Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation

Gongjie Zhang†, Zhipeng Luo†, Kaiwen Cui, Shijian Lu*, and Eric Xing

Abstract—Few-shot object detection has been extensively investigated by incorporating meta-learning into region-based detection frameworks. Despite its success, the said paradigm is still constrained by several factors, such as *(i)* low-quality region proposals for novel classes and *(ii)* negligence of the inter-class correlation among different classes. Such limitations hinder the generalization of base-class knowledge for the detection of novel-class objects. In this work, we design Meta-DETR, which *(i)* is the first image-level few-shot detector, and *(ii)* introduces a novel inter-class correlational meta-learning strategy to capture and leverage the correlation among different classes for robust and accurate few-shot object detection. Meta-DETR works entirely at image level without any region proposals, which circumvents the constraint of inaccurate proposals in prevalent few-shot detection frameworks. In addition, the introduced correlational meta-learning enables Meta-DETR to simultaneously attend to multiple support classes within a single feedforward, which allows to capture the inter-class correlation among different classes, thus significantly reducing the misclassification over similar classes and enhancing knowledge generalization to novel classes. Experiments over multiple few-shot object detection benchmarks show that the proposed Meta-DETR outperforms state-of-the-art methods by large margins. The implementation codes are available at https://github.com/ZhangGongjie/Meta-DETR.

Index Terms—Object Detection, Few-Shot Learning, Meta-Learning, Few-Shot Object Detection, Class Correlation.

1 Introduction

Computer vision has experienced significant progress in recent years. However, there still exists a huge gap between current computer vision techniques and the human visual system in learning new concepts from very few examples: most existing methods require a large amount of annotated samples, while humans can effortlessly recognize a new concept even with just a glimpse of it [1]. Such human-like capability to generalize from limited examples is highly desirable for machine vision systems, especially when sufficient training samples are unavailable or their annotations are hard to obtain.

In this work, we explore the challenging task of *few-shot object detection*, which requires detecting novel objects with only a few training samples. With minimal supervision on novel classes, the key to few-shot object detection is to learn transferable knowledge from base classes and generalize it to novel classes. To this end, many studies [2], [3], [4], [5], [6] incorporate meta-learning into generic region-based object detection frameworks, mostly Faster R-CNN [7], and have achieved very promising results.

Despite their success, there still exist two underlying limitations that hinder better exploitation of base-class knowledge, as illustrated in Fig. 2. *First*, region-based detection frameworks rely on region proposals to produce final predictions, thus are sensitive to low-quality region proposals. However, as investigated by [5] and [8], it is not easy to produce high-quality region proposals for novel classes with limited supervision under the few-shot detection setups. Such a gap in the quality of region proposals obstructs the generalization from base classes to novel classes. *Second*,

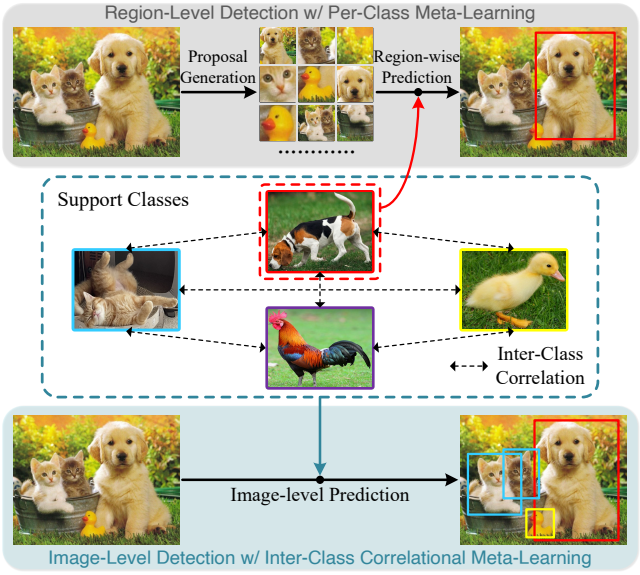
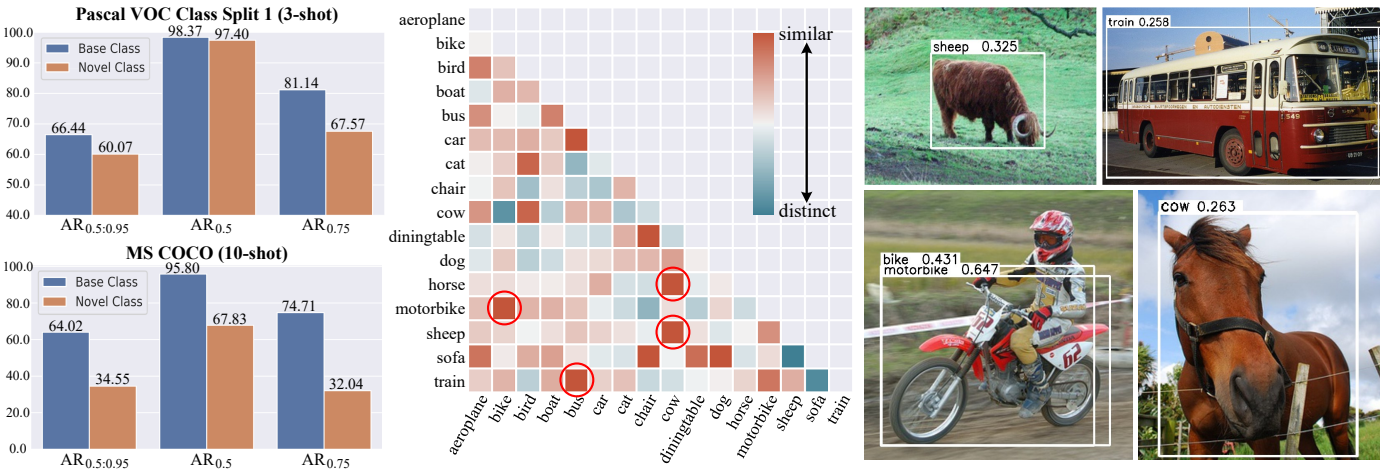


Fig. 1. Comparison of few-shot object detection pipelines: Prior studies (upper part) perform region-level detection, which are often constrained by inaccurate region proposals for novel classes. Besides, they can only deal with one support class at one go and overlook the correlation among different classes. The proposed Meta-DETR (lower part) works at image level without any proposals. It captures inter-class correlation by learning from multiple support classes simultaneously, which suppresses confusion among similar classes and enhances model generalization greatly.

- *Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
- *Eric Xing is with the School of Computer Science, Carnegie Mellon University. He also serves as the president of Mohamed bin Zayed University of Artificial Intelligence.*
- *E-mail: GongjieZhang@ntu.edu.sg, Zhipeng001@e.ntu.edu.sg, Kaiwen001@e.ntu.edu.sg, Shijian.Lu@ntu.edu.sg, Eric.Xing@mbzuai.ac.ae.*
- *† denotes equal contribution; ∗ denotes corresponding author.*

Manuscript received January 11, 2022, revised June 6, 2022.



(a) Quality Gap in Region Proposals **(b)** Cosine Similarity of Class Prototypes **(c)** Misclassification of Related Classes

Fig. 2. Existing few-shot detection frameworks tend to suffer from inaccurate region proposals and negligence of inter-class correlation. Due to very limited training samples for novel classes, the proposal quality (measured by Average Recall on top 1000 proposals) for novel classes is clearly lower than that of base classes as illustrated in (a). This hinders the knowledge generalization to novel classes. Additionally, object classes with similar appearances are highly correlated in feature space such as 'cow *vs.* horse' and 'motorbike *vs.* bike' as illustrated in (b), which tend to be misclassified if the learning does not incorporate the correlation among them as illustrated in (c).

most existing meta-learning-based approaches [2], [3], [4], [5] adopt 'feature reweighting' or its variants to aggregate query and support features, which can only deal with one support class (*i.e.*, target class to detect) at a time and essentially treat each support class independently. Without seeing multiple classes within a single feedforward, they largely overlook the important inter-class correlation among different support classes. This limits the ability to distinguish similar classes (*e.g.*, distinguishing from cows and horses) and to generalize from related classes (*e.g.*, learning to detect cows by generalizing from detecting sheep).

