FS-DETR: Few-Shot DEtection TRansformer with prompting and without re-training

Adrian Bulat[1,2], Ricardo Guerrero[1], Brais Martinez[1], Georgios Tzimiropoulos[1,3]
[1]Samsung AI Cambridge [2]Technical University of Iasi [3]Queen Mary University of London

Abstract

This paper is on Few-Shot Object Detection (FSOD), where given a few templates (examples) depicting a novel class (not seen during training), the goal is to detect all of its occurrences within a set of images. From a practical perspective, an FSOD system must fulfil the following desiderata: (a) it must be used as is, without requiring any fine-tuning at test time, (b) it must be able to process an arbitrary number of novel objects concurrently while supporting an arbitrary number of examples from each class and (c) it must achieve accuracy comparable to a closed system. Towards satisfying (a)-(c), in this work, we make the following contributions: We introduce, for the first time, a simple, yet powerful, few-shot detection transformer (FS-DETR) based on visual prompting that can address both desiderata (a) and (b). Our system builds upon the DETR framework, extending it based on two key ideas: (1) feed the provided visual templates of the novel classes as visual prompts during test time, and (2) "stamp" these prompts with pseudo-class embeddings (akin to soft prompting), which are then predicted at the output of the decoder. Importantly, we show that our system is not only more flexible than existing methods, but also, it makes a step towards satisfying desideratum (c). Specifically, it is significantly more accurate than all methods that do not require fine-tuning and even matches and outperforms the current state-of-the-art fine-tuning based methods on the most well-established benchmarks (PASCAL VOC & MSCOCO).

1. Introduction

Thanks to the advent of deep learning, object detection has witnessed tremendous progress over the last years. However, the standard setting of training and testing on a closed set of classes has specific important limitations. Firstly, it's unfeasible to annotate all objects of relevance present in-the-wild, thus, current systems are trained only on a small subset. It does not seem straightforward to significantly scale up this figure. Secondly, human perception operates mostly under the open set recognition/detection setting. Humans can detect/track new unseen objects on the fly, typically using a single template, without requiring any "re-training" or "fine-tuning" of their "detection" skills, arguably a consequence of the prior representation learned, an aspect we sought to exploit here too. Finally, important applications in robotics, where agents may interact with previously unseen objects, might require their subsequent detection on the fly without any re-training. Few-Shot Object Detection (FSOD) refers to the problem of detecting a novel class not seen during training and, hence, can potentially address many of the aforementioned challenges.

There are still important *desiderata* that current FSOD system must address in order to be practical and flexible to use: (a) They must be used as is, not requiring *any* re-training (*e.g.* fine-tuning) at test time - a crucial component for autonomous exploration [24]. However, many existing state-of-the-art FSOD systems (*e.g.* [37, 44, 32]) rely on re-training with the few available examples of the unseen classes. While such systems are still useful, the requirement for re-training makes them significantly more difficult to deploy on the fly and in real-time or on devices with limited capabilities for training. (b) They must be able to handle an arbitrary number of novel objects (and moreover an arbitrary number of examples per novel class) simultaneously during test time, in a single forward pass without requiring batching. This is akin to how closed systems work, which are able to detect multiple objects concurrently. However, to our knowledge there is no FSOD system possessing this property without requiring re-training. (c) They must attain classification accuracy that is comparable to that of closed systems. However, existing FSOD systems are far from achieving such high accuracy, especially for difficult datasets like MSCOCO.

This work aims to significantly advance the state-of-the-art in all three above-mentioned challenges. To this end, and building upon the DETR [3] framework, we propose a system, called Few-Shot Detection Prompting (FS-DETR), capable of detecting multiple novel classes at once, supporting a variable number of examples per class, and importantly, without any extra re-training. In our system, the visual template(s) (*i.e.* prompts) from the new class(es) are used, during test time, in two ways: (1) in FS-DETR's encoder to filter the backbone's image features via cross-attention, and more importantly, (2) as visual prompts in FS-DETR's decoder, "stamped" with special pseudo-class encodings and prepended to the learnable object queries. The pseudo-class encodings are used as pseudo-classes which a classification head attached to the object queries is trained to predict via a Cross-Entropy loss. Finally, the output of the decoder are the predicted pseudo-classes and regressed bounding boxes. The two components, when combined allow the creation of a FSOD model that can localise, within one forward pass multiple objects at once, each with an arbitrary number of examples, without retraining.

Contrary to prior work (*e.g*. TSF [21] and AirDet [24]), FS-DETR, akin to soft-prompting [19], "instructs" the model in the input space regarding the visual appearance of the searched object(s). The network is then capable of predicting for each prompt (*i.e*. visual template) all the locations at which it is present in the image, if any. This is achieved without any additional modules or carefully engineered structures and feature filtering mechanisms (*e.g*. TSF [21] AirDet [24]). Instead, we directly append the prompts to the object queries of the decoder.

In summary, **our main contributions** are:

1. We propose a fine-tuning-free Few-Shot Detection Prompting (FS-DETR) method which is capable of detecting multiple novel objects at once, and can support an arbitrary number of samples per class in an efficient manner via soft visual prompting.

2. We show that all these features can be enabled by extending DETR based on two key ideas: (1) feed the provided visual templates of novel classes as visual prompts during test time, and (2) "stamp" these prompts with (class agnostic) pseudo-class embeddings, which are then predicted at the output of the decoder along with bounding boxes (akin to soft-prompting).

3. We also propose a simple and efficient yet powerful pipeline consisting of unsupervised pre-training followed by prompt-like base class training.

