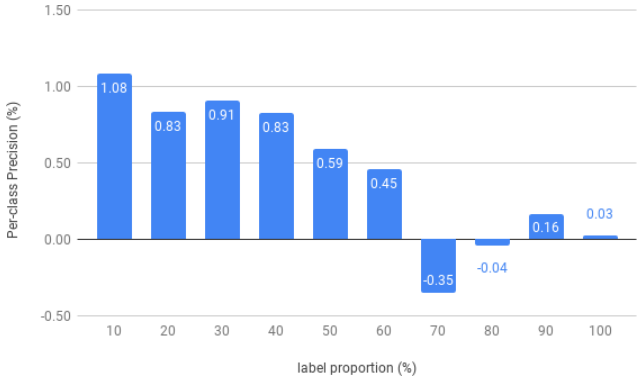
0-1 exact match



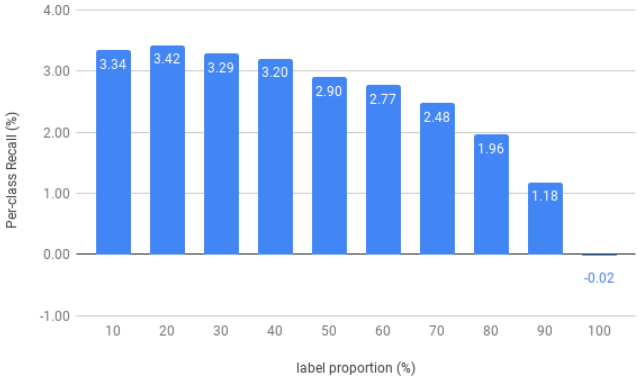
Macro-F1



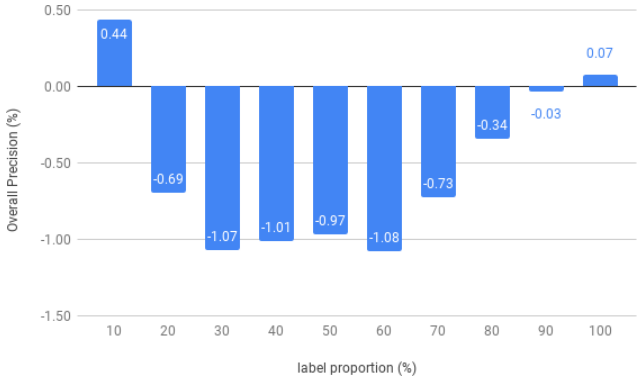
Micro-F1



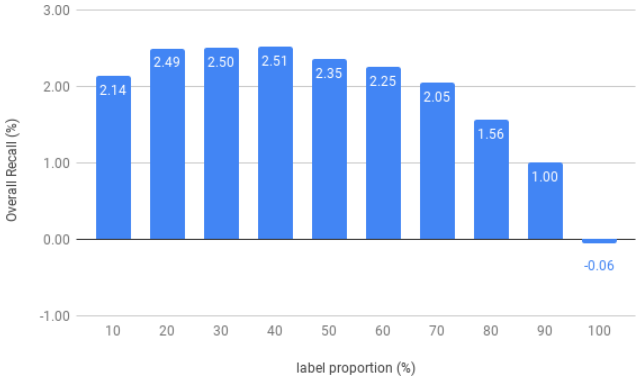
Per-class Precision



Per-class Recall

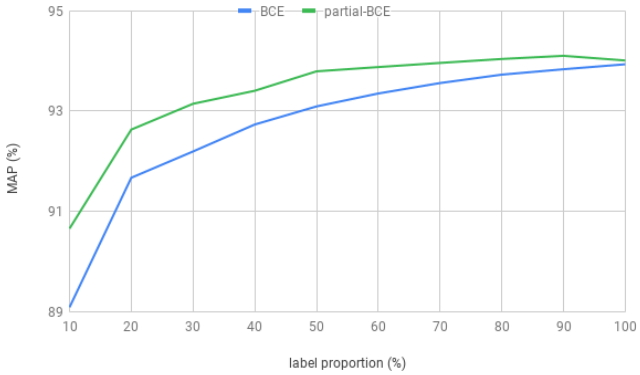


Overall Precision

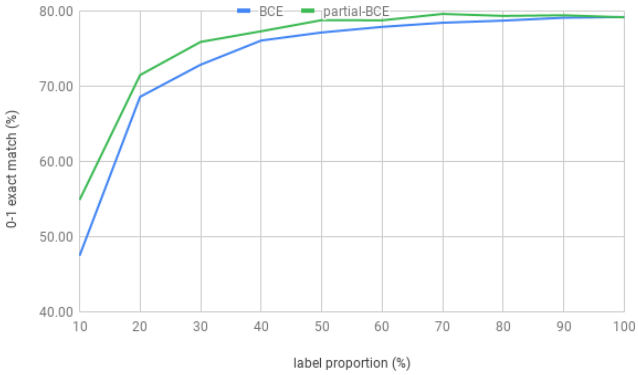


Overall Recall

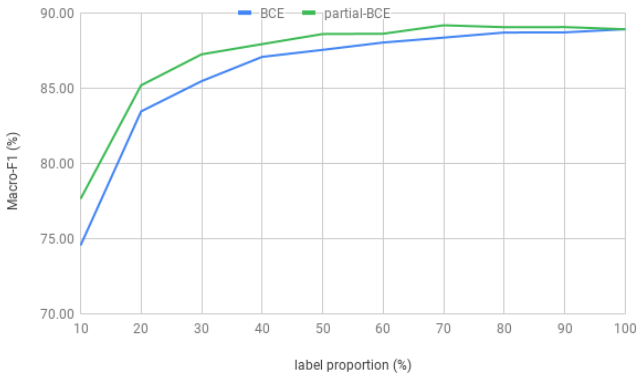
Figure 6. Improvement analysis between partial-BCE and BCE for differents metrics on MS COCO val2014.



