

## 5 Supplementary materials

This section contains the supplementary materials.

### 5.1 Code and Demo

We share our implementation code to make it easy to reproduce our results. The source-code with the detailed running instructions is available [here](#)<sup>1</sup>. In addition to the code, we also include a ‘Demo.mp4’ video file that contains a demonstration of our framework. This is based on screen recording of a web application we built to showcase the use-cases of our method in real-world scenarios. Figure 7 shows a screenshot of the demo application.

### 5.2 Ablation studies on CIFAR-10, CIFAR-100, and ImageNet

Figure 8 shows the results of ablation studies of our EBJR method with different architectures for Student and Teacher models on CIFAR (10 and 100) and ImageNet. We observe that the results do not vary excessively, which shows the robustness of the proposed method.

### 5.3 More experiments with RANet

In this experiment, we evaluate the performance of EBJR when the SOTA architectures are used as our Student and Teacher models. In other words, we investigate whether our method can be added on top of other efficient methods such as RANet to benefit both from their designs and our joint inference. To this end, we trained the RANet architecture with three scales (as suggested in RANet work) on CIFAR-10, CIFAR-100, ImageNet. The accuracy and computational cost of the used Student and Teacher models for the three datasets are

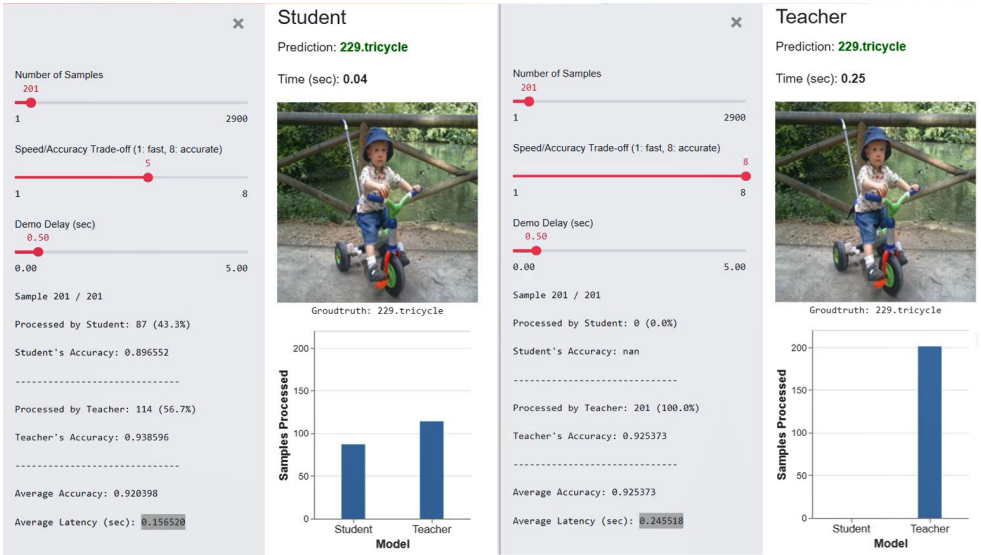


Figure 7: A screenshot of the provided demo application.

<sup>1</sup><https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=7c9eb8ea-4ace-4616-9a70-85d186e51962>