4. In addition to being more flexible, our system matches and outperforms state-of-the-art results on the standard FSOD setting on PASCAL VOC and MSCOCO. Specifically, FS-DETR outperforms the not re-trained methods of [13, 24] and most re-training based methods on extreme few-shot settings ($k = 1, 2$), while being competitive for more shots.

2. Related work

DEtection TRansformer (DETR) approaches: After revolutionizing NLP [40, 33], Transformer-based architectures have started making significant impact in computer vision problems [5, 30]. In object detection, methods are typically grouped into two-stage (proposal-based) [34, 16, 2] and single-stage (proposal-free)[26, 29, 39, 52, 22] methods. In this field, a recent breakthrough is the DEtection TRansformer (DETR) [3], which is a single-stage approach that treats the task as a direct set prediction without requiring hand-crafted components, like non-maximum suppression or anchor generation. Specifically, DETR is trained in an end-to-end manner using a set loss function which performs bipartite matching between the predicted and the ground-truth bounding boxes. Because DETR has slow training convergence, several methods have been proposed to improve it [31, 54, 4]. Conditional DETR [31] learns a conditional spatial query from the decoder embeddings that are used in the decoder for cross-attention with the image features. Deformable DETR [54] proposes deformable attention in which attention is performed only over a small set of key sampling points around a reference point. Unsupervised pre-training of DETR [4] (UP-DETR), improves its convergence, where randomly cropped patches are summed to the object queries and the model is then trained to detect them in the original image. A follow-up work, DETReg [1], replaces the random crops with proposals generated by Selective Search. While our approach is agnostic to the exact variant of DETR, due to its fast training convergence, we opted to use Conditional DETR as the model that we build our FS-DETR approach upon. Beyond this, the above mentioned works are on closed set recognition and while UP-DETR's unsupervised pre-training could be potentially used for few-shot detection, the experimental setting presented in their work doesn't match the standard settings for few-shot detection and no code is provided for its training. We re-implemented UP-DETR [4] for few-shot detection and found that our method outperforms it. This is expected as their goal is unsupervised pre-training and not FSOD.

Few Shot Object Detection (FSOD) methods can be categorised into *re-training based* and *without re-training* methods. Re-training based methods assume that during test time, but before deployment, the provided samples of the novel categories can be used to fine-tune the model. This setting is restrictive as it requires training before deployment. Instead, without re-training methods can be directly deployed on the fly for the detection of novel examples.

Re-training based approaches can be divided into meta-learning and fine-tuning approaches. Meta-learning based approaches attempt to transfer knowledge from the base classes to the novel classes through meta-learning [11, 12, 47, 43, 23, 46]. Fine-tuning based methods follow the standard pre-train and fine-tune pipeline. They have been shown to significantly outperform meta-learning approaches. TFA [42] proposes fine-tuning the final classification layer of a Faster R-CNN model (first trained on base classes), with a balanced subset containing also the examples of the novel classes. SRR-FSD [53] proposes to construct a semantic space using word embeddings, and then train a FSOD by projecting and aligning object visual features with their corresponding text embeddings. CME [25] proposes to learn a feature embedding space where the margins between novel classes are maximised. Retentive R-CNN [10] addresses the problem of learning a FSOD without catastrophic forgetting (*i.e.* without compromising base class accuracy). FSCE [37] aims to decrease instance similarity between objects belonging to different categories by adding a secondary branch to the primary RoI head, which is trained via supervised contrastive learning. The method of [51] proposes a hallucinator network to generate examples which can help the classifier learn a better decision boundary for the novel classes. FSOD-UP [44] proposes to construct universal prototypes capturing invariant object characteristics which, via fine-tuning, are adapted to the novel categories. DeFRCN [32] proposes to perform stop-gradient between the RPN and the backbone, and scale-gradient between RCNN and the backbone.

More recently, FSODMC [9] proposes to address base class bias via novel class fine-tuning while calibrating the RPN, detector and backbone components to preserve well-learned prior knowledge. KFSOD [50] improves upon [8] by replacing the class-specific average-pooling of features with kernel-pooled representations that are meant to capture non-linear patterns. TENET [49] extends KFSOD with a multi-head attention transformer block on 2nd-, 3rd- and 4th-order pooling. FCT [15] extends [13] by incorporating a cross-transformer into both the feature backbone and detection head to encourage query-support multi-level interactions. Their approach is based on two-stage Faster-RCNN trained with a binary cross-entropy loss, *i.e.* it is entirely different from our architecture and training objective based on pseudo-class prediction. Meta-DETR [48] proposes a correlation aggregation module, which is then placed before a standard DETR encoder-decoder, that filters the query image tokens using the support images and tasks. In contrast, we model the interactions directly via a novel visual template prompting formulation, without any additional modules and can process an arbitrary number of examples per-object and object within the same forward pass. Moreover, their method requires finetuning for FSOD deployment, while our doesn't require any retraining. TSF [21] proposes a transformer plugin module for modelling interactions the input features f and a set of learnable parameters θ representing base class information (*i.e.* prototypes). In contrast to [21], our approach does not learn any type of base class prototypes and is fully dynamic (interactions between data and data as opposed to data and prototypes).