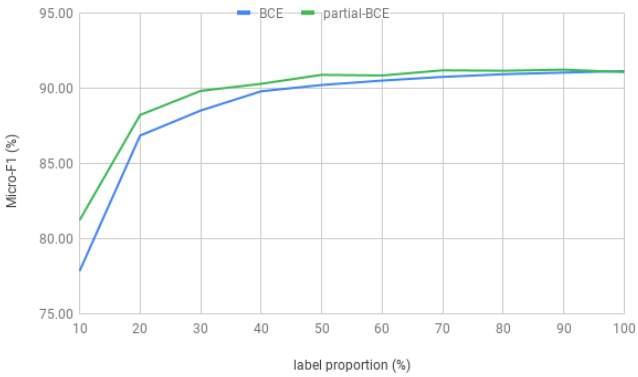
MAP



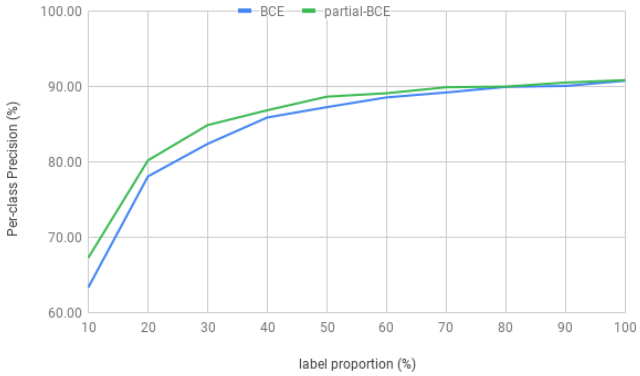
0-1 exact match



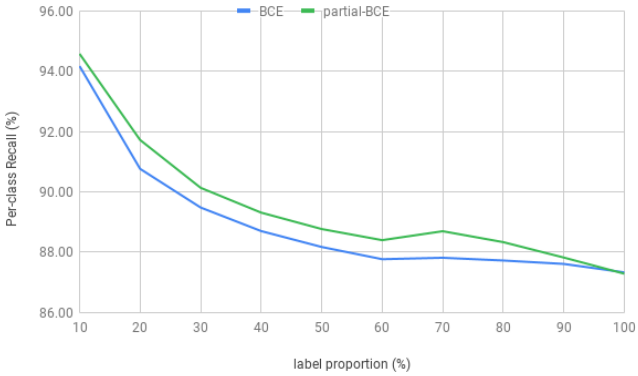
Macro-F1



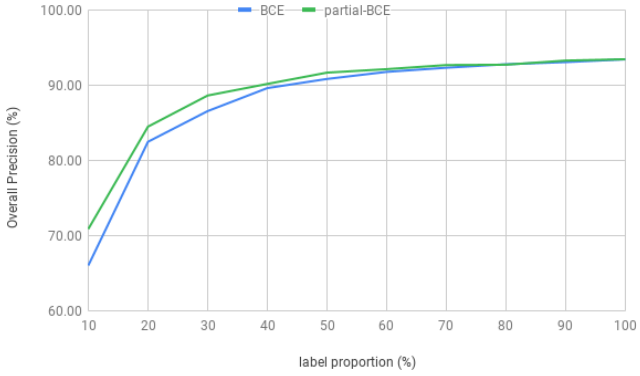
Micro-F1



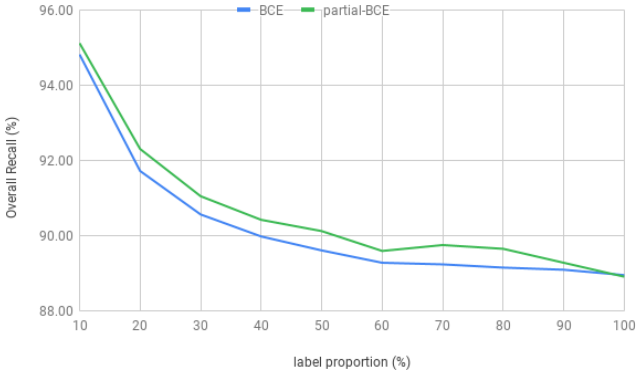