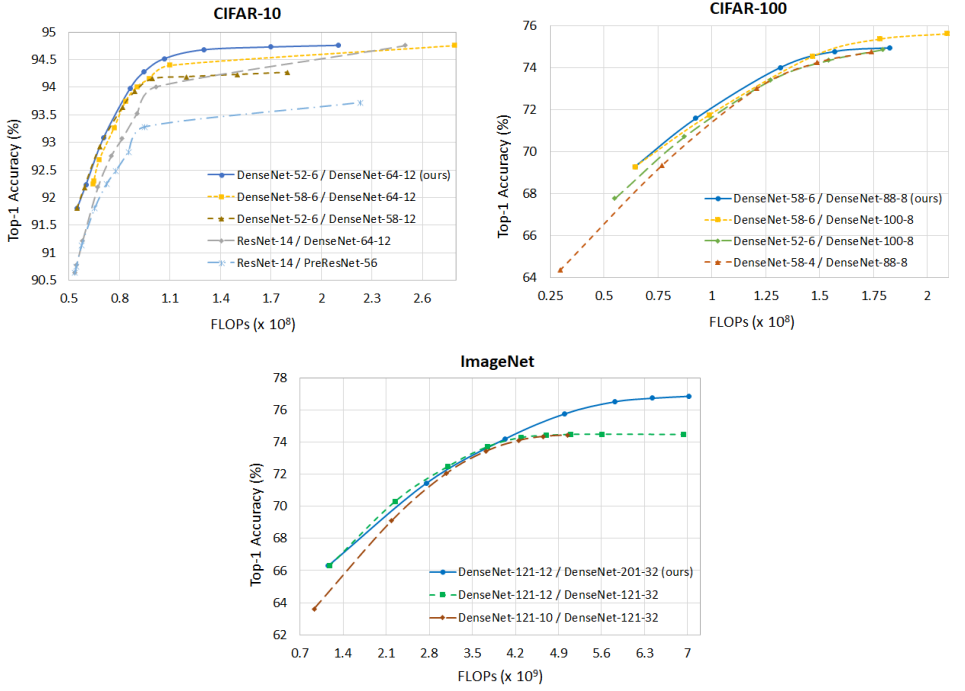


Figure 8: Ablation studies of EBJR on CIFAR-10, CIFAR-100, and ImageNet with different architectures for the Student and Teacher models. DenseNet- $d$ - $g$  denotes a densenet model with depth of  $d$  and growth rate of  $g$ . ResNet- $d$  denotes a resnet model with depth of  $d$ .

	CIFAR-10		CIFAR-100		ImageNet	
	S	T	S	T	S	T
<b>Accuracy (%)</b>	91.18	93.61	67.28	74.73	56.18	71.69
<b>FLOPs (<math>\times 10^8</math>)</b>	0.3162	1.898	0.3166	1.9	3.36	33.62

Table 3: Details of the Student (S) and Teacher (T) EBJR (RANet) models for CIFAR-10, CIFAR-100, and ImageNet experiments.

summarized in Table 3. For the Student, we employed the RANet’s first classifier from the first scale with  $0.316 (\times 10^8)$  FLOPs. For the Teacher, the last classifier from the last scale with  $1.89 (\times 10^8)$  FLOPs was used. Figure 9 shows the corresponding adaptive inference results compared with the RANet baseline on CIFAR-10, CIFAR-100, and ImageNet. We observe that our method is orthogonal to RANet, and can improve it further.

## 5.4 Alternative routing mechanisms: Softmax and Entropy

In Section 3.1, an ablation study with some experiments (Figure 3-right) was presented to analyze the softmax and entropy scores as alternative means of analyzing the Student. Here, we study the mathematical connection of them with the energy score and their potential to solve the routing problem.