Without re-training approaches are primarily based on metric learning [41, 36]. A standard approach is [17], which uses cross-attention between the backbone's and the query's features to refine the proposal generation, then re-uses the query to re-weight the RoI features channel-wise (in a squeeze-and-excitation manner) for novel class classification. A similar approach for proposal generation is described in [8], where the squeeze-and-excitation module is replaced with a multi-relation network. QA-FewDet [13] extends [17, 8] by modelling class-class, class-proposal and proposal-proposal relationships using various GCNs. Finally, the concurrent work of AirDet [24] attempts to learn a set of prototypes and a cross-scale support guided proposal network, with the association and regression performed at the end of the model via a detection head. To our knowledge, AirDet represents the state-of-the-art FSOD without re-training. We show that the proposed FS-DETR outperforms it by a large margin.

Relation to our work: Our method is the first to perform re-training free visual prompting for few shot object detection. Different to many other works (*e.g.* TSF [21], AirDet [24]), FS-DETR does not learn perform visual prompting nor learn class-related prototypes (*i.e.* soft prompts-like). We emphasize that the pseudo-class embeddings in FS-DETR are class-agnostic Finally, there are methods which are trained using metric learning [8, 13, 15] using a binary cross entropy loss. In contrast, FS-DETR is trained to predict pseudo-classes using cross entropy (in a class-agnostic way) which is a more powerful training objective.

3. Method

Given a dataset where each image is annotated with a set of bounding boxes representing the instantiations of C known base classes, our goal is to train a model capable of localizing objects belonging to novel classes, *i.e.* unseen during training, using up to k examples per novel class. In practice, we partition the available datasets into two disjoint sets, one containing C_{novel} classes for testing, and another with C_{base} classes for training (*i.e.* $C = C_{novel} \cup C_{base}$ and $C_{novel} \cap C_{base} = \emptyset$).

3.1. Overview of FS-DETR

We build the proposed Few-Shot DEtection TRansformer (FS-DETR) upon DETR's architecture [1]. FS-DETR's architecture consists of: (1) the CNN backbone used to extract visual features from the target image and the templates, (2) a transformer encoder that performs self-attention on the image tokens and cross-attention between the templates and the image tokens, and (3) a transformer decoder that processes object queries and templates to make predictions for pseudo-classes (see also below) and bounding boxes. Contrary to the related works of [8, 13, 14], our system processes an arbitrary number of templates (*i.e.* new classes) jointly,

[1] We note that, in practice, due to its superior convergence properties, we used the Conditional DETR as the basis of our implementation but for simplicity of exposition we will use the original DETR architecture.

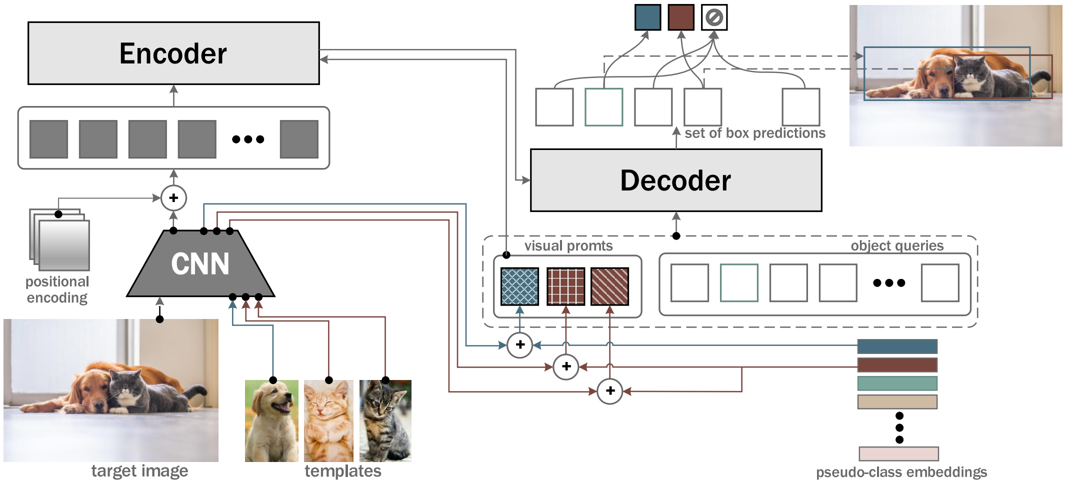

Figure 1. In the proposed FS-DETR, the available templates are provided as additional *visual prompts* to the system in order to condition and control the output. To train and test the system, these prompts are "stamped" with *pseudo-class embeddings* (see Sec. 3.2) which are predicted at the output of the decoder along with bounding boxes (note, that there is *no correlation* between actual classes and pseudo-classes, *e.g*. the cat could be of either class: "blue" or "red" as there is no preferred order). FS-DETR naturally supports $k-$shot detection, as the model can process multiple examples per class at once. Templates belonging to the same class will share the same pseudo-class embedding. Red and blue colors denote the different pseudo-classes associated to the input templates.

in a single forward pass, *i.e*. without requiring batching, significantly improving the efficiency of the process.

Key contributions: DETR re-formulates object detection as a set prediction problem via prompting, making object predictions by "tuning" a set of N learnable queries $\mathbf{O} \in \mathbb{R}^{N \times d}$ to the image features through cross-attention. The queries $\mathbf{O}$ are used as prompts in DETR for closed-set object detection. To accommodate for open-set FSOD, we propose to provide novel classes' templates as additional *visual prompts* in order to condition and control the detector's output. To train the system, we also propose to "stamp" these prompts with *pseudo-class embeddings*, akin to soft-prompting, which are then predicted by the decoder along with bounding boxes. This can be viewed as an analogous component to the traditional positional embeddings. The proposed FS-DETR is depicted in Fig. 1. Compared to [3], we highlight key differences in our mathematical formulation in red.

