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

Gongjie Zhang[†]   **Zhipeng Luo**[†]   **Kaiwen Cui**   **Shijian Lu**[∗]

Nanyang Technological University, Singapore

gongjiezhang@ntu.edu.sg   zhipeng001@e.ntu.edu.sg   kaiwen001@e.ntu.edu.sg   shijian.lu@ntu.edu.sg

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 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, a novel few-shot detection framework that incorporates correlational aggregation for meta-learning into DETR detection frameworks. Meta-DETR works entirely at image level without any region proposals, which circumvents the constraint of inaccurate proposals in prevalent few-shot detection frameworks. Besides, Meta-DETR can simultaneously attend to multiple support classes within a single feed-forward. This unique design allows capturing the inter-class correlation among different classes, which significantly reduces the misclassification of similar classes and enhances 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 will be released.

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 very few instructions (Landau, Smith, and Jones 1988). 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 extremely limited supervision from annotated samples, the key is to exploit knowledge from base classes and generalize it to novel classes. To this

---

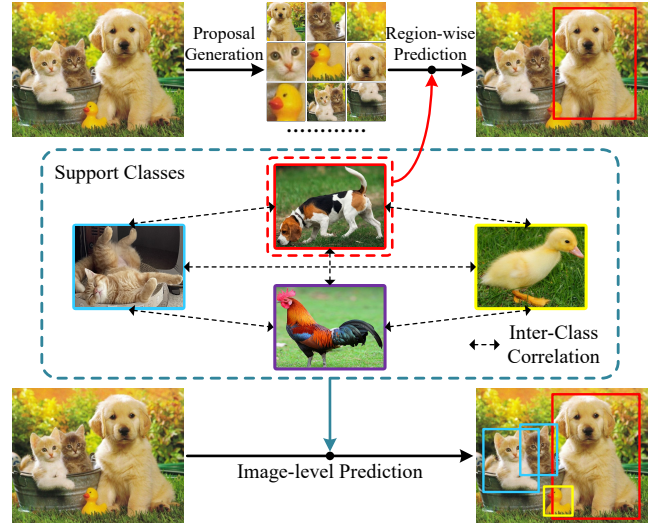† denotes equal contribution.

∗ denotes corresponding author.

Figure 1: Comparison of few-shot object detection pipelines: Prior works (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.

end, many works (Kang et al. 2019; Yan et al. 2019; Xiao and Marlet 2020; Fan et al. 2020; Hu et al. 2021) incorporate meta-learning into generic object detection frameworks, mostly Faster R-CNN (Ren et al. 2015), 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. Unfortunately, as investigated by Fan et al. (2020) and Zhang, Wang, and Forsyth (2020), it is not easy to produce high-quality region proposals for novel classes with limited supervision under the few-shot detection setup. Such a gap

(a) Quality Gap in Region Proposals   **(b)** Cosine Similarity of Class Prototypes   **(c)** Misclassification of Related Classes
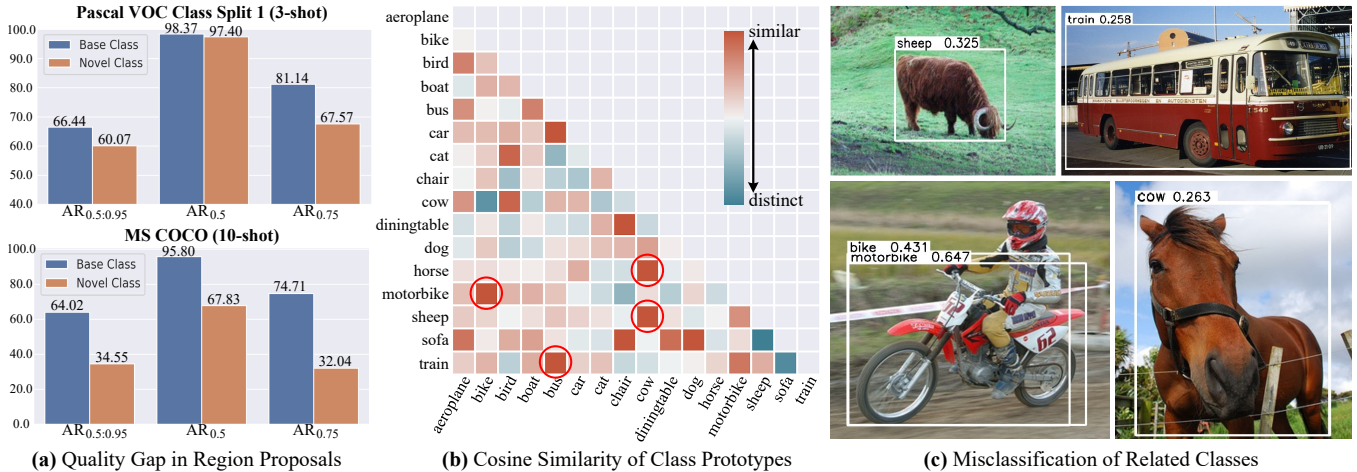
Figure 2: Existing few-shot detection frameworks tend to suffer from inaccurate region proposals and under-exploitation of inter-class correlation. Due to very limited training samples, the proposal quality (measured by Average Recall on top 1000 proposals) for novel classes is clearly lower than that of base classes, as shown in (a). This hinders the knowledge generalization to novel classes. Additionally, object classes with similar appearances are highly correlated in feature space as shown in (b), which tend to be misclassified if the learning does not incorporate the correlation among them, as illustrated in (c).

in the quality of region proposals obstructs the generalization from base classes to novel classes. *Second*, most existing meta-learning-based approaches (Kang et al. 2019; Yan et al. 2019; Fan et al. 2020; Xiao and Marlet 2020) 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 feed-forward, 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 sheep) 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 achieves meta-learning at image level and at the same time explicitly exploits the inter-class correlation among different support classes. To our best knowledge, this is the first work that explores incorporating meta-learning into the recently proposed DETR detection frameworks (Carion et al. 2020; Zhu et al. 2021b), which can skip proposal generation and directly perform detection at image level. With image-level meta-learning, the proposed Meta-DETR lifts the constraint of inaccurate region proposals as in prevalent few-shot detection frameworks. In addition, as shown in Fig. 1, Meta-DETR can 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 *(i)* the inter-class commonality to facilitate generalization among related classes and *(ii)* the inter-class uniqueness to reduce misclassification among similar classes.