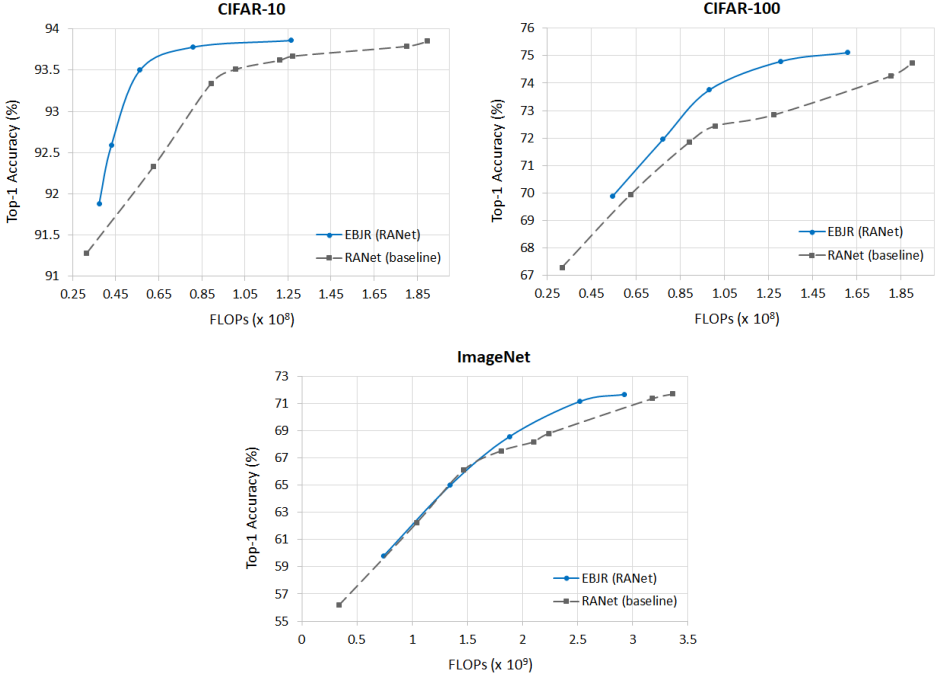


Figure 9: The performance of EBJR with RANet architecture, compared to the baseline RANet on CIFAR-10, CIFAR-100, and ImageNet.

### 5.4.1 Softmax-based Router

The softmax score for a classifier is expressed by:

$$\max_y p(y|\mathbf{x}) = \max_y \frac{e^{S_y^c(\mathbf{x})}}{\sum_i^C e^{S_i^c(\mathbf{x})}} = \frac{e^{S_{\max}^c(\mathbf{x})}}{\sum_i^C e^{S_i^c(\mathbf{x})}}. \quad (19)$$

By taking the logarithm of both sides, we start to see the connection between the log of the softmax and the free energy score formulated in (3):

$$\log \max_y p(y|\mathbf{x}) = \log(e^{S_{\max}^c(\mathbf{x})}) - \log \sum_i^C e^{S_i^c(\mathbf{x})} = S_{\max}^c(\mathbf{x}) + F^c(\mathbf{x}; S^c), \quad (20)$$

where all logits are shifted by their maximum logit  $S_{\max}^c(x)$ . Plugging in the energy term to (5) yields:

$$\log \max_y p(y|\mathbf{x}) = -\log(p(\mathbf{x})) + S_{\max}^c(\mathbf{x}) - \log(Z^c). \quad (21)$$

It is observed that for the samples with high likelihood of being in the Student’s distribution, the free energy goes lower, but the maximum logit tends to go higher. Due to this shifting, unlike the energy score, the softmax confidence score is not well-aligned with the probability density  $p(\mathbf{x})$ . As a result, the confidence score is less reliable for our Router to analyze the performance of the Student.

### 5.4.2 Entropy-based Router

The entropy score is a measure of the randomness in the information being processed, and is calculated as follows:

$$H(\mathbf{x}; S^c) = - \sum_i^C S_i^c \cdot \log(S_i^c), \quad (22)$$

where  $S_i^c(\mathbf{x})$  is the probability (logit) corresponding to the  $i$ -th class label.

Let  $U$  be the internal energy (i.e., the expectation value of the energy function [49]), defined by:

$$U(\mathbf{x}; S^c) = \sum_i^C E(\mathbf{x}, i) S_i^c. \quad (23)$$

According to [49], the entropy can be defined in terms of the internal and free energy functions as:

$$H(\mathbf{x}; S^c) = U(\mathbf{x}; S^c) - F(\mathbf{x}; S^c), \quad (24)$$

where all logits are shifted by the internal energy  $U$ .