To mitigate the above limitations, we design Meta-DETR, an innovative few-shot object detector that performs pure image-level prediction and at the same time exploits the inter-class correlation among different classes. Fig. 1 illustrates its major differences with prior designs. To our best knowledge, this is the first work that identifies the constraint caused by region-based detection under the few-shot setups and explores to address few-shot object detection with DETR-based detection frameworks, which can skip proposal generation and directly perform detection at image level. With image-level prediction, Meta-DETR fully bypasses the constraint of inaccurate region proposals as in prevalent few-shot detection frameworks. In addition, the introduced inter-class correlational meta-learning strategy enables Meta-DETR to attend to multiple support classes at one go instead of class-by-class meta-learning with repeated runs as in most existing methods. By integrating detection tasks that involve multiple classes into meta-learning, Meta-DETR can explicitly leverage the inter-class correlation, including the inter-class commonality to facilitate generalization among related classes and the inter-class uniqueness to reduce misclassification among similar classes.

In summary, the contributions of this work are threefold. *First*, we identify the quality gap of proposals for base and novel classes in region-based prediction, and propose Meta-DETR to address few-shot object detection. Being the first pure image-level few-shot detector, Meta-DETR fully circumvents the gap of inaccurate proposals for novel-class objects, thus enabling better generalization to novel classes. *Second*, we design a novel correlational meta-learning strategy, which can deal with multiple support classes simultaneously. It effectively exploits inter-class correlation among different classes, thus greatly reducing misclassification and enhancing model generalization. *Third*, extensive experiments show that, without bells and whistles, the proposed Meta-DETR consistently outperforms state-of-the-art methods by large margins on detecting novel objects.

2 Related Work

2.1 Object Detection

Generic object detection [9] is a joint task on object localization and classification. Modern object detectors are mostly region-based and can be broadly classified into two categories: two-stage and single-stage detectors. Two-stage detectors include Faster R-CNN [7] and its variants [10], [11], [12], which first adopt a Region Proposal Network (RPN) to generate region proposals, and then produce final predictions based on the proposals. Differently, single-stage detectors [13], [14], [15] employ densely placed anchors as region proposals and directly make predictions over them. Recently, another line of research featuring DETR [16] and its variants [17], [18], [19], [20] has received vast attention, thanks to the merits of pure image-level framework, fully end-to-end pipeline, and comparable or even better performance. However, these aforementioned generic detectors still heavily rely on large amounts of annotated training samples, thus will suffer from drastic performance drop when directly applied to few-shot object detection.

2.2 Few-Shot Object Detection

Existing works on few-shot object detection can be categorized into two paradigms: transfer learning and meta-learning. Methods with transfer learning mainly include LSTD [21], TFA [22], MPSR [23], and FSCE [24], where novel

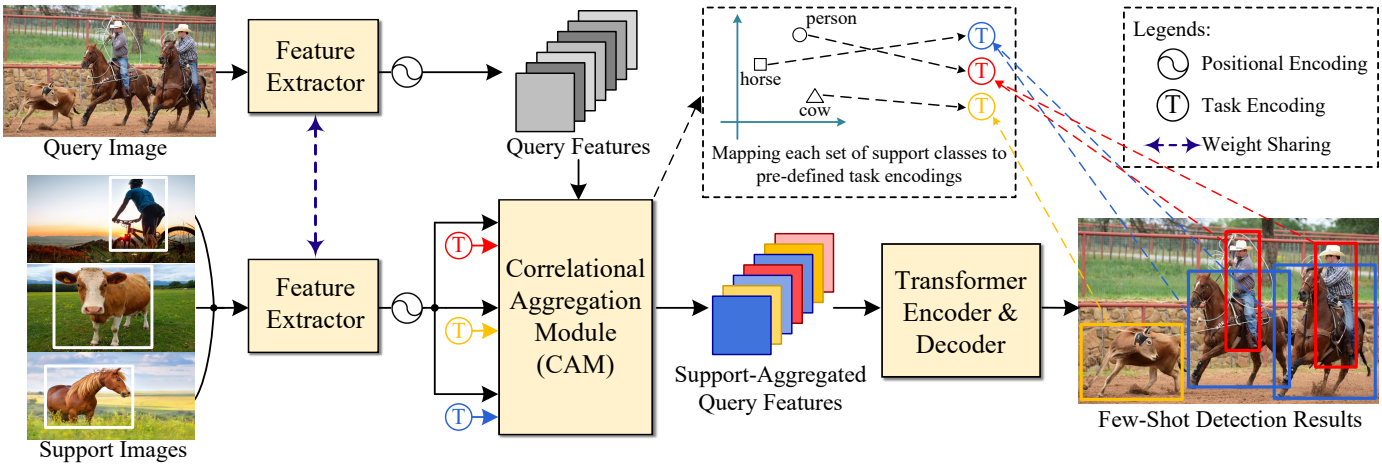


Fig. 3. The framework of Meta-DETR: *Query Image* and *Support Images* are processed by a weight-shared *Feature Extractor* to produce *Query Features* and support features. To leverage the inter-class correlation in meta-learning, a *Correlational Aggregation Module (CAM)* is designed, which first matches the query features with multiple support classes simultaneously and then introduces multiple task encodings (*i.e.*, the three illustrative Ⓣ of different colors) to differentiate these support classes. Finally, few-shot detection is achieved with a class-agnostic *Transformer Encoder & Decoder* that learns to predict objects' locations and their corresponding task encodings (instead of directly predicting objects' class labels). The architecture of CAM is detailed in Section 4.2 and Fig. 4. The training objectives of Meta-DETR are formulated in Section 4.3.

concepts are learned via fine-tuning. Differently, methods with meta-learning [2], [3], [4], [5], [6], [25], [26] extract knowledge that can generalize across various tasks via 'learning to learn', *i.e.*, learning a class-agnostic predictor on various auxiliary tasks.

Our proposed Meta-DETR falls under the umbrella of meta-learning, but differs from existing approaches by achieving image-level detection and effectively leveraging the correlation among various support classes. To the best of our knowledge, Meta-DETR is the first work that incorporates meta-learning into the recently proposed DETR frameworks. It is also the pioneering work to explicitly integrate the inter-class correlation among support classes into few-shot object detection frameworks using meta-learning.

3 Preliminaries

3.1 Problem Definition

Given two sets of classes $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \varnothing$, a few-shot object detector aims at detecting objects of $\mathcal{C}_{\text{base}} \cup \mathcal{C}_{\text{novel}}$ by learning from a base dataset $\mathcal{D}_{\text{base}}$ with abundant annotated objects of $\mathcal{C}_{\text{base}}$ and a novel dataset $\mathcal{D}_{\text{novel}}$ with very few annotated objects of $\mathcal{C}_{\text{novel}}$. In the task of K-shot object detection, there are exactly K annotated objects for each novel class in $\mathcal{D}_{\text{novel}}$.

3.2 Rethink Region-Based Detection Frameworks

Most existing few-shot object detectors are developed on top of Faster R-CNN [7], a region-based object detector, thanks to its robust performance and ease for optimization. However, by relying on region proposals to produce detection results, these approaches are inevitably constrained by the inaccurate proposals for novel classes due to very limited supervision under the few-shot detection setups. As illustrated in Fig. 2(a), there is a clear gap in the quality of region proposals for base and novel classes, hindering region-based detection frameworks from exploiting base-class knowledge to generalize to novel classes. Though several studies [5], [8] attempt to acquire more accurate region proposals, this issue still remains as it is rooted in the region-based detection frameworks under the few-shot learning setups.

3.3 Rethink Meta-Learning via Feature Reweighting

To meta-learn a class-agnostic detector that can generalize across various classes, most existing methods [2], [3], [4], [5] adopt 'feature reweighting' or its variants to aggregate query features with support class information, acquiring class-specific meta-features to detect objects corresponding to the support class. However, such meta-learning strategies can deal with only one support class within each feed-forward process, *i.e.*, C repeated runs are required to detect C support classes within each query image. More importantly, by treating each support class independently, 'feature reweighting' overlooks the essential inter-class correlation among different support classes. As shown in Fig. 2(b), many object classes with similar appearances are highly correlated. Intuitively, their correlation can effectively facilitate the distinction and the generalization among similar classes. However, as shown in Fig. 2(c), we observe that objects misclassified as highly correlated classes constitute a major source of error due to the negligence of inter-class correlation in existing methods.