3.2. FS-DETR

The following subsections detail FS-DETR's architecture and main components.

Template encoding: Let $\mathbf{T}_{i,j} \in \mathbb{R}^{H_p \times W_p \times 3}, i \in \{1,\dots,m\}, j \in \{1,\dots,k\}$ be the template images of the available classes (sampled from C_{base} during training) where m is the number of classes at the current training iteration (m can vary), and k is the number of examples per class (*i.e*. k-shot detection; k can also vary). A CNN backbone (*e.g*. ResNet-50) generates template features $\mathbf{X} = \texttt{CNN}(\mathbf{T}),\ \mathbf{X} \in \mathbb{R}^{mk \times d}$ using either average or attention pooling (see Sec. 5).

Pseudo-class embeddings: We propose to dynamically and randomly associate, at each training iteration, the k template prompts in $\mathbf{X}$ belonging to the i-th class (for that iteration) with a pseudo-class represented by a pseudo-class embedding $\mathbf{c_i^s} \in \mathbb{R}^d$, which are added to the templates as follows:

$$\mathbf{X}^s = \mathbf{X} + \mathbf{C}^s, \tag{1}$$

where $\mathbf{C}^s \in \mathbb{R}^{mk \times d}$ contains the pseudo-class embeddings for all templates at the current iteration. The pseudo-class embeddings are initialised from a normal distribution and learned during training. They are not determined by the ground-truth categories and are class-agnostic. During each inference step, we arbitrarily associate to a template prompt (belonging to some class) the i-th embedding as described by Eq. 1. The goal is to retrieve the pseudo-class i. Note that the actual class information is not used. As the assigned embedding changes at every iteration, there is no correlation between the actual classes and the learned embeddings. See also Fig. 1 that exemplifies this process. In the proposed FS-DETR, each decoded object query $\mathbf{o}_i$ in $\mathbf{O}$ will attempt to predict a pseudo-class using a classifier. Pseudo-class embeddings add a signature to each visual prompt allowing the network to track the template within and dissociate it from the rest of the templates belonging to a different class. As transformers are permutation invariant, this vectors are required in order to track the visual prompt within the model.

Templates as visual prompts: We propose to provide the templates $\mathbf{X}^s$ as *visual prompts* to the system by prepending them to the sequence of object queries fed to the decoder:

$$\mathbf{O}' = [\mathbf{X}^s \quad \mathbf{O}], \quad \mathbf{O}' \in \mathbb{R}^{(mk+N) \times d}. \tag{2}$$

As shown below, the templates will induce pseudo-class related information into the object queries via attention. This can be interpreted as a new form of training-aware soft-prompting [28].

FS-DETR encoder: Given a target image $\mathbf{I} \in \mathbb{R}^{H' \times W' \times 3}$, the same CNN backbone used for template feature extraction first generates image features $\mathbf{Z} = \mathtt{CNN}(\mathbf{I})$, $\mathbf{Z} \in \mathbb{R}^{S \times d}$, $S = H \times W$, which are enriched with positional information through positional encodings $\mathbf{Z} \leftarrow \mathbf{Z} + \mathbf{P}_s$, $\mathbf{P}_s \in \mathbb{R}^{S \times d}$. The features $\mathbf{Z}$ are then processed by FS-DETR's encoder layers in order to be enriched with global contextual information. The $l-$th encoding layer processes the output features of the previous layer $\mathbf{Z}^{l-1}$ using a series of Multi-Head Self-Attention (MHSA), Layer Normalization (LN), and MLP layers (typical in [40] and [3]), as well as a newly proposed Multi-Head Cross-Attention (MHCA) layer as follows [2]:

$$\begin{aligned} \mathbf{Z}' &= \text{MHSA}(\text{LN}(\mathbf{Z}^{l-1})) + \mathbf{Z}^{l-1}, && (3)\\ \mathbf{Z}'' &= \text{MHCA}(\text{LN}(\mathbf{Z}'), \mathbf{X}^s) + \mathbf{Z}', && (4)\\ \mathbf{Z}^l &= \text{MLP}(\text{LN}(\mathbf{Z}'')) + \mathbf{Z}''. && (5) \end{aligned}$$

The purpose of the MHCA layer above is to filter and highlight early on, before decoding, the image tokens of interest. We have found that such a layer noticeably increases few-shot accuracy (see also Section 5). FS-DETR's encoder is implemented by stacking $L = 6$ blocks, each following Eq. (3)-(5). As image tokens are permutation invariant, we followed [3] and used a fixed positional encoding. For the templates, pseudo-class embeddings serve as positional encodings.

FS-DETR decoder: FS-DETR's decoder accepts as input the concatenated templates and learnable object queries $\mathbf{O}'$ which are transformed by the decoder's layers through self-attention and cross-attention layers in order to be eventually used for pseudo-class prediction and bounding box regression. The $l-$th decoding layer processes the output features of the previous layer $\mathbf{V}^{l-1}$ as follows:

$$\begin{aligned} \mathbf{V}' &= \text{MHSA}(\text{LN}(\mathbf{V}^{l-1}) + \mathbf{O}') + \mathbf{V}^{l-1}, && (6)\\ \mathbf{V}'' &= \text{MHCA}(\text{LN}(\mathbf{V}') + \mathbf{O}', \mathbf{Z}^l) + \mathbf{V}', && (7)\\ \mathbf{V}^l &= \text{MLP}(\text{LN}(\mathbf{V}'')) + \mathbf{V}'', && (8) \end{aligned}$$

where $\mathbf{V}^0 = [\mathbf{X}^s \quad \mathtt{zeros}(N, d)]$. Notably, different MLPs are used to process the decoder's features $\mathbf{V} = [\mathbf{V}_{\mathbf{X}^s} \quad \mathbf{V}_{\mathbf{O}}]$ corresponding to the templates $\mathbf{V}_{\mathbf{X}^s}$ and the object queries $\mathbf{V}_{\mathbf{O}}$:

$$\mathtt{MLP}(\mathbf{V}) = [\mathtt{MLP}(\mathbf{V}_{\mathbf{X}^s}) \quad \mathtt{MLP}(\mathbf{V}_{\mathbf{O}})]. \qquad (9)$$

FS-DETR's decoder consists of $L = 6$ layers implemented using Eqs. (6)-(9).