Substituting the free energy term from (5) yields:

$$H(\mathbf{x}; S^c) = \log(p(\mathbf{x})) + U(\mathbf{x}; S^c) + \log(Z^c), \quad (25)$$

which shows that, due to the shifting caused by the internal energy, the entropy score is not reliably aligned with the probability density  $p(\mathbf{x})$ . Thus, it is a less suitable mechanism to be used as a routing mechanism in our Router, as opposed to the energy score.

## 5.5 Imbalance in class distributions

In Section 2.4, it was mentioned that in many practical applications, training or testing datasets are imbalanced. For example, consider a cloud inference API, which receives images as input, and most of the input images belong to a limited number of popular classes or categories. This motivated the specialized EBJR case. We studied the class distribution for the Caltech-256, OID, and MS-COCO datasets in Figures 10 and 11, and the statistics confirm our intuition.

## 5.6 More results on the specialized EBJR

Figure 12 shows the adaptive inference results for the specialized EBJR case. This figure shows the top-1 classification accuracy of joint models when top- $\bar{C}$ =10 or 20 popular classes are used. For top- $\bar{C}$ =10, we choose the top-10 class labels with the most number of samples in the OID training set, and for testing, we randomly select a new set of size 1.7K from the OID validation set, where 70% of the data have the top-10 of the class labels. For top- $\bar{C}$ =20, the size of the corresponding randomly selected validation set is 2K, where 75% of the samples belong to the top-20 labels. It should be noted that the Teacher accuracies over the top-10 and top-20 validation sets are not the same because the validation sets are not identical (different data/label distribution).

It is observed from Figure 12 that top- $\bar{C}$ =20 results in a better overall performance, and can achieve the same accuracy as the teacher but with  $1.35\times$  faster inference. Reducing the

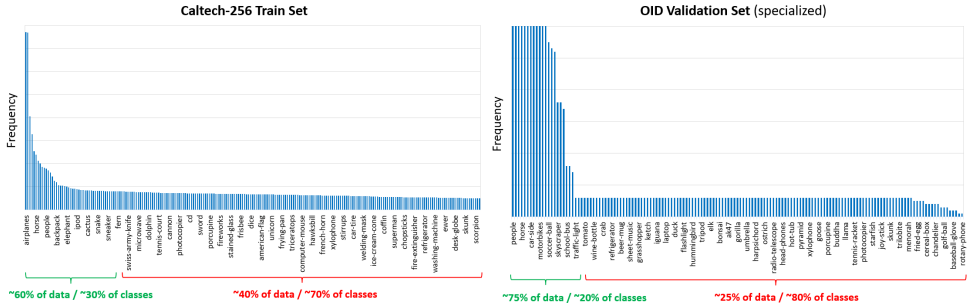


Figure 10: Data and class distributions over the Caltech-256 train set and OID validation set (for image classification with 256 labels). Due to the limited space, only a subset of class names are shown on the X axis for better visualization.

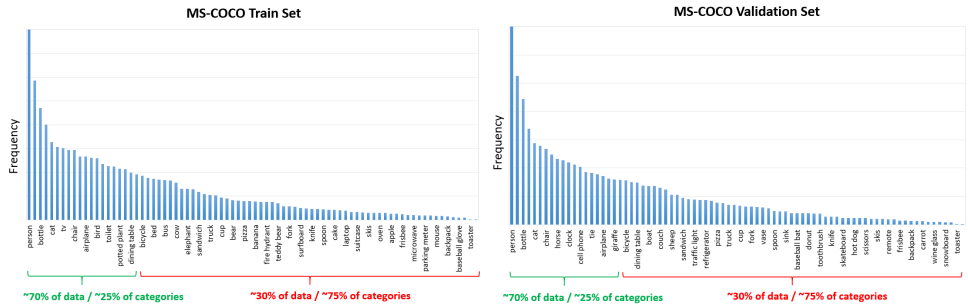


Figure 11: Data and category distributions over the MS-COCO train set and MS-COCO validation set (for object detection with 80 categories). Due to the limited space, only a subset of category names are shown on the X axis for better visualization.