4 Meta-DETR

This section provides a detailed description of the proposed Meta-DETR, including its network architecture, training objective, as well as the learning and inference procedure.

4.1 Model Overview

Fig. 3 shows the architecture of the proposed Meta-DETR. Motivated by previous discussions, Meta-DETR employs the recently proposed Deformable DETR [17], a fully end-to-end Transformer-based [27] detector, as the basic detection framework. As Meta-DETR does not rely on predicted region proposals to make final predictions, it can fully

bypass the constraint of inaccurate proposals on novel-class objects. Besides, thanks to the introduced correlational meta-learning, Meta-DETR can aggregate query features with multiple support classes simultaneously, thus capturing and leveraging the inter-class correlation among different classes to reduce misclassification and boost generalization.

Given a query image and a set of support images with instance annotations, a weight-shared feature extractor first encodes them into the same feature space. Subsequently, a *Correlational Aggregation Module (CAM)*, which will be introduced later, performs simultaneous aggregation between the query features and the set of support classes. To differentiate between different support classes in a class-agnostic manner, CAM introduces a set of task encodings assigned to each support class. Finally, a transformer architecture detects objects by predicting their locations and corresponding task encodings. As the detection targets are dynamically determined by support classes and their mappings to task encodings, Meta-DETR is trained as a meta-learner to extract generalizable knowledge not specific to certain classes.

4.2 Inter-Class Correlational Meta-Learning

The *Correlational Aggregation Module (CAM)* is the key component in Meta-DETR to perform inter-class correlational meta-learning, which aggregates query features with support classes for the subsequent class-agnostic prediction. CAM differs from existing aggregation methods in that it can aggregate multiple support classes simultaneously, which enables it to capture their inter-class correlation to reduce misclassification and enhance model generalization. Specifically, as illustrated in Fig. 4, the query and support features are first processed by a weight-shared multi-head attention module, encoding them into the same embedding space. Then the prototype for each support class is obtained by applying RoIAlign [28], followed by average pooling on the support features, where RoIAlign ensures that class prototypes are obtained from the relevant regions that contain corresponding support object instances. After that, CAM performs feature matching and encoding matching, which will be elaborated in the remainder of this subsection to match the query features with support class prototypes and task encodings, respectively. The matching results are summed together and fed to a feed-forward network (FFN) to produce the final output. Note that the support class prototypes are obtained in CAM before feature matching and encoding matching.

4.2.1 Feature Matching

Feature matching, which aims to filter out features irrelevant to support classes, is achieved by an attention mechanism with minor modifications. Specifically, given a query feature map $\mathbf{Q} \in \mathbb{R}^{HW \times d}$ and the support class prototypes $\mathbf{S} \in \mathbb{R}^{C \times d}$, where HW is the spatial size, C is the number of support classes, and d is the feature dimensionality, the matching coefficients are obtained via:

$$\mathbf{A} = \mathrm{Attn}(\mathbf{Q}, \mathbf{S}) = \mathrm{Softmax}(\frac{(\mathbf{QW})(\mathbf{SW})^{\mathrm{T}}}{\sqrt{d}}), \quad (1)$$

where $\mathbf{W}$ is a linear projection shared by $\mathbf{Q}$ and $\mathbf{S}$, which ensures they are embedded into the same feature space.

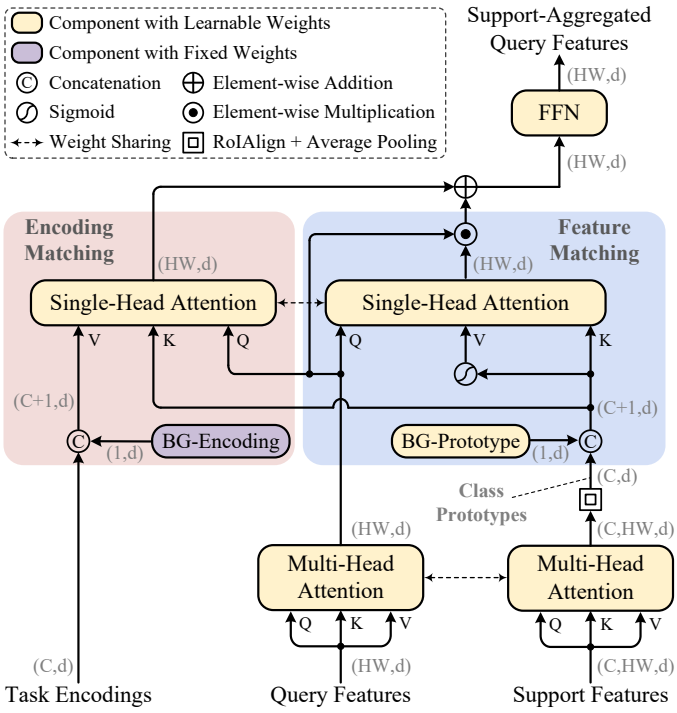


Fig. 4. The architecture of the Correlational Aggregation Module (CAM). CAM first obtains class prototypes from support features. Then, it performs two matching processes: *Feature Matching* filters out query features that are unrelated to support classes, while *Encoding Matching* matches query features to a set of pre-defined task encodings that differentiate their corresponding support classes in a class-agnostic manner.

Subsequently, the output of the feature matching module can be obtained via:

$$\mathbf{Q_F} = \mathbf{A}\sigma(\mathbf{S}) \odot \mathbf{Q}, \quad (2)$$

where $\sigma(\cdot)$ denotes sigmoid function and $\odot$ denotes Hadamard product. $\sigma(\mathbf{S})$ serves as feature filters for each individual support class with the function of extracting only class-related features from query features. By applying the matching coefficients $\mathbf{A}$ to $\sigma(\mathbf{S})$, we have filters that can filter out query features that are not matched to any support class, producing a filtered query feature map $\mathbf{Q_F}$ that only highlights objects belonging to the given support classes.

4.2.2 Encoding Matching

To achieve correlational meta-learning, we introduce a set of pre-defined task encodings assigned to each support class and match query features to their corresponding task encodings, so that final predictions can be made on the task encodings instead of specific classes. We implement task encodings $\mathbf{T} \in \mathbb{R}^{C \times d}$ with sinusoidal functions, following the positional encodings of the Transformer [27]. Encoding matching uses the same matching coefficients as feature matching, and the matched encodings $\mathbf{Q_E}$ are obtained via:

$$\mathbf{Q_E} = \mathbf{AT}. \quad (3)$$

4.2.3 Modeling Background for Open-Set Prediction

Object detection features an open-set setup where background, which does not belong to any of the target classes, often takes up most of the spatial locations in a query image.

Therefore, as shown in Fig. 4, we additionally introduce a learnable prototype and a corresponding task encoding (fixed to zeros), denoted as BG-Prototype and BG-Encoding respectively, to explicitly model the background class. This eliminates the matching ambiguity when query does not match any of the given support classes.

4.3 Training Objective

4.3.1 Target Generation

We let N denote the fixed number of object queries, which means Meta-DETR infers N predictions within a single feed-forward process. Let x_{query} denote the query image, and $y=\{y_i\}_{i=1}^{N}$ denote the ground truth objects within the query image, where y is a set of size N. When y_i indicates an object, $y_i=(c_i,b_i)$, where c_i denotes the target class label and b_i denotes the bounding box of the object. When y_i indicates no object, $y_i=(\varnothing,\varnothing)$.