Per-class Precision



Per-class Recall

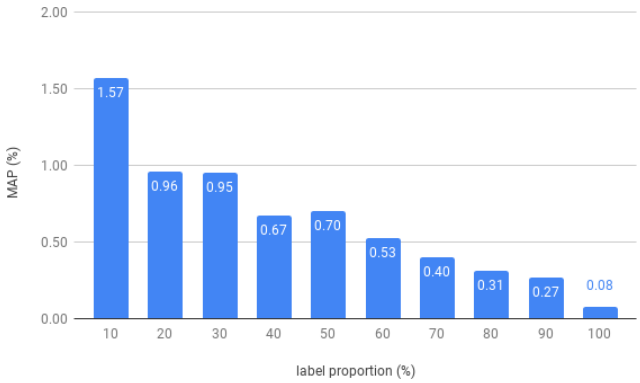


Overall Precision

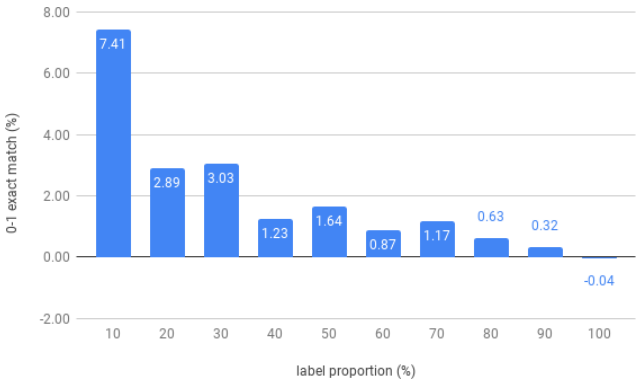


Overall Recall

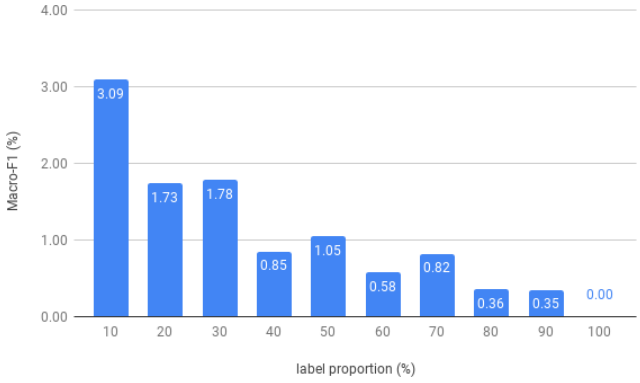
Figure 7. Results for different metrics on Pascal VOC 2007.