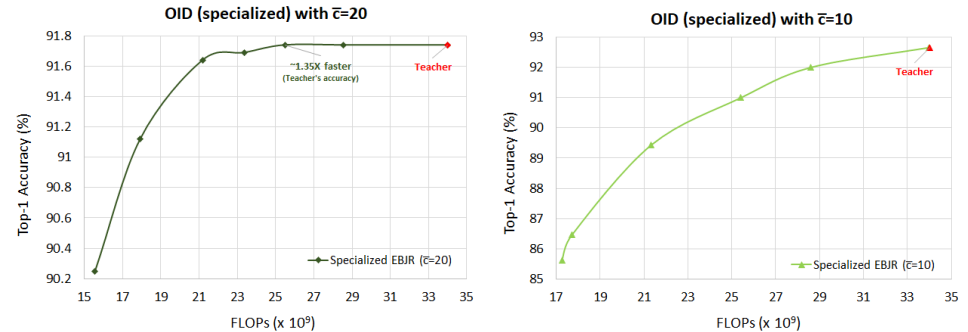


Figure 12: Adaptive inference results of the specialized EBJR with  $\bar{C} = 20$  and  $\bar{C} = 10$  on OID dataset.

	Caltech-256			OID		
	S	S*	T	S	S*	T
<b>Accuracy</b> (%)	83.16	70.71	89.87	75.0	74.1	86.64
<b>FLOPs</b> ( $\times 10^9$ )	5.4	5.4	34.0	5.4	5.4	34.0

Table 4: The performance of the supervised and unsupervised Student (respectively denoted by **S** and **S\***) and the Teacher (**T**) on Caltech-256 and OID validation sets.

number of classes to 10 will make the performance worse, where almost no speed-up can be achieved compared to the Teacher. This suggests that limiting the majority of popular categories to a very low number of classes may hurt the performance.

### 5.7 More insights on inequality (15)

The free energy of the Student  $S^c(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^C$  in (3) can be broken into the logarithm of two terms as:

$$F^c(\mathbf{x}; S^c) = -\log \left( e^{S_y^c} + \sum_i^{\bar{C}} e^{S_i^c(\mathbf{x})} \right), \quad (26)$$

where  $C = \bar{C} + 1$  and  $y \in \{\bar{C} + 1\}$ . Factoring out the term  $e^{S_y^c}$  from inside the logarithm yields:

$$F^c(\mathbf{x}; S^c) = -\log(e^{S_y^c}) - \log \left( 1 + \frac{\sum_i^{\bar{C}} e^{S_i^c(\mathbf{x})}}{e^{S_y^c}} \right). \quad (27)$$

By denoting the second term as  $\hat{F}^c(\mathbf{x}; S^c)$ , we will have:

$$F^c(\mathbf{x}; S^c) = -\left( S_y^c + \hat{F}^c(\mathbf{x}; S^c) \right). \quad (28)$$

Let  $(x, y)$  be in-distribution and  $(\bar{x}, \bar{y})$  be out-of-distribution samples, where  $y \in [1, \bar{C}]$  and  $\bar{y} \notin [1, \bar{C}]$ . Based on (28), the inequality (15) can be reformulated as:

$$\underbrace{|\bar{F}^c(\mathbf{x}; \bar{S}^c) - \bar{F}^c(\bar{\mathbf{x}}; \bar{S}^c)|}_{\text{decrease}} > \underbrace{\left( \underbrace{S_y^c}_{\text{decrease}} + \underbrace{\hat{F}^c(\bar{\mathbf{x}}; S^c)}_{\text{increase}} \right)}_{\text{increase}} - \underbrace{\left( \underbrace{S_y^c}_{\text{increase}} + \underbrace{\hat{F}^c(\mathbf{x}; S^c)}_{\text{decrease}} \right)}_{\text{decrease}}, \quad (29)$$

where the followings can be observed for the left side of this inequality:

- Since  $(x, y)$  is an in-distribution sample (high likelihood) and also  $y \in [1, \bar{C}]$ , the free energy  $\bar{F}^c(\mathbf{x}; \bar{S}^c)$  tends to be lower.
- Since  $(\bar{x}, \bar{y})$  is an out-of-distribution sample (low likelihood) and also  $\bar{y} \notin [1, \bar{C}]$ , the free energy  $\bar{F}^c(\bar{\mathbf{x}}; \bar{S}^c)$  tends to be higher.