FS-DETR training and loss functions: For each base class that exists in the target image, we create a template for that class by randomly sampling and cropping an object from that category using a different image (containing an object of the same class) from the train set. After applying image augmentation, the cropped object/template is passed through the CNN backbone of FS-DETR. For each target image and template i (depicted in that image), the ground truth is $y_i = (c_i^s, b_i)$, where c_i^s is the target pseudo-class label (up to m classes in total) and $b_i \in [0, 1]^4$ are the normalised bounding box coordinates. To calculate the loss for training FS-DETR, only the N transformed object queries $\mathbf{V}_{\mathbf{O}}^L \in \mathcal{R}^{N \times d}$ from the output of the last decoding layer are used for pseudo-class classification and bounding box regression (*i.e.* $\mathbf{V}_{\mathbf{X}^s}^L$ is not used). To this end, pseudo-class and bounding box prediction heads are used to produce a set of N predictions $\{\hat{y}_i\}_{i=1}^N$ consisting of the pseudo-class probabilities $\hat{p}_i(c^s)$ and the bounding box coordinates $\hat{b}_i$. The heads are implemented using a 3-layer MLP and ReLU activations. Similarly to [3], we used an additional special pseudo-class $\varnothing$ to denote tokens without valid object predictions. Note that as the training is done in a class-agnostic way via mapping of the base class templates to pseudo-classes (the actual class information is discarded) the model is capable to generalise to the unseen novel categories.

Following [3], bipartite matching is used to find an optimal permutation $\{\hat{y}_{\sigma i}\}_{i=1}^N$. Finally, the loss is:

$$\begin{aligned} L = \sum_{i=1}^{N} & \lambda_1 L_{\text{CE}}(c_i^s, \hat{p}_{\sigma(i)}(c^s)) \\ & + \lambda_2 ||b_i - \hat{b}_{\sigma(i)}||_1 + \lambda_3 \mathtt{IoU}(b_i, \hat{b}_{\sigma(i)}), \end{aligned} \qquad (10)$$

where $\mathtt{IoU}$ is the GIoU loss of [35] and λ_i are the loss weights.

Pre-training: Transformers are generally *more data hungry* than CNNs due to their lack of inductive bias [5]. Therefore, building representations that generalise well to unseen data, and prevent overfitting within the DETR framework, requires larger amounts of data. To this end, we used images from ImageNet-100 [38] and to some extent MSCOCO, for unsupervised pre-training where the classes and the bounding boxes are generated on-the-fly using an object proposal system, *without using any labels*. Note, that unlike all prior works, we make use of neg. templates as prompts, training the network using our proposed loss. See also the appendix.

4. Experiments

Datasets: Experiments presented here were all conducted using PASCAL VOC [7, 6] and MSCOCO [27] datasets. Moreover, ImageNet100 [38], consisting of ∼125K images and 100 categories, is used (without labels) to pre-train our object detector. PASCAL VOC and MSCOCO are used to train and evaluate few-shot experiments. Following previous works [20, 42, 13], we evaluate the proposed method on PASCAL VOC 2007+2012 and MSCOCO 2014, using the same data splits provided by [20, 42]. Specifically, PASCAL

[2] We follow DETR's notation where $\mathbf{O}'$ is added to $LN(\mathbf{V}^{l-1})$ and then projected to form the query Q and key K for self-attention. Here, the first layer $\mathbf{V}^{l-1}$ is initialised as $[X_s, zeros]$ while in DETR with $zeros$.

VOC is randomly divided into three different splits, each consisting of 15 base and 5 novel classes; training is done on the PASCAL VOC 2007+2012 train/val sets, and evaluation on the PASCAL VOC 2007 test set. Similarly, MSCOCO is split into base and novel categories, where the 20 overlapping categories with PASCAL VOC are considered novel, while the remainder are the base categories; following recent convention [20, 42, 13], 5k samples from the validation set are held out for testing, while the remaining samples from both train and validation sets are used for training.

Evaluation setting: There are currently two widely-used FSOD evaluation protocols. The first focuses exclusively on novel classes while disregarding base class performance, thus not monitoring catastrophic forgetting. The second, more comprehensive protocol, called generalised few-shot object detection (G-FSOD), considers both base and novel classes. The choice of protocol and, hence, results interpretation, bears special importance for re-training based methods, as base class generalizability might be compromised. Without re-training methods, as FS-DETR, adhere to the second protocol (G-FSOD) by default, as base class catastrophic forgetting is not applicable. As in [42], we report results from several runs using different templates, and hence, report competing results using a likewise setting when available.