Meta-DETR dynamically conditions its detection targets on the sampled support classes and their mappings to the task encodings. As discussed in Section 4.1, Meta-DETR predicts over C support classes (*i.e.*, target classes) simultaneously. The C support classes are randomly sampled, denoted as $c_{\text{supp}}=\{s_i\}_{i=1}^{C}$. Besides, these support classes are further mapped to a set of task encodings. We denote the mapping function from the labels of support classes to the labels of task encodings as $\chi(\cdot)$. A specific case of $\chi(\cdot)$ can be formulated as:

$$\chi(s_i)=i \qquad i\in\{1,2,\cdots,C\}. \tag{4}$$

Note that the exact format of the mapping function $\chi(\cdot)$ does not matter. Then, the detection targets of Meta-DETR can be formulated as:

$$y'=\{y'_i\}_{i=1}^{N}=\{(c'_i,b'_i)\}_{i=1}^{N}=\{\psi(y_i,c_{\text{supp}})\}_{i=1}^{N}, \tag{5}$$

where $\psi(y_i,c_{\text{supp}})$ acts to remove annotations of irrelevant objects (objects with labels not in c_{supp}) and to map the labels of target classes to the labels of the corresponding task encodings, which can be formulated as:

$$\psi(y_i,c_{\text{supp}})=\begin{cases}(\varnothing,\varnothing), & \text{if } y_i=(\varnothing,\varnothing) \text{ or } \mathrm{c_i}\notin \mathrm{c_{supp}}\\ (\chi(c_i),b_i), & \text{if } c_i\in c_{\text{supp}}.\end{cases} \tag{6}$$

Note that y' can completely consist of $(\varnothing,\varnothing)$ when there is no objects that belong to the provided support classes.

4.3.2 Loss Function

Assume the N predictions for target class made by Meta-DETR are $\hat{y}=\{\hat{y}_i\}_{i=1}^{N}=\{(\hat{c}_i,\hat{b}_i)\}_{i=1}^{N}$. We adopt a pair-wise matching loss $\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)})$ to search for a bipartite matching between $\hat{y}$ and y' with the lowest cost:

$$\hat{\sigma}=\arg\min_{\sigma}\sum\nolimits_{i=1}^{N}\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)}), \tag{7}$$

where σ denotes a permutation of N elements, and $\hat{\sigma}$ denotes the optimal assignment between predictions and targets. Since the matching should consider both classification and localization, the matching loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)})=&\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{cls}}(c'_i,\hat{c}_{\sigma(i)})+\\ &\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{box}}(b'_i,\hat{b}_{\sigma(i)})\ .\end{aligned} \tag{8}$$

With the optimal assignment $\hat{\sigma}$ obtained with Eq. 7 and Eq. 8, we optimize the network using the following loss function:

$$\mathcal{L}(y',\hat{y})=\sum_{i=1}^{N}\left[\mathcal{L}_{\text{cls}}(c'_i,\hat{c}_{\hat{\sigma}(i)})+\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{box}}(b'_i,\hat{b}_{\hat{\sigma}(i)})\right], \tag{9}$$

where we adopt sigmoid focal loss [29] for $\mathcal{L}_{\text{cls}}$ and adopt a linear combination of $\ell1$ loss and GIoU loss [30] for $\mathcal{L}_{\text{box}}$. Similar to DETR [16] and Deformable DETR [17], $\mathcal{L}(y',\hat{y})$ is applied to every layer of the transformer decoder.

Following Meta R-CNN [3], we introduce a cosine similarity cross-entropy loss [31] to classify the class prototypes obtained by our designed CAM. It encourages prototypes of different classes to be distinguished from each other.

4.4 Training and Inference Procedure

4.4.1 Two-Stage Training Procedure

The training procedure consists of two stages. The first stage is *base training stage*. During this stage, the model is trained on the base dataset $\mathcal{D}_{\text{base}}$ with abundant training samples for each base class. The second stage is *few-shot fine-tuning stage*. In this stage, we train the model on both base and novel classes with limited training samples. Only K object instances are available for each novel category in K-shot object detection. Following prior works [3], [4], [22], we also include objects from base classes to prevent performance drop for base classes. In both *base training* and *few-shot fine-tuning* stages, the whole network is optimized in an end-to-end manner with the same training objective described in Section 4.3.

4.4.2 Efficient Inference

Unlike the training stage, there is no need to repeatedly sample support images and extract their features with the feature extractor. We can first compute the prototype for each support class once and for all, then directly use them for every query image to predict. This promises efficient inference of our proposed Meta-DETR.

5 Experiments

5.1 Datasets

We follow the well-established data setups for few-shot object detection [2], [3], [4], [22], [25]. Concretely, two widely used few-shot object detection benchmarks are adopted in our experiments.

Pascal VOC [37] is a commonly used dataset for object detection that consists of images with object annotations of 20 classes. We use *trainval 07+12* for training and perform evaluations on *test 07*. We use 3 novel / base class splits, *i.e.*, ("bird", "bus", "cow", "motorbike", "sofa" / others), ("aeroplane", "bottle","cow","horse","sofa" / others), and ("boat", "cat", "motorbike","sheep", "sofa" / others). The number of shots is set to 1, 2, 3, 5 and 10. Mean average precision at IoU threshold 0.5 (mAP@0.5) is used as the evaluation metric. Results are averaged over 10 randomly sampled support datasets.

TABLE 1
Few-shot detection performance (mAP@0.5) on Pascal VOC for novel classes

	Class Split 1					Class Split 2					Class Split 3					*Avg.*
Method \ Shots	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10	
Results over a single run:																
LSTD [21]	8.2	1.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3	17.1
RepMet [32] ‡	26.1	32.9	34.4	38.6	41.3	17.2	22.1	23.4	28.3	35.8	27.5	31.1	31.5	34.4	37.2	30.8
Meta-YOLO [2]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9	28.4
Meta Det [25]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1	31.0
Meta R-CNN [3]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1	31.1
TFA w/ fc [22] ‡	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2	38.7
TFA w/ cos [22] ‡	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8	39.9
MPSR [23] ‡	41.7	43.1	51.4	55.2	61.8	24.4	29.5	39.2	39.9	47.8	35.6	40.6	42.3	48.0	49.7	43.3
TFA w/ cos + Halluc [33] ‡	45.1	44.0	44.7	55.0	55.9	23.2	27.5	35.1	34.9	39.0	30.5	35.1	41.4	49.0	49.3	40.6
Retentive R-CNN [34] ‡	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1	41.1
CME [35] ‡	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5	44.4
SRR-FSD [36] ‡⊎	**47.8**	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4	44.8
FSCE [24] ‡	44.2	43.8	51.4	**61.9**	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5	46.6
Meta-DETR (Ours)	40.6	**51.4**	**58.0**	59.2	**63.6**	**37.0**	**36.6**	**43.7**	**49.1**	**54.6**	**41.6**	**45.9**	**52.7**	**58.9**	**60.6**	**50.2**
Results averaged over multiple random runs:																
FRCN+ft-full [7] ‡	9.9	15.6	21.6	28.0	35.6	9.4	13.8	17.4	21.9	29.8	8.1	13.9	19.0	23.9	31.0	19.9
Deformable-DETR+ft-full [17] ‡	5.6	13.3	21.7	34.2	45.0	10.9	13.0	18.4	27.3	39.4	7.3	16.6	20.8	32.2	41.8	23.2
TFA w/ fc [22] ‡	22.9	34.5	40.4	46.7	52.0	16.9	26.4	30.5	34.6	39.7	15.7	27.2	34.7	40.8	44.6	33.8
TFA w/ cos [22] ‡	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6	34.7
FsDetView [4]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6	36.7
MPSR [23] ‡△	34.7	42.6	46.1	49.4	56.7	22.6	30.5	31.0	36.7	43.3	27.5	32.5	38.2	44.6	50.0	39.1
DCNet [6] ‡	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7	39.2
FSCE [24] ‡	32.9	44.0	46.8	52.9	59.7	23.7	30.6	**38.4**	43.0	48.5	22.6	33.4	39.5	47.3	54.0	41.2
Meta-DETR (Ours)	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**	**27.9**	**32.3**	**38.4**	**43.2**	**51.8**	**34.9**	**41.8**	**47.1**	**54.1**	**58.2**	**45.8**

"‡" indicates methods using multi-scale features. *"△" indicates re-evaluated results using official codes.* *"⊎" indicates usage of external data.*

TABLE 2
Few-shot detection performance (mAP@0.5)
on Pascal VOC class split 1 for both base and novel classes

	Base Classes				Novel Classes			
Method \ Shots	1	3	5	10	1	3	5	10
Meta-YOLO [2]	66.4	64.8	63.4	63.6	14.8	26.7	33.9	47.2
FsDetView [4] §	64.2	69.4	69.8	71.1	24.2	42.2	49.1	57.4
TFA w/ cos [22] §	**77.6**	**77.3**	**77.4**	**77.5**	25.3	42.1	47.9	52.9
MPSR [23] §	60.6	65.9	68.2	69.8	34.7	46.1	49.4	56.7
FSCE [24] §	75.5	73.7	75.0	75.2	32.9	46.8	52.9	59.7
Meta-DETR (Ours) §	67.2	70.0	73.0	73.5	**35.1**	**53.2**	**57.4**	**62.0**

"§" indicates results averaged over multiple random runs.