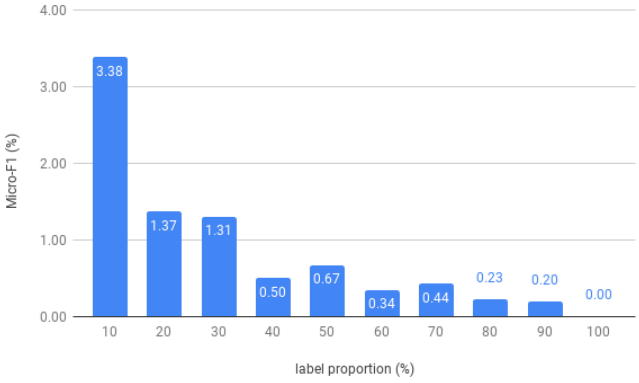
MAP



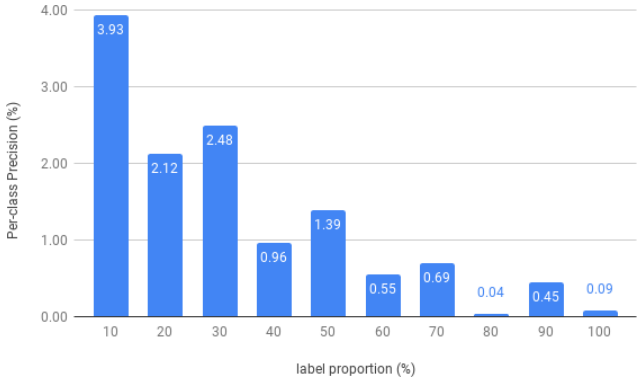
0-1 exact match



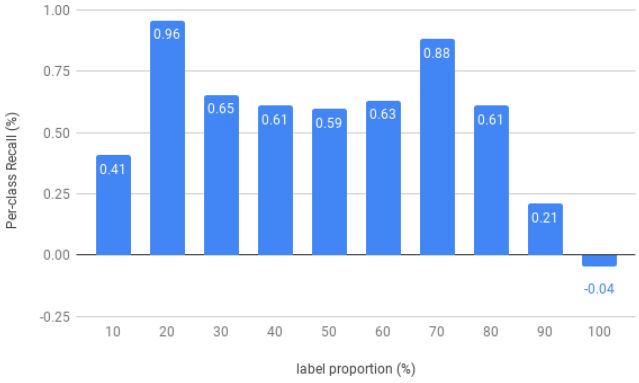
Macro-F1



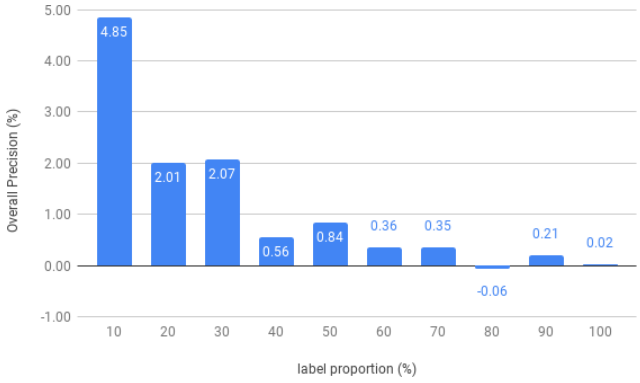
Micro-F1



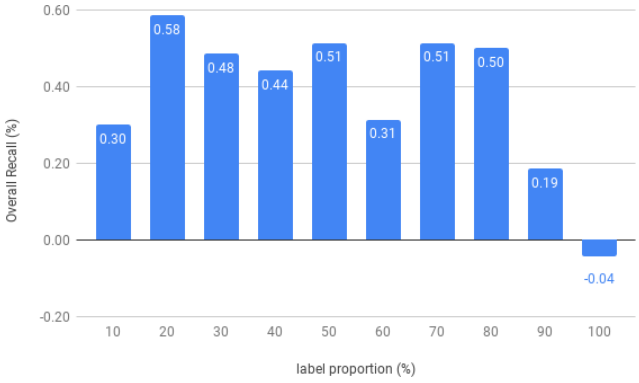
Per-class Precision



Per-class Recall



Overall Precision

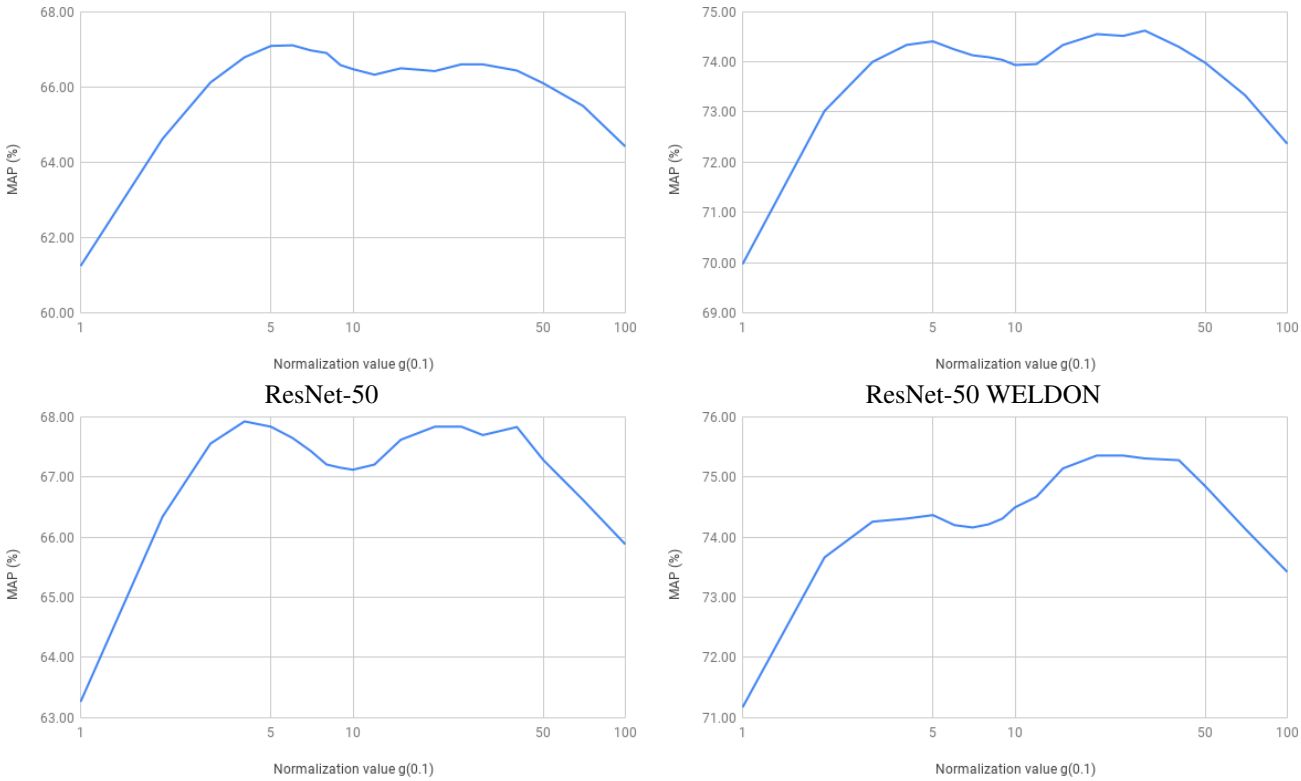


Overall Recall

Figure 8. Improvement analysis between partial-BCE and BCE for differents metrics on Pascal VOC 2007.

A.7. Analysis of the loss function

In this section, we analyze the hyperparameter of the loss function for several network architectures. The models are trained on the train2014 set minus 5000 images that are used as validation set to evaluate the performances. The Figure 9 shows the results on MS COCO. We observe a similar behavior for all the architectures. Overall, using a normalization value $g(0.1)$ between 3 and 50 significantly improves the performances with respect to the normalization by the number of categories ($g(0.1) = 1$). The loss is robust to the value of this hyperparmeter.