Baselines: Existing FSOD methods can be broadly categorised into: re-training based, and without re-training. The latter can handle few-shot detection on the fly at deployment, while re-training based FSOD methods generally tend to perform better. Re-training based methods can be further subdivided into "meta-learning" and "fine-tuning" approaches. *"Re-training based: meta-learning"* approaches include: Xiao *et al*. [46], DCNET [18], TIP [23] and QA-FewDet [13]. *"Re-training based: fine-tuning"* methods include: Fan *et al*. [8], CME [25], SRR-FSD [53], Zhang *et al*. [51], DeFRCN [32], FCT [15], KFSOD [50], TENET [49], FSODMC [9], DETReg [1], Meta-DETR [48] and tsf [21]. *"Without re-training"* methods include: UP-DETR [4], Fan *et al*. [8], QA-FewDet [13], Meta Faster R-CNN [14] and AirDet [24]. Note that these 4 last methods can also be re-trained, offering improved accuracy.

4.1. Overview of Results

From our results below on both datasets, we can take away two messages:

- **Conclusion 1:** FS-DETR outperforms **all** without re-training based approaches by a large margin, *i.e*. those directly comparable.
- **Conclusion 2:** FS-DETR outperforms the majority of re-training based approaches (some by a large margin). Importantly, on average across all novel sets from PASCAL VOC, it outperforms **all** for $k = 1$, while at the same time being more robust across splits, *i.e*. FS-DETR has lower variance across novel sets. Similarly, in MSCOCO with $k = 1, 2$ it outperforms nearly all re-trained methods.
- **Conclusion 2:** FS-DETR Shows strong and conistent performance across a variable number of shots.

4.2. Results on PASCAL VOC

Table 1 summarises our results and compares them with the current state-of-the-art on PASCAL VOC. Experiments for k-shot detection were conducted for three data splits, where k was set to $1, 2, 3, 4, 5, 10$ and AP50 values are reported. Note that Table 1 is split into two sections: Methods at the top require an additional few-shot re-training stage, while those at the bottom, including our method, do not require any re-training. Here, it can be appreciated that our approach outperforms all without re-training methods by a large margin, improving the current state-of-the-art [13, 4] in any shot and all split experiments by up to 17.8 AP50 points, and, in most cases, by at least $\sim$10 AP50 points. Moreover, and contrary to [13, 14], our method can process multiple novel classes in a single forward pass. Finally, we re-implemented UP-DETR [4] for few-shot detection on PASCAL VOC (since there is no publicly available implementation for few-shot detection or results). Our method largely outperforms it, perhaps unsurprisingly, as the latter was not developed for few-shot detection, but for unsupervised pre-training. Importantly, the proposed method provides competitive results or even outperforms re-training based methods (meta-learned or fine-tuned), especially for extreme low-shot, $k = 1, 2$ (*e.g*. [23, 25, 53, 32, 48, 21]). Qualitative visualizations in appendix.

4.3. Results on MSCOCO

Table 2 shows evaluation results for FS-DETR and all competing state-of-the-art methods on MSCOCO. Similarly to above, Table 2 is split into methods requiring re-training at the top and those that do not require re-training at the bottom. There, it can be appreciated that FS-DETR outperforms all comparable state-of-the-art methods [13, 8, 24] by up to 3.1 AP50 points (1-shot) and, in most cases, by at least $\sim$ 1.1 AP50 points. In our experiments UP-DETR failed to converge on MSCOCO, hence, results are not included. We speculate that this might be due to UP-DETR's partitioning the input queries by the number query patches, therefore, limiting the number of tokens query patches interact with. This appears to be too restrictive for MSCOCO. Moreover, and in line with results observed on PASCAL VOC, FS-DETR achieves competitive results to those of re-trained based methods on MSCOCO, a far more challenging dataset. FS-DETR outperforms nearly all re-training based methods, for $k = 1, 2$ while performing comparably for $k = 3, 5, 10$. This in itself is a promising results given our methods doesn't require retraining.

Table 1. FSOD performance (AP50) on the PASCAL VOC dataset. Results with † are from [13] while those with ‡ were produced by us. Our approach outperforms all without re-training methods. Moreover, it provides competitive results compared with other re-training based methods for $k = 3, 5, 10$, and even outperforms most for $k = 1, 2$, *i.e.* extreme few-shot settings.