MS COCO [38] is a more challenging object detection dataset, which contains 80 classes including those 20 classes in Pascal VOC. We adopt the 20 shared classes as novel classes, and adopt the remaining 60 classes as base classes. The number of shots is set to 1, 3, 5, 10, and 30. We use *train 2017* for training, and perform evaluations on *val 2017*. Standard evaluation metrics for MS COCO are adopted. Results are averaged over 5 randomly sampled support datasets.

5.2 Implementation Details

We adopt the commonly used ResNet-101 [39] as the feature extractor. The network architectures and hyper-parameters remain the same as Deformable DETR [17]. We implement our model in single-scale version for fair comparison with other works. We also follow FsDetView [4] to implement the aggregation with a slightly more complex scheme compared with solely feature reweighting. We train our model with 8 x Nvidia V100 GPUs, using the AdamW [40] optimizer with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. Batch size is set to 32. In the base training stage, we train the model for 50 and 25 epochs for Pascal VOC and MS COCO, respectively. Learning rate is decayed at the 45^{th} and 20^{th} epoch by 0.1. In the few-shot fine-tuning stage, the same settings are applied to fine-tune the model until convergence.

5.3 Comparison with State-of-the-Art Methods

5.3.1 Pascal VOC

Table 1 shows the few-shot detection performance for novel classes of Pascal VOC. It can be seen that Meta-DETR consistently outperforms existing methods across various setups. With multiple runs over randomly sampled support datasets to reduce randomness, Meta-DETR achieves the best average performance across all setups, with a large margin of +4.6% overall mAP compared with the second-best. The strong performance demonstrates the superiority and robustness of our proposed Meta-DETR.

We also present results taking base classes into consideration in Table 2. While achieving good performance for novel classes with limited training samples, Meta-DETR can still detect objects of base classes with competitive performance. TFA [22] produces outstanding performance for base classes since it only fine-tunes detector's last layer, thus having relatively constrained capacity in generalizing on novel classes. We would highlight that our proposed Meta-DETR achieves the best base-class and novel-class performance among all compared methods using meta-learning (*i.e.*, Meta-YOLO [2] and FsDetView [4]).

TABLE 3
Few-shot detection performance on COCO for novel classes

Shot	Method	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$
1	FRCN+ft-full [7] ‡ §	1.7	3.3	1.6
	Deformable-DETR+ft-full [17] §	1.8	3.1	1.8
	TFA w/ cos [22] ‡ §	1.9	3.8	1.7
	TFA w/ cos + Halluc [33] ‡	3.8	6.5	4.3
	Meta-DETR (Ours) §	**7.5**	**12.5**	**7.7**
3	FRCN+ft-full [7] ‡ §	3.7	7.1	3.5
	Deformable-DETR+ft-full [17] §	4.9	7.8	5.1
	TFA w/ cos [22] ‡ §	5.1	9.9	4.8
	TFA w/ cos + Halluc [33] ‡	6.9	12.6	7.0
	Meta-DETR (Ours) §	**13.5**	**21.7**	**14.0**
5	FRCN+ft-full [7] ‡ §	4.6	8.7	4.4
	Deformable-DETR+ft-full [17] §	7.4	12.3	7.7
	TFA w/ cos [22] ‡ §	7.0	13.3	6.5
	FsDetView [4] §	10.7	24.5	6.7
	Meta-DETR (Ours) §	**15.4**	**25.0**	**15.8**
10	FRCN+ft-full [7] ‡ §	5.5	10.0	5.5
	Deformable-DETR+ft-full [17] §	11.7	19.6	12.1
	Meta-YOLO [2]	5.6	12.3	4.6
	Meta Det [25]	7.1	14.6	6.1
	Meta R-CNN [3]	8.7	19.1	6.6
	TFA w/ cos [22] ‡ §	9.1	17.1	8.8
	FSOD [5]	12.0	22.4	11.8
	FsDetView [4] §	12.5	27.3	9.8
	MPSR [23] ‡	9.8	17.9	9.7
	SRR-FSD [36] ‡ ⊎	11.3	23.0	9.8
	CME [35] ‡	15.1	24.6	16.4
	DCNet [6] ‡ §	12.8	23.4	11.2
	FSCE [24] ‡ §	11.1	-	9.8
	Meta-DETR (Ours) §	**19.0**	**30.5**	**19.7**
30	FRCN+ft-full [7] ‡ §	7.4	13.1	7.4
	Deformable-DETR+ft-full [17] §	16.3	27.2	16.7
	Meta-YOLO [2]	9.1	19.0	7.6
	Meta Det [25]	11.3	21.7	8.1
	Meta R-CNN [3]	12.4	25.3	10.8
	TFA w/ cos [22] ‡ §	12.1	22.0	12.0
	FsDetView [4] §	14.7	30.6	12.2
	MPSR [23] ‡	14.1	25.4	14.2
	SRR-FSD [36] ‡ ⊎	14.7	29.2	13.5
	CME [35] ‡	16.9	28.0	17.8
	DCNet [6] ‡ §	18.6	32.6	17.5
	FSCE [24] ‡ §	15.3	-	14.2
	Meta-DETR (Ours) §	**22.2**	**35.0**	**22.8**

"‡" indicates methods using multi-scale features.
"§" indicates results averaged over multiple runs.
"⊎" indicates usage of external data.

5.3.2 MS COCO

Table 3 shows experimental results on MS COCO. It can be seen that, although MS COCO is much more challenging than Pascal VOC with higher complexity like occlusions and large scale variations, Meta-DETR still outperforms all existing methods under all setups by even larger margins. This can be attributed to *(i)* the complete circumvention of even more inaccurate region proposals for novel classes (See Fig. 2(a)) caused by the higher complexity of MS COCO, and *(ii)* the effective exploitation of the correlations among more classes in MS COCO. In addition, Meta-DETR performs exceptionally well compared with other region-based methods under the stricter metric $AP_{0.75}$, which implies that our proposed Meta-DETR can effectively lift the constraint of inaccurate region proposals and produce more accurate few-shot object detection.

TABLE 4
Ablation study on region-level detection *vs.* image-level detection

Method	aligned network	R/I	Novel mAP@0.5 1	2	3	5	10
FsDetView [4]		R	24.2	35.3	42.2	49.1	57.4
FsDetView + Deform. Trans.	✓	R	**28.0**	36.3	41.8	48.9	57.4
Meta-DETR *w/o* CAM	✓	I	27.2	**42.1**	**50.5**	**52.9**	**59.3**

"R" denotes region-level detection. "I" denotes image-level detection.

TABLE 5
Ablation study on the impact of Correlational Aggregation Module

Detection Framework	R/I	Correlational Aggr. Module (CAM)	C	Novel mAP@0.5 1	2	3	5	10
Meta-DETR	I		1	27.2	42.1	50.5	52.9	59.3
		✓	1	30.3	44.0	52.1	55.7	**62.0**
		✓	5	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**
FsDetView [4]	R		1	24.2	35.3	42.2	49.1	57.4
		✓	5	**30.1**	**41.1**	**45.2**	**51.4**	**57.5**

"R" denotes region-level detection. "I" denotes image-level detection.
"C" denotes the number of support classes to aggregate simultaneously, which can only be 1 without the proposed Correlational Aggregation Module (CAM).

5.4 Ablation Studies

We conduct comprehensive ablation studies to verify the effectiveness of our design choices. Experimental results are averaged over 10 runs with different randomly sampled support datasets on the first class split of Pascal VOC.