ResNet-50 ResNet-50 WELDON

ResNet-101 ResNet-101 WELDON

Figure 9. Analysis of the normalization value for 10% of known labels (*i.e.* $g(0.1)$) on MS COCO. (x-axis log-scale)

A.8. Comparison to existing model for missing labels

As pointed out in the related work section, most of the existing models to learn with missing labels are not scalable and do not allow experiments on large-scale dataset like MS COCO and NUS-WIDE. We compare our model with the APG-Graph model [14] that models structured semantic correlations between images on the Pascal VOC 2007 dataset. Unlike our method, the APG-Graph model does not allow to fine-tune the ConvNet.

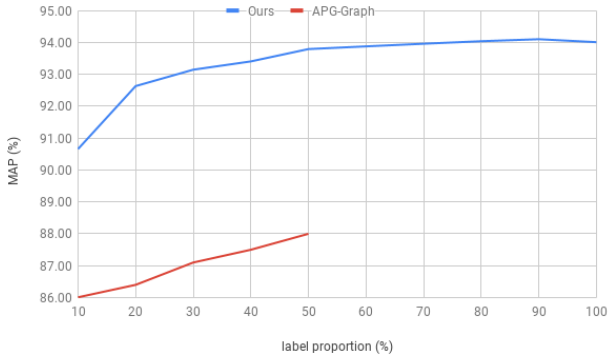


Figure 10. Comparison with APG-Graph model on Pascal VOC 2007 for different proportion of known labels.