Method / Shot	Venue	Backbone	Novel Set 1					Novel Set 2					Novel Set 3				
			1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
Re-training based methods (meta-learning or fine-tuning)																	
FSRW* [20]	ICCV'19	YOLOv2	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
MetaDet* [43]	ICCV'19	VGG16	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
Meta R-CNN* [47]	ICCV'19	RN-101	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
TFA w/fc [42]	ICML'20	RN-101	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
TFA w/cos [42]	ICML'20	RN-101	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
TFA w/cos* [42]	ICML'20	RN-101	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
Xiao *et al.* [46]	ECCV'20	RN-101	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
MPSR* [45]	ECCV'20	RN-101	41.7	42.5	51.4	55.2	61.8	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
Fan *et al.*† [8]	CVPR'20	RN-101	37.8	43.6	51.6	56.5	58.6	22.5	30.6	40.7	43.1	47.6	31.0	37.9	43.7	51.3	49.8
DCNET [18]	CVPR'21	RN-101	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
TIP [23]	CVPR'21	RN-101	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
CME [25]	CVPR'21	RN-101	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
SRR-FSD [53]	CVPR'21	RN-101	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
Zhang *et al.** [51]	CVPR'21	RN-101	47.0	44.9	46.5	54.7	54.7	26.3	31.8	37.4	37.4	41.2	40.4	42.1	43.3	51.4	49.6
QA-FewDet [13]	ICCV'21	RN-101	42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
DeFRCN [32]	ICCV'21	RN-101	40.2	53.6	58.2	63.6	66.5	29.5	39.7	43.4	48.1	52.8	35.0	38.3	52.9	57.7	60.8
DeFRCN+TSF [21]	ECCV'22	RN-101	43.6	57.4	61.2	65.1	65.9	31.0	40.3	45.3	49.6	52.5	39.3	51.4	54.8	59.8	62.1
DeFRCN* [32]	ICCV'21	RN-101	53.6	57.5	61.5	64.1	60.8	30.1	38.1	47.0	53.3	47.9	48.4	50.9	52.3	54.9	57.4
FCT [15]	CVPR'22	PVTv2-B2-Li	38.5	49.6	53.5	59.8	64.3	25.9	34.2	40.1	44.9	47.4	34.7	43.9	49.3	53.1	56.3
KFSOD [50]	CVPR'22	RN-50	44.6	-	54.4	60.9	65.8	37.8	-	43.1	48.1	50.4	34.8	-	44.1	52.7	53.9
TENET [49]	ECCV'22	RN-50	46.7	-	55.4	62.3	66.9	40.3	-	44.7	49.3	52.1	35.5	-	46.0	54.4	54.6
Meta Faster R-CNN [14]	AAAI'22	RN-101	43.0	54.5	60.6	66.1	65.4	27.7	35.5	46.1	47.8	51.4	40.6	46.4	53.4	59.9	58.6
Meta-DETR [48]	TPAMI'22	DETR-R101	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
Without re-training methods																	
Fan *et al.*† [8]	CVPR'20	RN-101	32.4	22.1	23.1	31.7	35.7	14.8	18.1	24.4	18.6	19.5	25.8	20.9	23.9	27.8	29.0
UP-DETR‡ [4]	ICCV'21	DETR-R50	38.2	40.4	44.5	45.8	46.0	20.0	23.6	25.8	28.0	33.9	34.1	35.3	37.0	40.1	40.3
QA-FewDet [13]	ICCV'21	RN-101	41.0	33.2	35.3	47.5	52.0	23.5	29.4	37.9	35.9	37.1	33.2	29.4	37.6	39.8	41.5
Meta Faster R-CNN [14]	AAAI'22	RN-101	40.2	30.5	33.3	42.3	46.9	26.8	32.0	39.0	37.7	37.4	34.0	32.5	34.4	42.7	44.3
FS-DETR (Ours)	this work	DETR-R50	**45.0**	**48.5**	**51.5**	**52.7**	**56.1**	**37.3**	**41.3**	**43.4**	**46.6**	**49.0**	**43.8**	**47.1**	**50.6**	**52.1**	**56.9**

5. Ablation studies

Herein, we ablate different variations and components of our method, analysing the impact of different design choices. Unless otherwise specified, we report results on Novel Set 1 on PASCAL VOC. For more details see the appendix.

Design of the prompt template encoder: An important component of our system is the extraction of discriminative prompts from the novel classes' templates. To this end, we re-use FS-DETR's input image CNN encoder. However, to focus on the most important components we used attention-based pooling instead of simple global average pooling. In Table 3 we report the impact of: (a) resolution, (b) augmentation level, and (c) pooling type. As the results show, increasing the resolution from 128 to 192px yields no additional gains. This suggests that, at least for the datasets in question, fine grained details are not quintessential for the identification of the targeted novel class and higher level concepts suffice. While spatial augmentation generally helps (*i.e.* for object recognition), we found that adding noise to the ground truth bounding box of the template at train time leads to lower accuracy. This makes the problem for the object detector too hard, and impedes convergence. Finally, attentive pooling, instead of global average, can further boost performance.

Pre-training: Many FSOD systems use pre-trained backbones on ImageNet for classification. Deviating from this, we pre-train our system in an unsupervised manner on ImageNet images and parts of MSCOCO without using the labels. We note that this is especially important for transformer based architectures which were shown to be more prone to over-fitting due to the lack of inductive bias [5]. As the results from Table 4 show, unsupervised pre-training, can significantly boost the performance, preventing over-fitting toward the base classes and improving overall discriminative capacity. To reduce over-fitting the pre-training loss on ImageNet data is applied during supervised training every 8th iteration. Additional details and experiments in appendix.

Auxiliary losses: We explored the impact of using additional auxiliary losses applied to the object queries, an L_2 feature loss and a contrastive loss, where the positive pairs are formed by taking the input templates with all the object query tokens assigned to it by the Hungarian assignment algorithm. We did not observe any further gains from the

Table 2. FSOD performance on the MSCOCO dataset. Results with † are from [13]. Our method consistently outperforms the state-of-the-art methods in most of the shots and metrics.