Region-Level Detection vs. Image-Level Detection. From Table 1 and Table 3, we can find that fine-tuning Deformable DETR (Deformable-DETR+ft-full) generally outperforms fine-tuning Faster R-CNN (FRCN+ft-full), especially in the MS COCO dataset, where it is much harder to obtain accurate region proposals for novel classes due to higher complexity (see Fig. 2(a)). This observation aligns well with our insight that region-based detection frameworks tend to suffer from inaccurate regional proposals for novel classes. To further verify the superiority of image-level few-shot object detection, we adopt FsDetView [4], a state-of-the-art meta-learning-based few-shot detector built on top of Faster R-CNN, as a solid baseline to compare with our method. For a fair comparison, we add deformable transformers to FsDetView (denoted as FsDetView + Deform. Trans.) to rule out the performance difference brought by the transformer architecture. Furthermore, we replace our proposed CAM in Meta-DETR with the feature aggregation module proposed in FsDetView (denoted as Meta-DETR *w/o* CAM). As shown in Table 4, even with aligned network architecture and aggregation scheme, Meta-DETR *w/o* CAM still outperforms FsDetView + Deform. Trans. under most setups. The results validate the superiority of solving few-shot object detection at image level.

Impact of Correlational Aggregation Module (CAM). As shown in Table 5, when incorporating CAM into our model, even if we keep the number of support classes for simultaneous aggregation (C) as 1, CAM can still boost few-shot detection performance under all settings. This demonstrates CAM's strong capacity in aggregating query and support information even without the leverage of inter-class

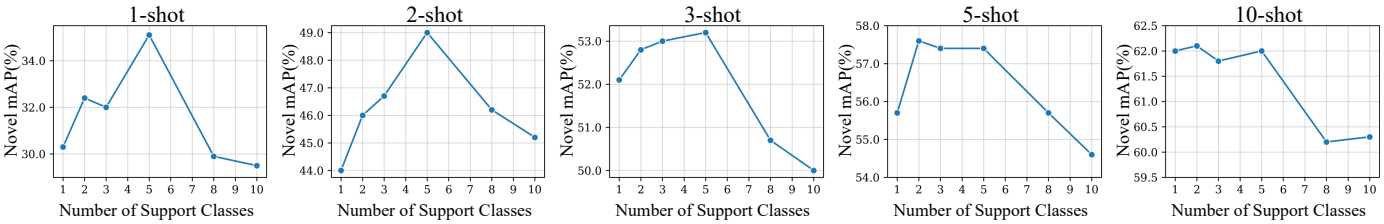


Fig. 5. Ablation study on the number of support classes for simultaneous correlational aggregation under different few-shot setups. Results are averaged over 10 repeated runs on Pascal VOC class split 1.

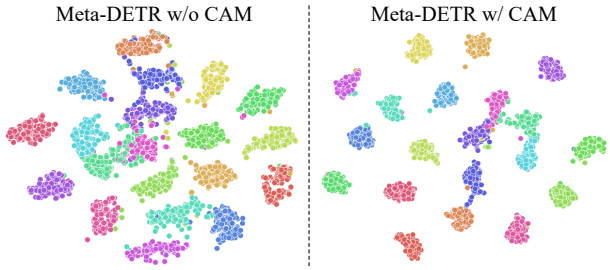


Fig. 6. t-SNE visualization of objects learned in the feature space with and without our designed Correlational Aggregation Module. Results are obtained on Pascal VOC class split 1 under the 2-shot setup.

TABLE 6
Confusion matrices of similar class pairs predicted with and without the proposed Correlational Aggregation Module

	GT \ Pred	missed	mbike	bike
Meta-DETR w/o CAM	mbike	89	247	33
	bike	63	10	316
Meta-DETR w/ CAM	mbike	67	286	16
	bike	58	7	324

	GT \ Pred	missed	cow	horse
Meta-DETR w/o CAM	cow	82	218	29
	horse	36	32	327
Meta-DETR w/ CAM	cow	46	273	10
	horse	25	23	347

Results obtained on Pascal VOC class split 1 under the 2-shot setup. "GT" denotes ground truth label; "Pred" denotes predicted label.

correlation. When multiple support classes are available ($C \geq 2$), CAM can further exploit their inter-class correlation to boost few-shot detection performance under lower-shot (≤ 5) settings, especially under 1-shot (+4.8% mAP) and 2-shot (+5.0% mAP), which shows the benefit of inter-class correlational meta-learning. No clear performance gain is observed for 10-shot, which implies that, when more training samples are available, the detector can already recognize novel classes and differentiate them from similar classes without explicitly modeling the inter-class correlation. We also apply our designed CAM to the commonly used region-based meta-detector FsDetView [4] and report the results in Table 5. Its steady performance gain demonstrates that CAM and the proposed inter-class correaltional meta-learning strategy can also benefit region-level few-shot object detection.

To understand how CAM functions to improve detection accuracy, we visualize the objects from different classes in the feature space learned with and without the proposed CAM with t-SNE [41]. As shown in Fig. 6, with CAM

TABLE 7
Ablation study on the design choices of the attention mechanism in the proposed Correlational Aggregation Module

(a) Apply Sigmoid	*(b)* Query Multiplication	*(c)* Modeling Background	Novel mAP@0.5 1	2	3	5	10
			29.8	44.8	51.2	54.8	59.6
		✓	31.2	46.1	52.5	56.2	61.5
✓	✓		32.6	45.6	51.3	56.1	60.9
✓	✓	✓	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**

included to perform inter-class correlational meta-learning, object classes are better separated from each other, which affirms our motivation of leveraging inter-class correlation to reduce misclassification among similar classes. To further verify our claim that CAM effectively reduces misclassification among similar classes, we select two pairs of similar classes (*motorbike vs. bike* and *cow vs. horse*) and plot their confusion matrices in Table 6. We can observe that CAM indeed reduces the misclassification by large margins with the exploitation of inter-class correlation. We also observe fewer missed predictions, which shows that the effective leverage of inter-class correlations also facilitates generalization to detect previously missed cases.

Number of Classes for Correlational Aggregation. Meta-DETR receives a fixed number of support classes (C) and simultaneously aggregates them with query features to capture the inter-class correlation among different support classes. With $C \geq 2$, Meta-DETR exploits the inter-class correlation among different classes. Fig. 5 investigates the impact of the number of support classes for aggregation. As the number of support classes C increases from 1 to 10, the lower-shot (≤ 5) detection performance first improves and then drops, while 10-shot performance first saturates and then drops. This validates the effectiveness of leveraging inter-class correlation under lower-shot (≤ 5) settings. The performance gain is considerable under extremely low-shots like 1-shot and 2-shot, indicating that it is highly beneficial to explore inter-class correlation when training samples are too scarce to model a novel class and differentiate it with other classes. We conjecture that the performance drop with a large number of support classes (≥ 8) for correlational aggregation is due to the model's limited capacity to differentiate too many support classes at one go. Based on the results, we set our method's number of support classes C as 5 in all other experiments unless otherwise stated.

Design Choices for Correlational Aggregation Module (CAM). The proposed CAM's attention mechanism differs from the original DETR attention in three aspects: *(a)* applying a sigmoid function to attention's *Value* in feature

Fig. 7. Visualization of Meta-DETR's 10-shot object detection results on various data setups. For simplicity, only detections of novel-class objects are illustrated. The qualitative experimental results show that Meta-DETR can detect novel objects effectively with very constrained training samples.

TABLE 8
Ablation study on early aggregation *vs.* late aggregation

CAM's Location @ Encoder Layers	Novel mAP@0.5				
	1	2	3	5	10
1	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**
3	27.1	42.9	50.6	54.0	59.2
6	15.2	31.5	37.7	50.3	53.4

matching, *(b)* multiplying attention's output with attention's *Query* in feature matching, and *(c)* explicitly modelling a prototype for the 'background' class. Among them, *(a)* and *(b)* are designed as a whole with *(a)* for generating 'filters' to remove query features that are irrelevant to the given support classes and *(b)* for applying the learned 'filters' to the query image features. And *(c)* enables Meta-DETR to better handle the 'no match' scenario where the query features do not match any of the support classes. We present ablation experiments in Table 7 that verify the effectiveness of the above three modifications.