A.9. What is the best strategy to predict missing labels?

This section extends the section 4.3 in the paper. First, to compute the Bayesian uncertainty, we use the setting used in the original paper [6]. The results for different strategies and hyperparameters are shown in Table 4. G defines how the examples are selected during training. In the paper, we only explain how to find the solution with respect to $\mathbf{v}$. G depends on the strategy and is defined as:

$$G(\mathbf{v};\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{C} v_{ic} \log\left(\frac{1}{1+e^{-\theta}}\right)$$

for strategy [a].

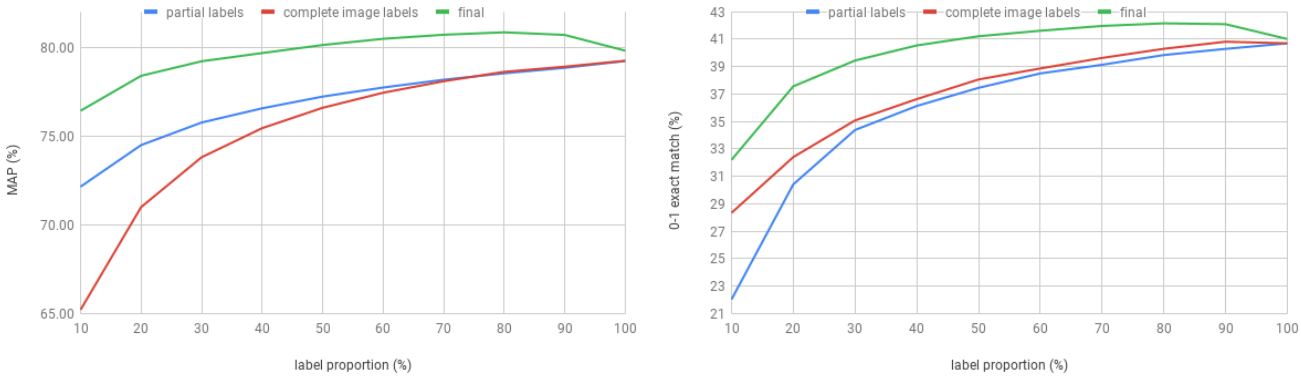
For strategy [a] and [d], we observe that using a small threshold is better than a large threshold. On the contrary, for strategy [c] we observe that using a large threshold is better than a small threshold, but the results are worse than strategy [a]. For strategy [b], labeling a large proportion of labels per mini-batch is better than labeling a small proportion of labels. For strategy [e], we note that using a GNN improves the performances of the model and the model is more robust to the threshold hyperparameter θ.

Relabeling	MAP	0-1	Macro-F1	Micro-F1	label prop.	TP	TN	GNN
2 steps (no curriculum)	-1.49	6.42	2.32	1.99	100	82.78	96.40	✓
[a] Score threshold $\theta = 1$	0.00	11.31	3.71	4.25	97.87	82.47	97.84	✓
[a] Score threshold $\theta = 2$	0.34	11.15	4.33	4.26	95.29	85.00	98.50	✓
[a] Score threshold $\theta = 5$	0.31	5.02	2.60	1.83	70.98	96.56	99.44	✓
[b] Score proportion $\theta = 0.1$	0.45	-1.20	-0.28	-0.68	26.70	99.28	99.19	✓
[b] Score proportion $\theta = 0.2$	0.36	0.20	0.70	0.10	42.09	98.35	99.33	✓
[b] Score proportion $\theta = 0.3$	0.28	0.91	1.09	0.37	55.63	97.82	99.38	✓
[b] Score proportion $\theta = 0.4$	0.55	2.95	2.33	1.28	67.41	96.87	99.38	✓
[b] Score proportion $\theta = 0.5$	0.22	4.02	2.76	1.74	77.40	95.52	99.30	✓
[b] Score proportion $\theta = 0.6$	0.41	6.17	3.63	2.52	85.37	93.16	99.15	✓
[b] Score proportion $\theta = 0.7$	0.35	7.49	3.83	3.07	91.69	89.40	98.81	✓
[b] Score proportion $\theta = 0.8$	0.17	8.40	3.70	3.25	96.24	84.40	98.10	✓
[c] Postitive only - score $\theta = 1$	-1.61	-31.75	-18.07	-18.92	16.79	36.42	-	✓
[c] Postitive only - score $\theta = 2$	-0.80	-21.31	-10.93	-12.08	14.71	47.94	-	✓
[c] Postitive only - score $\theta = 5$	0.31	-4.58	-1.92	-2.23	12.01	79.07	-	✓
[d] Ensemble score $\theta = 1$	-0.31	10.16	3.61	3.94	97.84	82.12	97.76	✓
[d] Ensemble score $\theta = 2$	0.23	11.31	4.16	4.33	95.33	84.80	98.53	✓
[d] Ensemble score $\theta = 5$	0.27	3.78	2.38	1.53	70.77	96.56	99.44	✓
[e] Bayesian uncertainty $\theta = 0.1$	0.26	1.84	1.36	0.64	22.63	25.71	99.98	
[e] Bayesian uncertainty $\theta = 0.2$	0.29	8.49	4.05	3.66	60.32	48.39	99.82	
[e] Bayesian uncertainty $\theta = 0.3$	0.34	10.15	4.37	3.72	77.91	61.15	99.24	
[e] Bayesian uncertainty $\theta = 0.4$	0.30	9.05	4.17	3.37	87.80	68.56	98.70	
[e] Bayesian uncertainty $\theta = 0.5$	0.26	8.32	3.83	3.05	92.90	70.96	98.04	
[e] Bayesian uncertainty $\theta = 0.1$	0.36	2.71	1.91	1.22	19.45	38.15	99.97	✓
[e] Bayesian uncertainty $\theta = 0.2$	0.30	10.76	4.87	4.66	57.03	62.03	99.65	✓
[e] Bayesian uncertainty $\theta = 0.3$	0.59	12.07	5.11	4.95	79.74	68.96	99.23	✓
[e] Bayesian uncertainty $\theta = 0.4$	0.43	10.99	4.88	4.46	90.51	70.77	98.57	✓
[e] Bayesian uncertainty $\theta = 0.5$	0.45	10.08	3.93	3.78	94.79	74.73	98.00	✓