In summary, the contributions of this work are threefold. *First*, we propose Meta-DETR, an innovative few-shot object detection framework that incorporates meta-learning into DETR detection frameworks. Being the first pure image-level meta-detector, Meta-DETR circumvents the gap of inaccurate region proposals for novel-class objects, enabling better generalization to novel classes. *Second*, we design a novel correlational aggregation module for few-shot object detection, which allows aggregating query features with multiple support classes simultaneously. It enables effective exploitation of the inter-class correlation, which greatly reduces misclassification and enhances model generalization. *Third*, extensive experiments show that, without bells and whistles, the proposed Meta-DETR outperforms state-of-the-art methods by large margins.

2   Related Work

Object Detection   Generic object detection (Liu et al. 2020) 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 (Ren et al. 2015) and its variants (Hu et al. 2018; Cai and Vasconcelos 2018; Zhang, Lu, and Zhang 2019), 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 (Liu et al. 2016; Redmon and Farhadi 2017; Zhang et al. 2018) employ densely placed anchors as region proposals and directly make predictions over them. Recently, another line of research featuring DETR (Carion et al. 2020) and its variants (Zhu et al. 2021b; Dai et al. 2021) 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.
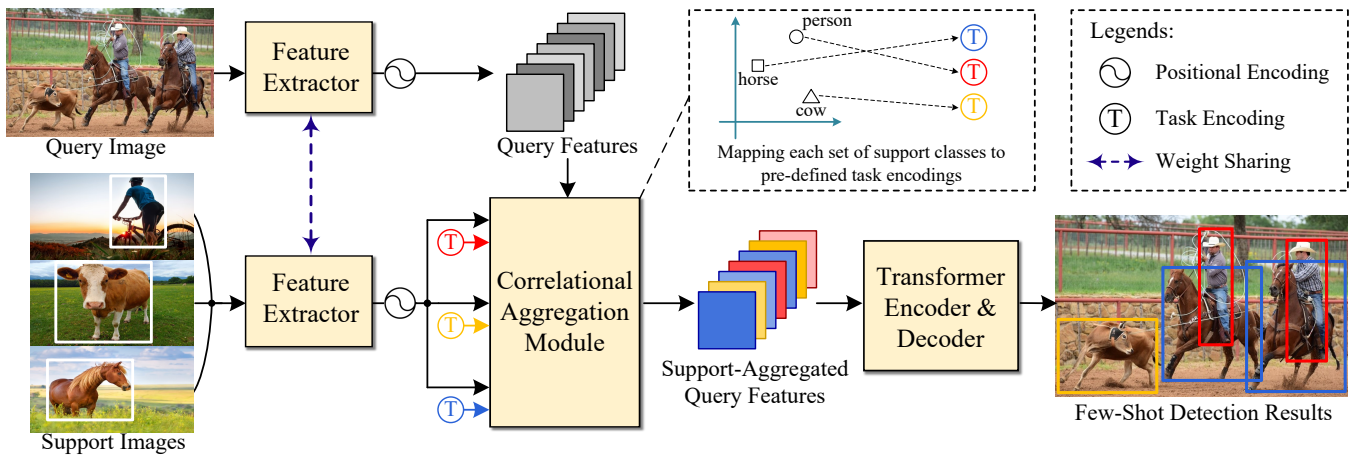
Figure 3: The framework of the proposed Meta-DETR. Query and support images are processed by a weight-shared feature extractor to produce query and support features. To leverage the inter-class correlation in meta-learning, the correlational aggregation module (CAM) first matches the query features with multiple support classes simultaneously, then introduces task encodings to differentiate these support classes. Finally, few-shot detection results are obtained via a class-agnostic transformer architecture that predicts objects' locations and corresponding task encodings.

Few-Shot Object Detection Existing works on few-shot object detection can be categorized into two paradigms: transfer learning and meta-learning. Transfer-learning-based methods include LSTD (Chen et al. 2018), TFA (Wang et al. 2020), MPSR (Wu et al. 2020), and FSCE (Sun et al. 2021), where novel concepts are learned via fine-tuning. Differently, meta-learning-based methods (Kang et al. 2019; Yan et al. 2019; Wang, Ramanan, and Hebert 2019; Perez-Rua et al. 2020; Xiao and Marlet 2020; Fan et al. 2020; Hu et al. 2021) 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 meta-learning 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 meta-learning-based few-shot object detection frameworks.

3 Preliminaries

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 $\mathcal{D}_{\text{novel}}$.

Rethink Region-Based Detection Frameworks Most existing works on few-shot object detection are developed on top of Faster R-CNN (Ren et al. 2015), a region-based object detector, thanks to its robust performance and easy 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 setup. 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 fully exploiting base-class knowledge to generalize to novel classes. Though several studies (Fan et al. 2020; Zhang, Wang, and Forsyth 2020) 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 setup.

Rethink Meta-Learning via Feature Reweighting To meta-learn a class-agnostic detector that can generalize across various classes, most existing methods (Kang et al. 2019; Yan et al. 2019; Fan et al. 2020; Xiao and Marlet 2020) 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 aggregation approaches can deal with only one support class within each feed-forward process, *i.e.*, C repeated runs are required to detect C 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), in existing methods, we observe that objects misclassified as highly correlated classes constitute a major source of error due to the negligence of inter-class correlation.

4 Meta-DETR

4.1 Model Overview

Fig. 3 presents the architecture of the proposed Meta-DETR. Motivated by previous discussions, Meta-DETR employs

the recently proposed Deformable DETR (Zhu et al. 2021b), a fully end-to-end Transformer-based (Vaswani et al. 2017) detector, as the basic detection framework to bypass the constraint of region-wise prediction. Besides, during meta-learning, Meta-DETR aggregates query features with multiple support classes simultaneously, thus can exploit the inter-class correlation among different classes to reduce misclassification and boost generalization.