Early Aggregation vs. Late Aggregation. The proposed CAM replaces one encoder layer in the transformer. As shown in Fig. 3, we place CAM ahead of the transformer encoder (as the first layer of the encoder). Table 8 studies the impact of the location of CAM in the transformer encoder. As shown, it is preferable to place CAM at the beginning stage of the transformer encoder for early aggregation, which also suggests the importance of learning a deep class-agnostic predictor.

5.5 Qualitative Results

In Fig. 7, we provide qualitative visualization of Meta-DETR's 10-shot object detection results on several sample

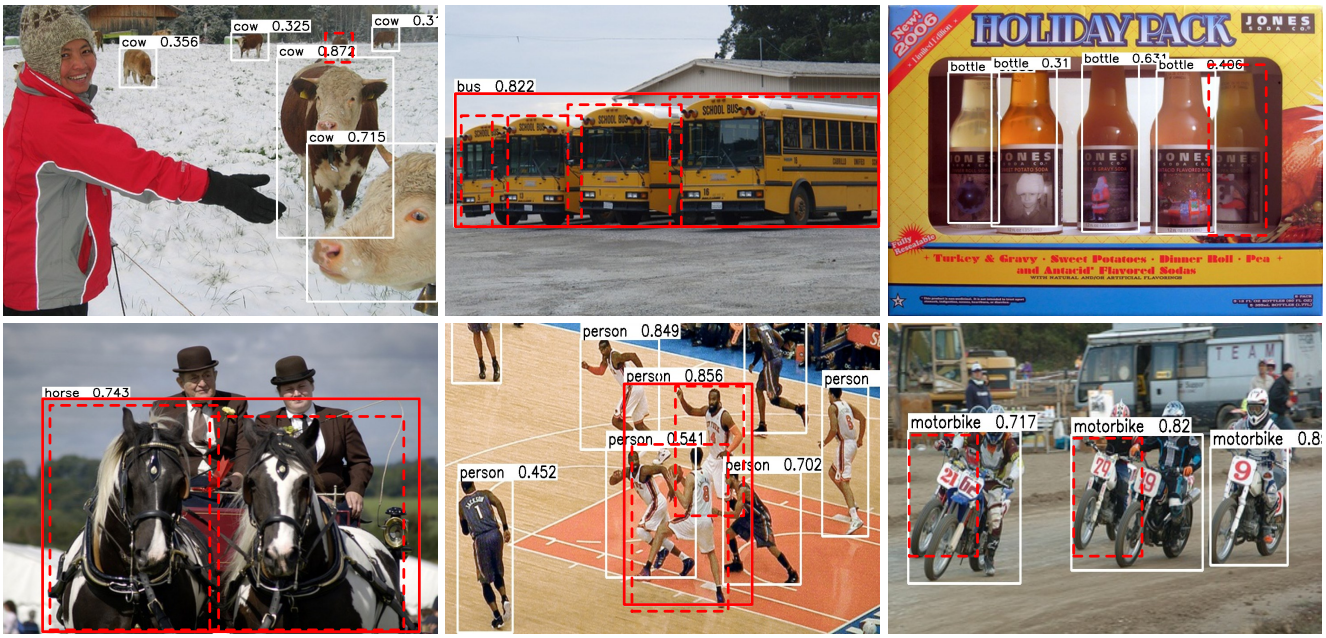


Fig. 8. Visualization of some failure cases of Meta-DETR's 10-shot object detection results. For simplicity, only detections of novel-class objects are illustrated. White boxes indicate true positives. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.

TABLE 9
Few-shot object detection and instance segmentation performance on COCO for novel classes

Shot	Method	Box						Mask					
		$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
5	Mask-RCNN+ft-full [28]	1.3	3.0	1.1	0.3	1.1	2.4	1.3	2.7	1.1	0.3	0.6	2.2
	Meta R-CNN [3]	3.5	9.9	1.2	1.2	3.9	5.8	2.8	6.9	1.7	0.3	2.3	4.7
	Meta-DETR (Ours)	**15.3**	**24.9**	**15.4**	**1.5**	**12.8**	**26.0**	**8.1**	**16.8**	**7.1**	**0.9**	**5.6**	**13.7**
10	Mask-RCNN+ft-full [28]	2.5	5.7	1.9	2.0	2.7	3.9	1.9	4.7	1.3	0.2	1.4	3.2
	Meta R-CNN [3]	5.6	14.2	3.0	2.0	6.6	8.8	4.4	10.6	3.3	0.5	3.6	7.2
	Meta-DETR (Ours)	**19.8**	**31.3**	**20.4**	**4.5**	**17.4**	**30.5**	**10.1**	**20.8**	**8.7**	**1.7**	**7.6**	**15.8**

images from their respective data setups. Note that we show the detection of novel classes only since the focus of few-shot object detection is to detect objects of novel classes. We show detection results with confidence scores higher than 0.25 to filter out low-confidence predictions. It can be observed that the proposed Meta-DETR is able to detect novel objects effectively even with very limited training samples.

5.6 Failure Cases and Future Directions

Fig. 8 illustrates typical failure cases of the proposed Meta-DETR. The most typical failure cases happen while multiple instances of novel objects are heavily clustered, largely due to the lack of supervision in such cases and the lack of a mechanism to discriminate objects' boundaries. Other typical failure cases include difficulty in detecting small objects as well as false negatives with less salient objects, which are also applicable in general object detectors.

Although the current few-shot object detection performance is still far from perfect, our proposed Meta-DETR establishes a new few-shot object detection paradigm that is conceptually simple with room for improvement. In our future work, we will investigate new mechanisms that can highlight object boundaries and thus help avoid some failure case as illustrated in Fig. 8. Besides, since we only explore single-scale features throughout all experiments, an interesting and promising direction is to exploit multi-scale features in meta-learning-based few-shot object detection. By properly designing a dual-scale-selection strategy for both query and support, we expect it can further improve the performance of few-shot object detection, especially on small objects.

5.7 Extension to Few-Shot Instance Segmentation

The proposed Meta-DETR adopts a meta-learning framework which is generic and can be adapted to other downstream vision tasks beyond object detection. We validate this feature by examining how it can be extended to perform instance segmentation with simple modifications.

As described in [16], the original DETR can be extended to perform instance segmentation by adding a mask head on top of the decoder outputs. We similarly introduce an additional mask head over Meta-DETR to predict objects' masks for few-shot instance segmentation. The additional mask head takes the output of the transformer decoder and encoded image features as input and predicts a binary mask for each object query. It also follows the designed inter-class correlational meta-learning strategy for better generalization. To train Meta-DETR to perform few-shot instance segmentation, we first train it on the previously

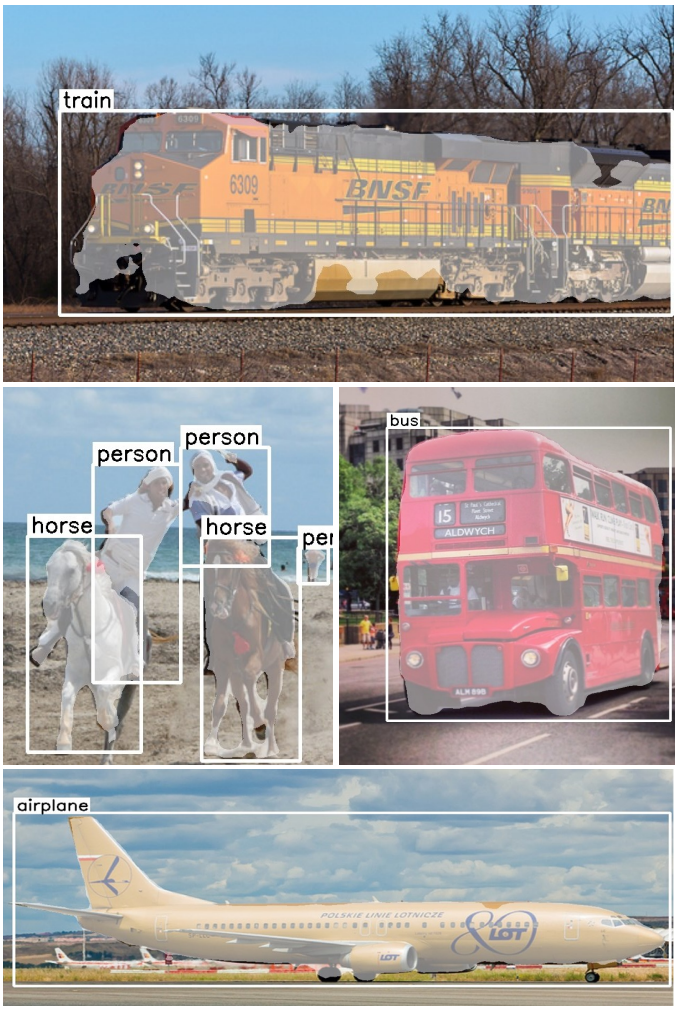


Fig. 9. Visualization of Meta-DETR's 10-shot instance segmentation results on MS COCO. For simplicity, only segmentations of novel-class objects are illustrated.

mentioned few-shot object detection tasks, and then freeze all the weights and train only the additional mask head for instance segmentation.