Table 4. Analysis of the labeling strategy of missing labels on Pascal VOC 2007 val set. For each metric, we report the relative scores with respect to a model that does not label missing labels. TP (resp. TN) means true positive (resp. true negative). Label proportion is the proportion of training labels (clean + weak labels) used at the end of the training. For the strategy labeling only positive labels, we report the label accuracy instead of the TP rate.

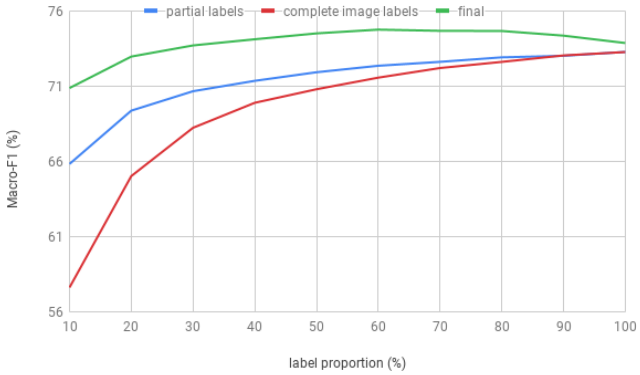
A.10. Final results

In Figure 11, we show the results of our final model that uses the partial-BCE loss, the GNN and the labeling of missing labels. We compare our model to two baselines: (a) a model trained with the standard BCE where the data are labeled with the partial labels strategy (blue) and (b) a model trained with the standard BCE where the data are labeled with the complete image labels strategy (red). We observe that our model has better performances than the two baselines for most of the metrics. In particular, our final model has significantly better 0-1 exact match performance than the baseline (b), whereas the baseline with partial labels (a) has lower performance than the baseline (b). We note that the overall precision of our model is worse than the baseline (b), but the overall recall of our model is largely better than the baseline (b).