Specifically, 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, the correlational aggregation module (CAM), which will be introduced in detail later, performs matching between the query features and the set of support classes. CAM further maps the set of support classes to a set of pre-defined task encodings that differentiate these support classes in a class-agnostic manner. Finally, detection results are obtained via a transformer architecture that predicts objects' locations and corresponding task encodings. As the detection targets are dynamically determined by support classes and their mappings to task encodings, the proposed Meta-DETR is trained as a meta-learner to extract generalizable knowledge not specific to certain classes.

4.2 Correlational Aggregation Module

The correlational aggregation module (CAM) is the key component in Meta-DETR, 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, given the query and support features, a weight-shared multi-head attention module first encodes them into the same feature space, and the prototype for each support class is obtained by applying RoIAlign (He et al. 2017) followed by average pooling on the support features. CAM then performs feature matching and encoding matching, which will be elaborated in the remainder of this subsection, to match the query features with support features and task encodings, respectively. Their results are summed together and processed by a feed-forward network (FFN) to produce the final output.

Feature Matching Feature matching is accomplished by a single-head attention mechanism. Specifically, given a query feature map $\mathbf{Q} \in \mathbb{R}^{HW \times d}$ and the support class prototypes $\mathbf{S} \in \mathbb{R}^{C \times d}$, the matching coefficients are obtained via:

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

where HW is the spatial size, C is the number of support classes, d is the feature dimensionality, and $\mathbf{W}$ is a linear projection shared by $\mathbf{Q}$ and $\mathbf{S}$, which ensures they are embedded into the same feature space. 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
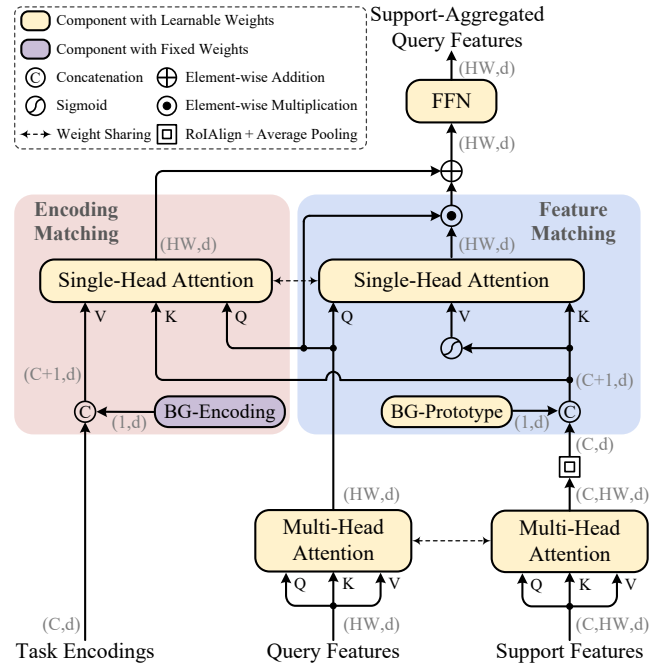

Figure 4: The architecture of the Correlational Aggregation Module (CAM). It performs two matching processes: feature matching filters out query features that are unrelated to support classes, while encoding matching maps support classes to a set of pre-defined task encodings that differentiate the support classes in a class-agnostic manner.

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 filter out features not matched to any support class, producing a feature map $\mathbf{Q_F}$ that highlights objects belonging to the given support classes.

Encoding Matching To achieve meta-learning that requires class-agnostic prediction, we introduce a set of pre-defined task encodings and map the given support classes to these 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 (Vaswani et al. 2017). Encoding matching uses the same matching coefficients as feature matching, and the matched encodings $\mathbf{Q_E}$ are obtained via:

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

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 space 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.

Table 1:

Method \ Shot	Class Split 1					Class Split 2					Class Split 3					Avg.
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10	
Results over a single run:																
LSTD (Chen et al. 2018)	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 (Schwartz et al. 2019) ‡	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 (Kang et al. 2019)	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 (Wang, Ramanan, and Hebert 2019)	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 (Yan et al. 2019)	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/ cos (Wang et al. 2020) ‡	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 (Wu et al. 2020) ‡	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 (Zhang and Wang 2021) ‡	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
CME (Li et al. 2021) ‡	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 (Zhu et al. 2021a) ‡ ⊎	**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 (Sun et al. 2021) ‡	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 (Ren et al. 2015) ‡	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 (Zhu et al. 2021b) ‡	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/ cos (Wang et al. 2020) ‡	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 (Xiao and Marlet 2020)	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 (Wu et al. 2020) ‡ △	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 (Hu et al. 2021) ‡	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 (Sun et al. 2021) ‡	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**

Table 1: Few-shot detection performance (mAP@0.5) on Pascal VOC *test 07* for novel classes. ‡ indicates methods using multi-scale features. △ indicates re-evaluated results using official codes. ⊎ indicates usage of external data.

4.3 Training Objective

Target Generation The detection targets of Meta-DETR are dynamically determined by the support classes and their mappings to task encodings. Concretely, given a query image, C support images representing different support classes are randomly sampled. Only ground truth objects belonging to the sampled support classes are kept as detection targets. Besides, the classification target for each object is the task encoding of the ground truth class instead of the ground truth class itself. We empirically set C as 5 according to our ablation study on this hyper-parameter in Fig. 5.

Loss Function The loss functions for our proposed Meta-DETR follow Deformable DETR (Zhu et al. 2021b), which adopts a set-based Hungarian loss that forces unique predictions for each object via bipartite matching. Following Meta R-CNN (Yan et al. 2019), we additionally introduce a cosine similarity cross-entropy loss (Chen et al. 2019) to classify the class prototypes obtained by our designed CAM. It encourages prototypes of different classes to be distinguished from each other. Please refer to appendix for a detailed description of the loss functions.