Experimental Results. We conduct experiments for few-shot instance segmentation on MS COCO under 5-shot and 10-shot setups. Similarly, the 20 classes shared with Pascal VOC are chosen as novel classes, and the remaining 60 classes are set as base classes. Note that AP for instance segmentation is evaluated with mask IoU. As shown in Table 9, Meta-DETR outperforms compared methods by large margins. The results demonstrate the superiority and universality of our Meta-DETR, which can extend to other instance-level few-shot learning tasks. Note that the compared Meta R-CNN [3] adopts region-level prediction together with the conventional class-by-class meta-learning via feature reweighting. The comparison between Meta R-CNN [3] and our proposed Meta-DETR verifies that the combination of the image-level prediction and the exploitation of inter-class correlation via correlational meta-learning can effectively benefit other instance-level few-shot learning tasks like few-shot instance segmentation. We also provide qualitative results for instance segmentation in Fig. 9.

6 Conclusion

This paper presents a new few-shot object detection framework, namely Meta-DETR. The proposed framework achieves *(i)* pure image-level prediction, which lifts the constraints caused by novel classes' inaccurate region proposals, and *(ii)* effective exploitation of categorical correlation via a inter-class correlational meta-learning strategy, which reduces misclassification and enhances generalization among similar or related classes. Despite its simplicity, our method achieves state-of-the-art performance over multiple few-shot object detection setups, outperforming prior works by large margins. It can also be easily extended to other instance-level few-shot learning tasks. We hope this work can offer good insights and inspire further researches in few-shot object detection and other related topics.

Acknowledgment

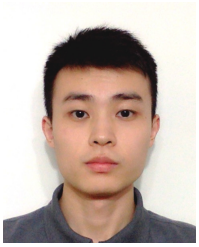
This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG94/20).

References

[1] B. Landau, L. Smith, and S. Jones, "The importance of shape in early lexical learning," *Cognitive Development*, vol. 3, pp. 299–321, 1988.
[2] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *ICCV*, 2019.
[3] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta R-CNN: Towards general solver for instance-level low-shot learning," in *ICCV*, 2019.
[4] Y. Xiao and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," in *ECCV*, 2020.
[5] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *CVPR*, 2020.
[6] H. Hu, S. Bai, A. Li, J. Cui, and L. Wang, "Dense relation distillation with context-aware aggregation for few-shot object detection," in *CVPR*, 2021.
[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.
[8] W. Zhang, Y.-X. Wang, and D. A. Forsyth, "Cooperating RPN's improve few-shot object detection," *arXiv preprint arXiv:2011.10142*, 2020.
[9] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, pp. 261–318, 2020.
[10] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *CVPR*, 2018.
[11] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018.
[12] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 015–10 024, 2019.
[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.
[14] J. Redmon and A. Farhadi, "YOLO 9000: Better, faster, stronger," in *CVPR*, 2017.
[15] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *CVPR*, 2018.
[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.
[17] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *ICLR*, 2021.

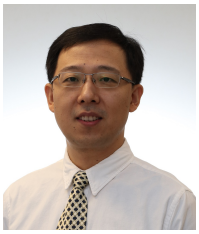
[18] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *CVPR*, 2021.
[19] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," in *ICCV*, 2021.
[20] G. Zhang, Z. Luo, Y. Yu, K. Cui, and S. Lu, "Accelerating DETR convergence via semantic-aligned matching," in *CVPR*, 2022.
[21] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," in *AAAI*, 2018.
[22] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," in *ICML*, 2020.
[23] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *ECCV*, 2020.
[24] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *CVPR*, 2021.
[25] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *ICCV*, 2019.
[26] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *CVPR*, 2020.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.
[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.
[30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *CVPR*, 2019.
[31] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.
[32] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. Feris, A. Kumar, R. Giries, and A. M. Bronstein, "RepMet: Representative-based metric learning for classification and one-shot object detection," in *CVPR*, 2019.
[33] W. Zhang and Y.-X. Wang, "Hallucination improves few-shot object detection," in *CVPR*, 2021.
[34] Z. Fan, Y. Ma, Z. Li, and J. Sun, "Generalized few-shot object detection without forgetting," in *CVPR*, 2021.
[35] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye, "Beyond max-margin: Class margin equilibrium for few-shot object detection," in *CVPR*, 2021.
[36] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *CVPR*, 2021.
[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[38] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.
[41] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.

Gongjie Zhang is currently working toward the Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore, under the supervision of Dr. Shijian Lu. He received his B.Eng. degree in electronic and information engineering in 2018 from Northeastern University, Shenyang, China. He has published multiple journal and conference papers in the field of computer vision. He has also served as reviewer for several top journals and conferences such as T-PAMI, T-IP, CVPR, ICCV, and ACM MM. His research interests mainly include computer vision, object detection, few-shot learning, and meta-learning.

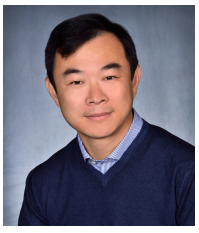


Zhipeng Luo is currently a Ph.D. student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, under the supervision of Dr. Shijian Lu. He received his B.Eng. degree in mechanical engineering in 2015 and M.Sc. degree in computing in 2018 from National University of Singapore. He has published multiple top conference papers in the field of computer vision. His research interests include computer vision, object detection, and object tracking.

Kaiwen Cui is currently working toward the Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore, under the supervision of Dr. Shijian Lu. He received his B.Eng. degree in 2016 and M.Sc. degree in 2017, both in electrical and electronic engineering from National University of Singapore. He has published multiple top conference papers in the field of computer vision. His research interests mainly include computer vision and data-limited image generation.



Shijian Lu received his Ph.D. in electrical and computer engineering from National University of Singapore. He is an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His major research interests include image and video analytics, visual intelligence, and machine learning. He has published more than 100 international refereed journal and conference papers and co-authored over 10 patents in these research areas. He is currently an Associate Editor for the journal Pattern Recognition (PR). He has also served in the program committee of a number of conferences, *e.g.*, the Area Chair of the International Conference on Document Analysis and Recognition (ICDAR) 2017 and 2019, the Senior Program Committee of the International Joint Conferences on Artificial Intelligence (IJCAI) 2018 and 2019, *etc.*



Eric Xing (Fellow, IEEE) received the Ph.D. degree in molecular biology from Rutgers University, New Brunswick, NJ, USA, in 1999, and the Ph.D. degree in computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2004. He is currently a Professor of machine learning with the School of Computer Science and the Director of the CMU Center for Machine Learning and Health, Carnegie Mellon University, Pittsburgh, PA, USA. His principal research interests lie in the development of machine learning and statistical methodology, especially for solving problems involving automated learning, reasoning, and decision-making in high-dimensional, multimodal, and dynamic possible worlds in social and biological systems. Dr. Xing is a member of the DARPA Information Science and Technology (ISAT) Advisory Group and the Program Chair of the International Conference on Machine Learning (ICML) 2014. He is also an Associate Editor of The Annals of Applied Statistics (AOAS), the Journal of American Statistical Association (JASA), the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), and PLOS Computational Biology and an Action Editor of the Machine Learning Journal (MLJ) and the Journal of Machine Learning Research (JMLR).