And for the right side:

- For  $(x, y)$ , the  $y$ -th logit value  $S_y^c$  tends to increase (high likelihood), which makes  $\hat{F}^c(\mathbf{x}; S^c)$  decrease.
- For  $(\bar{x}, \bar{y})$ , the  $\bar{y}$ -th logit value  $S_{\bar{y}}^c$  tends to decrease (low likelihood), which makes  $\hat{F}^c(\bar{\mathbf{x}}; S^c)$  to increase.

The two terms on the left side tend to go in the opposite directions, thereby enlarging the energy difference. On the other hand, the two terms on the right side of (29) do not show a similar behaviour, and thus their gap does not necessarily increase.

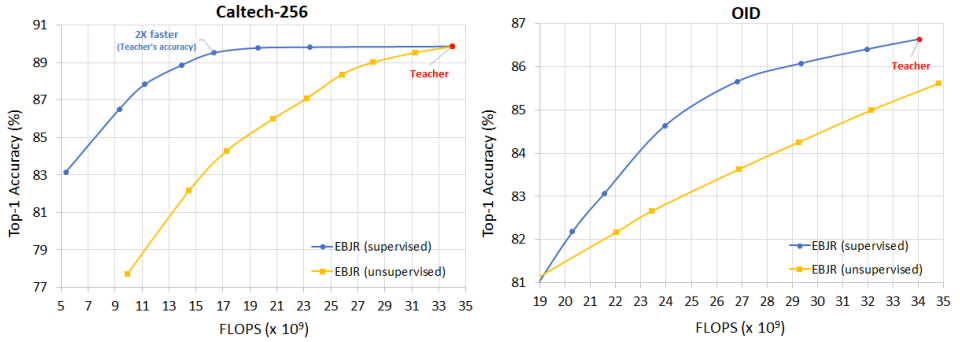


Figure 13: The performance of supervised and unsupervised EBJR on Caltech-256 and OID validation sets.

## 5.8 Unsupervised EBJR

So far, it was assumed that the Teacher and Student already-trained models are given, and with those we created a joint inference bundle model. This assumption may not always be true. Suppose we have a large model that is highly accurate, but also very slow. In addition, no dataset with ground truth labels is available to train a small and fast Student model. In this scenario, in order to achieve an efficient joint reasoning model, we can effectively distill the Teacher model to a small and fast Student architecture, in a completely unsupervised manner. Unsupervised knowledge distillation is an emerging technique for leveraging the abundance of unlabeled data for label-free model training. Our framework is flexible in that it can organically incorporate the unsupervised distillation.

The most straight-forward application of unsupervised EBJR is for cloud services, which include very large models for different machine learning tasks served through cloud APIs. Such inference services can be replaced by our EBJR architecture in which a side Student model is created for each large model. In this case, there is no need for re-training the large models nor acquiring data labels. By replacing the current large models behind the APIs with the joint reasoning equivalent, a speed-up gain can be achieved without a considerable loss in accuracy. For the classification problem, as an example, the commonly used cross-entropy loss function for training the Student is given by:

$$CE = - \sum_i^C \tau_i \log(S_i^c(\mathbf{x})) \quad \text{with} \quad \tau_i = T^c(\mathbf{x}), \quad (30)$$

where the pseudo-labels generated by the Teacher model are utilized as the targets (denoted by  $\tau_i$ ) in the loss function.