4.4 Training and Inference Procedure

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 (Yan et al. 2019; Wang et al. 2020; Xiao and Marlet 2020), we also include objects from base classes to prevent performance drop for base classes. In both stages, the network is optimized in an end-to-end manner with the same training objective described in Section 4.3.

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

5 Experiments

5.1 Datasets

We follow the well-established data setups for few-shot object detection (Kang et al. 2019; Wang et al. 2020). Concretely, two widely used few-shot object detection benchmarks are adopted in our experiments.

Pascal VOC (Everingham et al. 2010) 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 (mAP) at IoU threshold 0.5 is used as the evaluation metric. Results are averaged over 10 randomly sampled support datasets.

MS COCO (Lin et al. 2014) 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 (He et al. 2016) as the feature extractor. The network architectures and hyperparameters remain the same as Deformable DETR (Zhu et al. 2021b). We implement our model in single-scale version for fair comparison with other works. We also follow FsDetView (Xiao and Marlet 2020) to implement the aggregation with a slightly more complex scheme compared with solely feature reweighting. Following Deformable DETR, we train our model with 8 x Nvidia V100 GPUs, using the AdamW (Loshchilov and Hutter 2019) optimizer with an initial learning rate of $2{\times}10^{-4}$ and a weight decay of $1{\times}10^{-4}$. Batch size is set to 32. In the base training stage, we train the model for 50 epochs for both Pascal VOC and MS COCO. Learning rate is decayed at the 45th 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

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, our method achieves the best average performance across all setups, with a large margin of $+4.6\%$ mAP compared with the second-best. The strong performance demonstrates the superiority and robustness of our proposed method.

MS COCO Table 2 shows the 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 potentially attributed to 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 our method can effectively lift the constraint of inaccurate region proposals, thus producing more accurate detection results.

5.4 Ablation Studies

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

Region-Level vs. Image-Level From Table 1 and Table 2, 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 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 (Xiao and Marlet 2020), a state-of-the-art meta-learning-based few-shot detector built on top of Faster

Shot	Method	AP	$AP_{0.5}$	$AP_{0.75}$
1	FRCN-ft-full (Ren et al. 2015) ‡ §	1.7	3.3	1.6
	Deformable-DETR-ft-full (Zhu et al. 2021b) §	1.8	3.1	1.8
	TFA w/ cos (Wang et al. 2020) ‡ §	1.9	3.8	1.7
	TFA w/ cos + Halluc (Zhang and Wang 2021) ‡	3.8	6.5	4.3
	Meta-DETR (Ours) §	**7.5**	**12.5**	**7.7**
3	FRCN-ft-full (Ren et al. 2015) ‡ §	3.7	7.1	3.5
	Deformable-DETR-ft-full (Zhu et al. 2021b) §	4.9	7.8	5.1
	TFA w/ cos (Wang et al. 2020) ‡ §	5.1	9.9	4.8
	TFA w/ cos + Halluc (Zhang and Wang 2021) ‡	6.9	12.6	7.0
	Meta-DETR (Ours) §	**13.5**	**21.7**	**14.0**
5	FRCN-ft-full (Ren et al. 2015) ‡ §	4.6	8.7	4.4
	Deformable-DETR-ft-full (Zhu et al. 2021b) §	7.4	12.3	7.7
	TFA w/ cos (Wang et al. 2020) ‡ §	7.0	13.3	6.5
	FsDetView (Xiao and Marlet 2020) §	10.7	24.5	6.7
	Meta-DETR (Ours) §	**15.4**	**25.0**	**15.8**
10	FRCN-ft-full (Ren et al. 2015) ‡ §	5.5	10.0	5.5
	Deformable-DETR-ft-full (Zhu et al. 2021b) §	11.7	19.6	12.1
	Meta-YOLO (Kang et al. 2019)	5.6	12.3	4.6
	Meta Det (Wang, Ramanan, and Hebert 2019)	7.1	14.6	6.1
	Meta R-CNN (Yan et al. 2019)	8.7	19.1	6.6
	TFA w/ cos (Wang et al. 2020) ‡ §	9.1	17.1	8.8
	FSOD (Fan et al. 2020)	12.0	22.4	11.8
	FsDetView (Xiao and Marlet 2020) §	12.5	27.3	9.8
	MPSR (Wu et al. 2020) ‡	9.8	17.9	9.7
	SRR-FSD (Zhu et al. 2021a) ‡	11.3	23.0	9.8
	CME (Li et al. 2021) ‡	15.1	24.6	16.4
	DCNet (Hu et al. 2021) ‡ §	12.8	23.4	11.2
	FSCE (Sun et al. 2021) ‡ §	11.1	-	9.8
	Meta-DETR (Ours) §	**19.0**	**30.5**	**19.7**
30	FRCN-ft-full (Ren et al. 2015) ‡ §	7.4	13.1	7.4
	Deformable-DETR-ft-full (Zhu et al. 2021b) §	16.3	27.2	16.7
	Meta-YOLO (Kang et al. 2019)	9.1	19.0	7.6
	Meta Det (Wang, Ramanan, and Hebert 2019)	11.3	21.7	8.1
	Meta R-CNN (Yan et al. 2019)	12.4	25.3	10.8
	TFA w/ cos (Wang et al. 2020) ‡ §	12.1	22.0	12.0
	FsDetView (Xiao and Marlet 2020) §	14.7	30.6	12.2
	MPSR (Wu et al. 2020) ‡	14.1	25.4	14.2
	SRR-FSD (Zhu et al. 2021a) ‡	14.7	29.2	13.5
	CME (Li et al. 2021) ‡	16.9	28.0	17.8
	DCNet (Hu et al. 2021) ‡ §	18.6	32.6	17.5
	FSCE (Sun et al. 2021) ‡ §	15.3	-	14.2
	Meta-DETR (Ours) §	**22.2**	**35.0**	**22.8**

Table 2: Few-shot detection performance on MS COCO *val 2017* for novel classes. ‡ indicates methods using multi-scale features. § indicates results averaged on multiple runs.