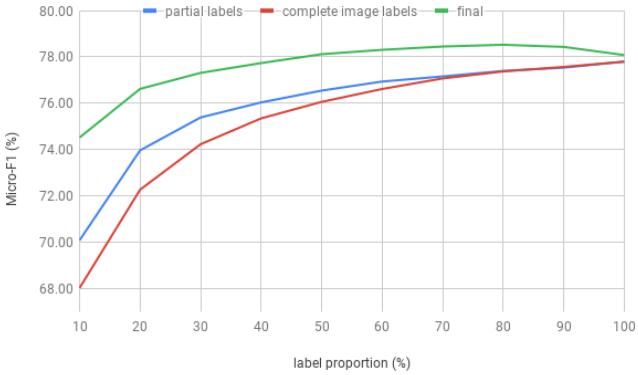


MAP

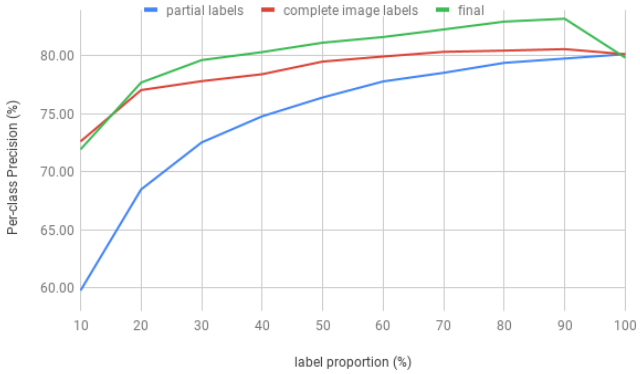
0-1 exact match



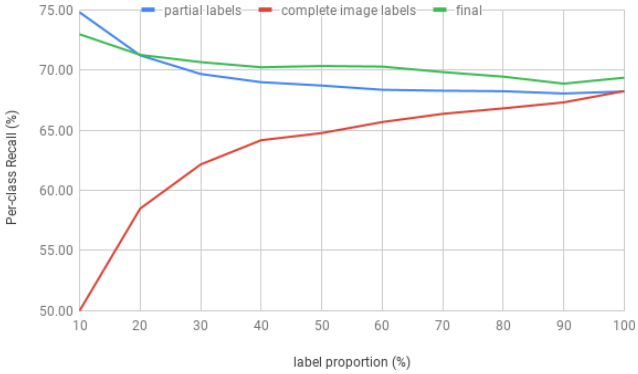
Macro-F1



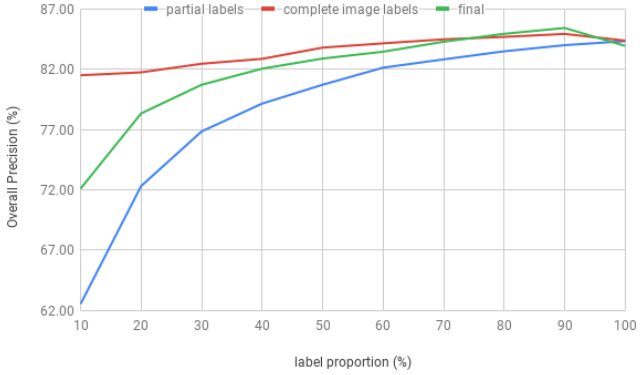
Micro-F1



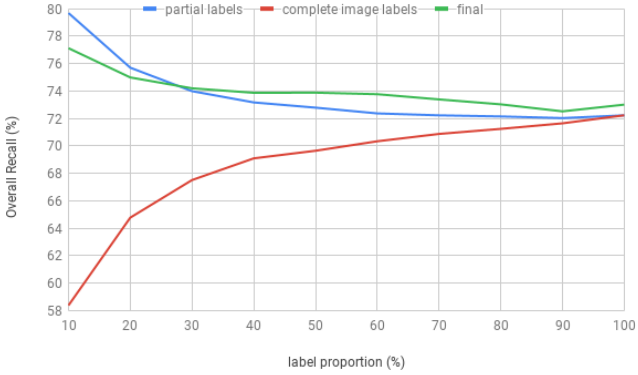
Per-class Precision



Per-class Recall



Overall Precision



Overall Recall

Figure 11. The results of our final model with two baselines (complete image labeling and BCE with partial labels) for different metrics on MS COCO val2014.

References

[1] 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 (CIVR)*, 2009. 1

[2] Thibaut Durand, Nicolas Thome, and Matthieu Cord. WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[3] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Exploiting Negative Evidence for Deep Latent Structured Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 1, 9

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 2015. 1

[5] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep Convolutional Ranking for Multilabel Image Annotation. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[6] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 16

[7] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. 2018. 1

[8] Yuncheng Li, Yale Song, and Jiebo Luo. Improving Pairwise Ranking for Multi-label Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollr. Microsoft COCO: Common Objects in Context. 2014. 1

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 1

[11] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3D Graph Neural Networks for RGBD Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[12] C. J. Van Rijsbergen. *Information Retrieval*. 1979. 2

[13] Lei Tang, Suju Rajan, and Vijay K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW*, 2009. 2

[14] 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)*, 2016. 16

[15] Yiming Yang. An evaluation of statistical approaches to text categorization. 1999. 2