### 5.8.1 Experimental results - image classification

In this section, we study the performance of the unsupervised version of our method (see Section 5.8). To this end, we perform unsupervised distillation on the Student, using a set of unlabeled examples, which are passed to the Teacher to obtain pseudo-labels. The Student is then trained purely with these pseudo-labels. In this experiment, we use a ResNet-152 pre-trained on the Caltech-256 training set (22K examples in 256 classes) as the Teacher. The Student is a ResNet-18 trained with a 56K unlabeled random subset of OID.

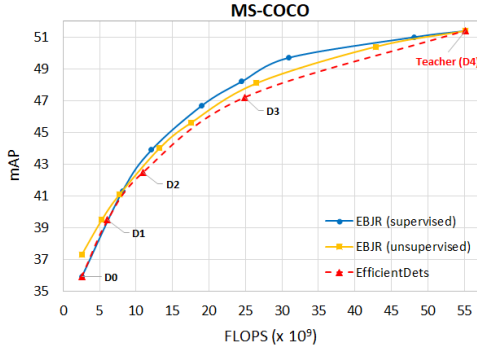


Figure 14: The adaptive inference performance of the supervised and unsupervised EBJR for object detection on MS-COCO (compared with the EfficientDet models [66]).

For testing, we evaluated our approach on two validation sets including Caltech-256 (7.8K images) and a subset of OID validation set (12K images). The accuracies and computational costs of the Student (supervised and unsupervised) and Teacher on both validations sets are reported in Table 4. Note that we study these two validations sets since they can both be valid measures depending on the target application. One represents the case when a user provides a large Teacher model with some validation data for which the joint model needs to attain a high accuracy. The other represents the case where a user provides a large Teacher model, and the joint model is supposed to work well for data that hit the cloud API, which are similar to the unlabeled data used to train the unsupervised joint model.

Figure 13 presents the adaptive inference results with the unsupervised EBJR and also its comparison with the supervised case on both validation sets. For the supervised EBJR, the Student is trained on Caltech-256 (similar to the Teacher). As observed in Figure 13-left, the unsupervised EBJR does not perform as well as the supervised case, which is because the distributions of the training and testing sets are different (OID vs. Caltech-256). However, when evaluated on the OID validation set, which follows the same distribution with which the Student is trained, a better performance is achieved (Figure 13-right).

It is shown in [65, 44, 52] that using large amounts of unlabeled data for pseudo-label self-training can achieve results even higher than the supervised models. In agreement with this observation, we will see later in this section that the performance of the unsupervised joint model tends to improve if larger amounts of unlabeled data are used. In some cases, it may even surpass the performance of the supervised model (see Figure 14). That being said, the results in Figure 13 are excellent for the supervised case, and still very promising for the unsupervised case, as the later is not using any labels for training the joint model.

### 5.8.2 Experimental results - object detection

We also analyze the performance of the unsupervised variant of EBJR on the task of object detection on the MS-COCO dataset, where we employ the EfficientDet-D0 and EfficientDet-D4 architectures [66] for the Student and Teacher, respectively. For the unsupervised EBJR, the OID training set is used as the unlabeled set.

Table 5 reports the performance of the Student model trained in the supervised and unsupervised settings, compared to the Teacher. For the unsupervised case, we tested different amounts of unlabeled data from OID. We observe that when sufficient unlabeled data (e.g., 1.7M in Table 5) are provided, the unsupervised Student can perform even better than the



Mode	Student				Teacher
	Supervised	Unsupervised			Supervised
Train-set size	118K	160K	320K	<b>1.7M</b>	118K
mAP	0.359	0.329	0.350	<b>0.373</b>	0.514
FLOPs ( $\times 10^9$ )	2.54				55.2

Table 5: The performance of the Teacher and the supervised and unsupervised Student with different train-set sizes on MS-COCO.

supervised one.

Moreover, Figure 14 shows the adaptive inference results for both supervised and unsupervised (with 1.7M samples from OID) cases compared to the EfficientDet models (D0, D1, D2, D3, and D4). Both supervised and unsupervised EBJR outperform the standard EfficientDet models.