R-CNN, as a solid baseline to compare with our method. For a fair comparison, we add a deformable transformer to FsDetView 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 in FsDetView (denoted as Meta-DETR w/o CAM). As shown in Table 3, even with aligned network architecture and aggregation scheme, Meta-DETR w/o CAM still outperforms FsDetView + Deform Transformer 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 4, when incorporating CAM into our model, even if we keep the number of support classes as 1, which means CAM cannot explicitly leverage inter-class correlation among different support classes, CAM can still boost few-shot detection performance under all set-
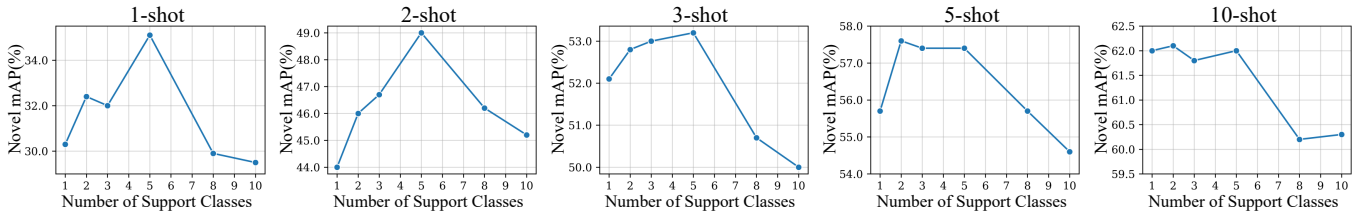
Figure 5: Ablation study over the number of support classes for correlational aggregation under different few-shot setups.

Method \ Shot	Novel mAP@0.5				
	1	2	3	5	10
FsDetView (Xiao and Marlet 2020)	24.2	35.3	42.2	49.1	57.4
FsDetView + Deform Transformer	**28.0**	36.3	41.8	48.9	57.4
Meta-DETR w/o CAM	27.2	**42.1**	**50.5**	**52.9**	**59.3**

Table 3: Performance comparison between region-level and image-level meta-learning-based few-shot object detection.

Method	CAM	Modeling Background	C	Novel mAP@0.5				
				1	2	3	5	10
Meta-DETR			1	27.2	42.1	50.5	52.9	59.3
	✓	✓	1	30.3	44.0	52.1	55.7	**62.0**
	✓		5	32.6	45.6	51.3	56.1	60.9
	✓	✓	5	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**

Table 4: Ablation study to evaluate the effectiveness of our designed CAM and its design choices. C denotes the number of support classes to aggregate simultaneously.

Method \ Shot	C	Novel mAP@0.5				
		1	2	3	5	10
FsDetView (Xiao and Marlet 2020)	1	24.2	35.3	42.2	49.1	57.4
FsDetView w/ CAM	5	**30.1**	**41.1**	**45.2**	**51.4**	**57.5**

Table 5: Ablation study on the effectiveness of our designed CAM on region-based detection frameworks. C denotes the number of support classes to aggregate simultaneously.

tings. This demonstrates CAM's strong capacity in aggregating query and support information. When multiple support classes are available, CAM can further enable the exploitation of 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). 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 their inter-class correlation. We also apply our designed CAM to the commonly used region-based meta-detector FsDetView and report the results in Table 5. Its steady performance gain demonstrates CAM's strong adaptability.

Fig. 6 further shows the visualization of objects of different classes in the feature space learned with and without the explicit exploitation of inter-class correlation. As shown, with CAM introduced to capture inter-class correlation, object classes are better separated from each other, which affirms our motivation of leveraging inter-class correlation to reduce misclassification among similar classes.
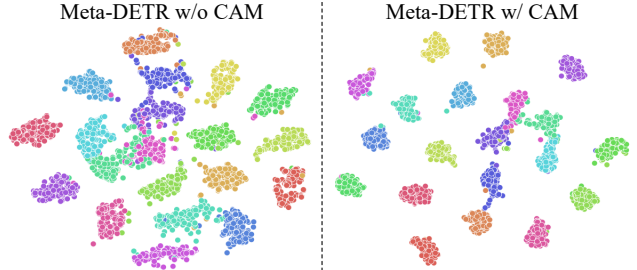

Figure 6: t-SNE visualization of objects learned in the feature space with and without our designed CAM. Results are obtained on split 1 of Pascal VOC under the 2-shot setup.

Number of Classes for Correlational Aggregation Meta-DETR receives a fixed number of support classes and simultaneously aggregates them with query features to capture the inter-class correlation among different support classes. Fig. 5 investigates the impact of the number of support classes to aggregate at a time. As the number of support classes 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 also validates the effectiveness of leveraging inter-class correlation under lower-shot settings. We conjecture the performance drop with a large number of support classes for correlational aggregation is due to the model's limited capacity to differentiate so many support classes at one go. Based on the results, we set our method's number of support classes as 5 in all other experiments unless otherwise stated.

Impact of Explicitly Modeling Background Table 4 also validates the effectiveness of explicitly modeling a prototype and a task encoding for background, which allows our method to better handle the 'no match' scenario where the query features do not match any of the support classes.

6   Conclusion

This paper presents a novel few-shot object detection framework, namely Meta-DETR. The proposed framework achieves *(i)* pure image-level meta-learning, which lifts the constraints caused by novel classes' inaccurate region proposals, and *(ii)* effective exploitation of inter-class correlation, 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. We hope this work can offer good insights and inspire further researches in few-shot object detection.

References

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving into high quality object detection. In *CVPR*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *ECCV*.

Chen, H.; Wang, Y.; Wang, G.; and Qiao, Y. 2018. LSTD: A low-shot transfer detector for object detection. In *AAAI*.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; and Huang, J.-B. 2019. A Closer Look at Few-shot Classification. In *ICLR*.

Dai, Z.; Cai, B.; Lin, Y.; and Chen, J. 2021. UP-DETR: Unsupervised Pre-training for Object Detection with Transformers. In *CVPR*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2): 303–338.

Fan, Q.; Zhuo, W.; Tang, C.-K.; and Tai, Y.-W. 2020. Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector. In *CVPR*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hu, H.; Bai, S.; Li, A.; Cui, J.; and Wang, L. 2021. Dense Relation Distillation with Context-aware Aggregation for Few-Shot Object Detection. In *CVPR*.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *CVPR*.

Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; and Darrell, T. 2019. Few-shot Object Detection via Feature Reweighting. In *ICCV*.

Landau, B.; Smith, L.; and Jones, S. 1988. The importance of shape in early lexical learning. *Cognitive Development*, 3: 299–321.

Li, B.; Yang, B.; Liu, C.; Liu, F.; Ji, R.; and Ye, Q. 2021. Beyond Max-Margin: Class Margin Equilibrium for Few-shot Object Detection. In *CVPR*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; and Pietikäinen, M. 2020. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, 128: 261–318.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. In *ECCV*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled weight decay regularization. In *ICLR*.

Perez-Rua, J.-M.; Zhu, X.; Hospedales, T. M.; and Xiang, T. 2020. Incremental Few-Shot Object Detection. In *CVPR*.

Redmon, J.; and Farhadi, A. 2017. YOLO 9000: Better, Faster, Stronger. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Pankanti, S.; Feris, R.; Kumar, A.; Giries, R.; and Bronstein, A. M. 2019. RepMet: Representative-based metric learning for classification and one-shot object detection. In *CVPR*.

Sun, B.; Li, B.; Cai, S.; Yuan, Y.; and Zhang, C. 2021. FSCE: Few-shot object detection via contrastive proposal encoding. In *CVPR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*.

Wang, X.; Huang, T. E.; Darrell, T.; Gonzalez, J. E.; and Yu, F. 2020. Frustratingly Simple Few-Shot Object Detection. In *ICML*.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2019. Meta-learning to detect rare objects. In *ICCV*.

Wu, J.; Liu, S.; Huang, D.; and Wang, Y. 2020. Multi-Scale Positive Sample Refinement for Few-Shot Object Detection. In *ECCV*.

Xiao, Y.; and Marlet, R. 2020. Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild. In *ECCV*.

Yan, X.; Chen, Z.; Xu, A.; Wang, X.; Liang, X.; and Lin, L. 2019. Meta R-CNN: Towards General Solver for Instance-level Low-shot Learning. In *ICCV*.

Zhang, G.; Lu, S.; and Zhang, W. 2019. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12): 10015–10024.

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018. Single-shot refinement neural network for object detection. In *CVPR*.

Zhang, W.; and Wang, Y.-X. 2021. Hallucination Improves Few-Shot Object Detection. In *CVPR*.

Zhang, W.; Wang, Y.-X.; and Forsyth, D. A. 2020. Cooperating RPN's Improve Few-Shot Object Detection. *arXiv preprint arXiv:2011.10142*.

Zhu, C.; Chen, F.; Ahmed, U.; Shen, Z.; and Savvides, M. 2021a. Semantic relation reasoning for shot-stable few-shot object detection. In *CVPR*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021b. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*.

7 Appendix

This section provides more details of our proposed method and experimental results, which are omitted in the main paper due to space limitation.

7.1 Detailed Architecture of Meta-DETR

The transformer encoder and decoder in the proposed Meta-DETR have similar setups as Deformable DETR (Zhu et al. 2021b). Concretely, both transformer encoder and decoder have 6 layers and adopt the multi-scale deformable attention module, with the proposed correlational aggregation module (CAM) counted as one encoder layer. The channel dimension d is 256, and the intermediate dimension of fully-connected layers (FC) inside the transformer is 1024. The dropout probability is set to 0.1. The number of attention heads is 8. The number of object queries N is 300.

Fig. 7 illustrates the prediction head that produces final predictions. The prediction head locates after the transformer encoder-decoder architecture, and is omitted for simplicity in Fig. 3 of the main paper. It consists of a 1-layer MLP for confidence prediction and a 3-layer MLP for box prediction. The prediction head is shared for all the embeddings generated from the transformer decoder.

7.2 Training Objective of Meta-DETR

Section 4.3 only provides a brief description of the training objective. Here, we provide a mathematically formulated description of the training objective in detail.

Target Generation Let us denote the fixed number of object queries as N, which means Meta-DETR infers N predictions within a single feed-forward process. Let us denote by x_{query} the query image, and $y = \{y_i\}_{i=1}^N$ the ground truth set of objects within the query image, which 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 support classes and their mappings to the task encodings. As discussed in Section 4, 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:

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

Therefore, the detection targets of Meta-DETR can be formulated as:

$$y' = \{y_i'\}_{i=1}^N = \{(c_i', b_i')\}_{i=1}^N = \{\psi(y_i, c_{\text{supp}})\}_{i=1}^N, \quad (5)$$

where $\psi(y_i, c_{\text{supp}})$ acts to filter out irrelevant object annotations and to map the labels of target classes to the labels of the corresponding task encodings, which can be formulated as:

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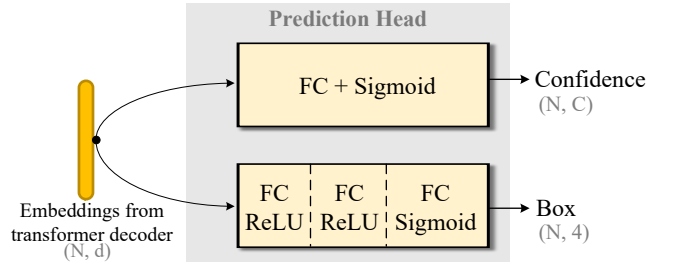

Figure 7: Illustration of the prediction head after the transformer encoder-decoder architecture to produce final predictions. It is shared for all the embeddings generated from the transformer decoder.

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

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:

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^N \mathcal{L}_{\text{match}}(y_i', \hat{y}_{\sigma(i)}), \qquad (7)$$

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:

$$\mathcal{L}_{\text{match}}(y_i', \hat{y}_{\sigma(i)}) = \mathbb{1}_{\{c_i' \neq \varnothing\}} \mathcal{L}_{\text{cls}}(c_i', \hat{c}_{\sigma(i)}) + \\ \mathbb{1}_{\{c_i' \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i', \hat{b}_{\sigma(i)}) . \qquad (8)$$

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

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

where we adopt sigmoid focal loss (Lin et al. 2017) for $\mathcal{L}_{\text{cls}}$ and adopt a linear combination of $\ell 1$ loss and GIoU loss (Rezatofighi et al. 2019) for $\mathcal{L}_{\text{box}}$. Similar to Deformable DETR (Zhu et al. 2021b), $\mathcal{L}(y', \hat{y})$ is applied to every layer of the transformer decoder.

Following Meta R-CNN (Yan et al. 2019), we additionally introduce a cosine similarity cross-entropy loss (Chen et al. 2019) to classify the class prototypes obtained by our designed CAM. It encourages prototypes of different classes to be distinguished from each other.

7.3 Additional Comparison with State of the Art

We also present results taking base classes into consideration in Table 6. While achieving good performance for novel classes with limited training samples, Meta-DETR can still detect objects of base classes with competitive performance. TFA (Wang et al. 2020) produces outstanding performance

Method \ Shot	Base Classes				Novel Classes			
	1	3	5	10	1	3	5	10
Meta-YOLO (Kang et al. 2019)	66.4	64.8	63.4	63.6	14.8	26.7	33.9	47.2
FsDetView (Xiao and Marlet 2020) §	64.2	69.4	69.8	71.1	24.2	42.2	49.1	57.4
TFA w/ cos (Wang et al. 2020) §	**77.6**	**77.3**	**77.4**	**77.5**	25.3	42.1	47.9	52.9
MPSR (Wu et al. 2020) §	60.6	65.9	68.2	69.8	34.7	46.1	49.4	56.7
FSCE (Sun et al. 2021) §	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**

Table 6: Few-shot detection performance (mAP@0.5) for both base and novel classes on the first split of Pascal VOC. § indicates results averaged on multiple runs.

CAM's Location @ Encoder Layer	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

Table 7: Ablation study on the location of our designed CAM. Results are averaged over 10 repeated runs on the first class split of Pascal VOC.

for base classes since it works more like conventional detectors with fine-tuning, thus having relatively constrained capacity in generalizing on novel classes. We also wish to highlight that our proposed Meta-DETR achieves the best base-class and novel-class performance of all the compared meta-learning-based methods.

7.4 Additional Ablation Study

Early Aggregation *vs.* Late Aggregation   We conduct experiments to study the location of our designed correlational aggregation module (CAM) to place. As shown in Table 7, it is preferable to place CAM at the beginning of the transformer encoder, which implies the importance of learning a deep class-agnostic predictor.

7.5 Qualitative Results

We provide multiple qualitative visualizations of Meta-DETR's few-shot detection results in Figs. 8-11, which give a straightforward illustration of the performance of our method. Note that only detection results of novel classes are presented, as the major focus is to detect objects of novel classes. In addition, we only show results with confidence scores higher than 0.25. White boxes indicate correct detections, red solid boxes indicate false positives, and red dashed boxes indicate false negatives. It can be observed that the proposed Meta-DETR is able to detect novel objects at a satisfactory performance even with scarce training samples.

In addition, we also provide a demo video attached in the supplementary materials, which consists of several short clips with Meta-DETR's predictions on novel classes as a reference.
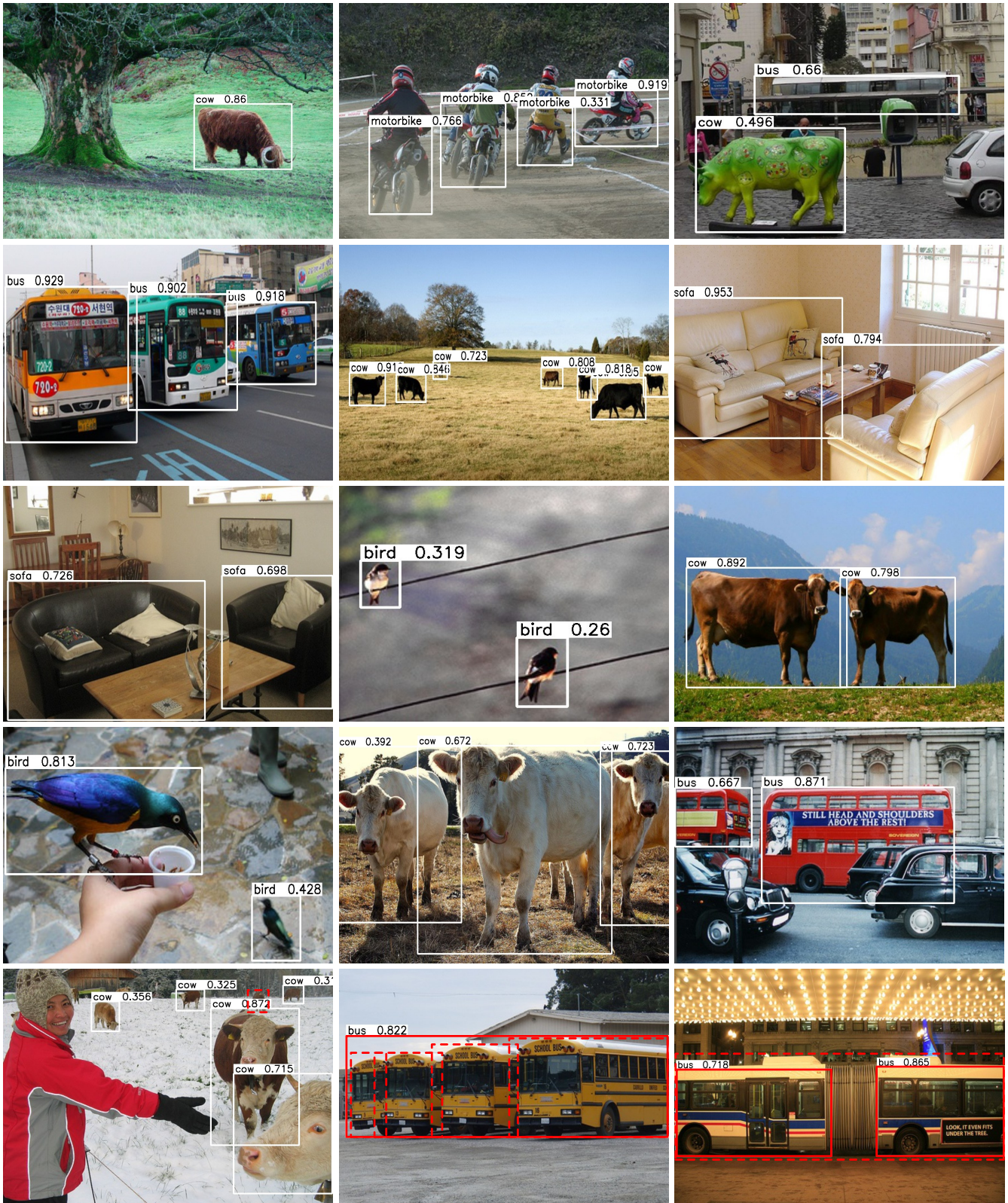
Figure 8: Visualization of Meta-DETR's 10-shot object detection results on Pascal VOC class split 1. Novel classes include bird, bus, cow, motorbike, and sofa. For simplicity, only results of novel classes are illustrated. White boxes indicate correct detections. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.

Figure 9: Visualization of Meta-DETR's 10-shot object detection results on Pascal VOC class split 2. Novel classes include aeroplane, bottle, cow, horse, and sofa. For simplicity, only results of novel classes are illustrated. White boxes indicate correct detections. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.

Figure 10: Visualization of Meta-DETR's 10-shot object detection results on Pascal VOC class split 3. Novel classes include boat, cat, motorbike, sheep, and sofa. For simplicity, only results of novel classes are illustrated. White boxes indicate correct detections. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.
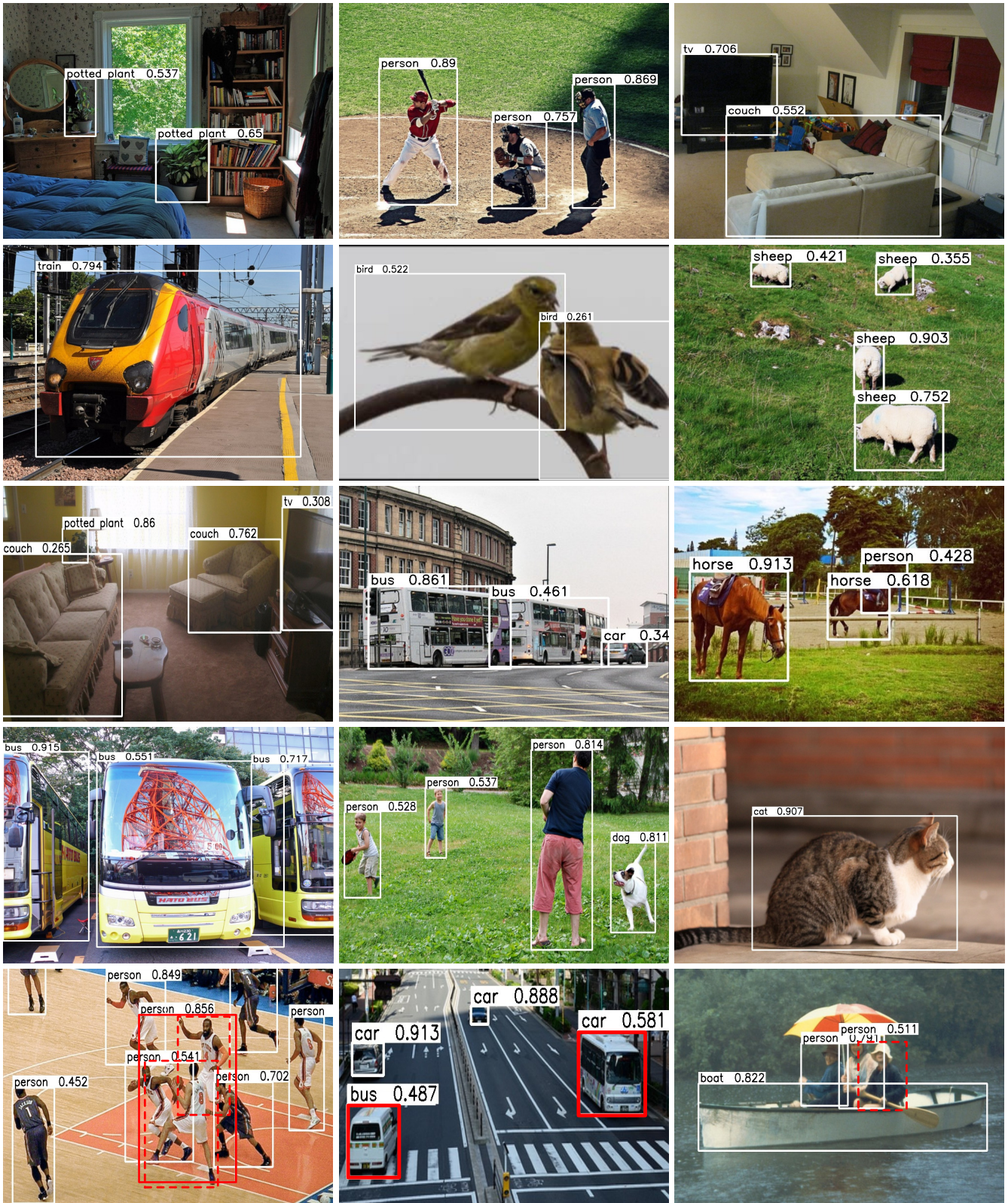
Figure 11: Visualization of Meta-DETR's 10-shot object detection results on MS COCO. Novel classes include person, bicycle, car, motorcycle, airplane, bus, train, boat, bird, cat, dog, horse, sheep, cow, bottle, chair, couch, potted plant, dining table, and tv. For simplicity, only results of novel classes are illustrated. White boxes indicate correct detections. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.