	1-shot			2-shot			3-shot			5-shot			10-shot		
Method	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
Re-trained methods (meta-learning or fine-tuning)															
FSRW* [20]	–	–	–	–	–	–	–	–	–	–	–	–	5.6	12.3	4.6
MetaDet* [43]	–	–	–	–	–	–	–	–	–	–	–	–	7.1	14.6	6.1
Meta R-CNN* [47]	–	–	–	–	–	–	–	–	–	–	–	–	8.7	19.1	6.6
TFA w/fc [42]†	2.9	5.7	2.8	4.3	8.5	4.1	6.7	12.6	6.6	8.4	16.0	8.4	10.0	19.2	9.2
TFA w/cos [42]	1.9	3.8	1.7	3.9	7.8	3.6	5.1	9.9	4.8	7.0	13.3	6.5	9.1	17.1	8.8
TFA w/cos*† [42]	3.4	5.8	3.8	4.6	8.3	4.8	6.6	12.1	6.5	8.3	15.3	8.0	10.0	19.1	9.3
Xiao *et al.*† [46]	3.2	8.9	1.4	4.9	13.3	2.3	6.7	18.6	2.9	8.1	20.1	4.4	10.7	25.6	6.5
Fan *et al.*† [8]	4.2	9.1	3.0	5.6	14.0	3.9	6.6	15.9	4.9	8.0	18.5	6.3	9.6	20.7	7.7
TIP [23]	-	-	-	-	-	-	-	-	-	-	-	-	16.3	33.2	14.1
CME [18]	-	-	-	-	-	-	-	-	-	-	-	-	15.1	24.6	16.4
SRR-FSD [53]	-	-	-	-	-	-	-	-	-	-	-	-	11.3	23.0	9.8
Zhang *et al.* [51]	4.4	7.5	4.9	5.6	9.9	5.9	7.2	13.3	7.4	-	-	-	-	-	-
QA-FewDet [13]	4.9	10.3	4.4	7.6	16.1	6.2	8.4	18.0	7.3	9.7	20.3	8.6	11.6	23.9	9.8
DeFRCN [32]	4.8	-	-	8.5	-	-	10.7	-	-	13.6	-	-	16.8	-	-
DeFRCN+TSF [21]	5.0	-	-	8.7	-	-	10.9	-	-	13.6	-	-	16.6	-	-
DetReg [1]	-	-	-	-	-	-	-	-	-	-	-	-	25.0	-	27.6
FCT [15]	5.1	-	-	7.2	-	-	9.8	-	-	12.0	-	-	15.3	-	-
KFSOD [50]	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7	14.6
FSODMC [9]	-	-	-	-	-	-	-	-	-	15.1	27.2	14.6	-	-	-
Meta Faster R-CNN [14]	5.1	10.7	4.3	7.6	16.3	6.2	-	-	-	-	-	-	12.7	25.7	10.8
FCT [15]	5.1	-	-	7.2	-	-	9.8	-	-	12.0	-	-	15.3	-	-
Meta-DETR [48]	7.5	12.5	7.7	-	-	-	13.5	21.7	14.0	15.4	25.0	15.8	19.0	30.5	19.7
AirDet [24]	6.10	11.40	6.04	8.73	16.24	8.35	9.95	19.39	9.09	10.81	20.75	10.27	13.0	23.9	12.4
Methods without re-training															
Fan *et al.*† [8]	4.0	8.5	3.5	5.4	11.6	4.6	5.9	12.5	5.0	6.9	14.3	6.0	7.6	15.4	6.8
QA-FewDet [13]	5.1	10.5	4.5	7.8	16.4	6.6	8.6	17.7	7.5	9.5	19.3	8.5	10.2	20.4	9.0
Meta Faster R-CNN [14]	5.0	10.2	4.6	7.0	13.5	6.4	-	-	-	-	-	-	9.7	18.5	9.0
AirDet [24]	5.97	10.52	5.98	6.58	12.02	6.33	7.00	12.95	6.71	7.76	14.28	7.31	8.7	15.3	8.8
FS-DETR (Ours)	**7.0**	**13.6**	**7.5**	**8.9**	**17.5**	**9.0**	**10.0**	**18.8**	**10.0**	**10.9**	**20.7**	**10.8**	**11.3**	**21.7**	**11.1**

Table 3. FSOD performance (AP50) on the PASCAL VOC dataset Novel Set 1 for various template construction configurations. † - result produced using bounding-box jittering for the patch extraction.

Resolution	Pool. type	Novel Set 1				
		1	2	3	5	10
128	global.avg.	42.9	46.0	49.4	50.5	54.0
128	attn.	45.0	48.5	51.5	52.7	56.1
128†	attn.	39.0	42.8	44.6	46.4	50.3
96	attn.	43.2	45.7	49.0	50.1	52.9
192	attn.	45.1	48.3	51.0	52.9	57.0

Table 4. FSOD performance (AP50) on the PASCAL VOC dataset on the Novel Set 1 for models with and without pre-training.

Pre-training	Novel Set 1				
	1	2	3	5	10
	19.0	21.1	23.3	24.0	24.6
✓	45.0	48.5	51.5	52.7	56.1

additional losses, suggesting that the pseudo-classification loss alone suffices for guiding the network.

Impact on individual components: Herein, we analyse the accuracy improvement obtained by two components of FS-DETR namely the MHCA layer in FS-DETR's encoder (see Eq. 5), and the type-specific MLPs (TS-MLP) in FS-DETR's decoder (see Eq. 9). As Table 5 shows, while our system, without both components, provides satisfactory results, unsurprisingly, the addition of TS-MLP further boosts the accuracy. This is expected as the information carried by the object queries and template tokens is semantically different, so ideally they should be transformed using different functions. Finally, the MHCA within the encoder injects template-related information early on to filter or highlight certain image regions, and also helps increase the accuracy.

Table 5. Impact of various components on the FSOD performance (AP50) on the PASCAL VOC dataset (Novel Set 1).

Approach	Novel Set 1				
	1	2	3	5	10
FS-DETR w/o TS-MLP	42.2	46.9	48.3	49.2	51.6
FS-DETR w/o MHCA of Eq. 4	38.1	40.6	41.7	42.2	45.6
FS-DETR	45.0	48.5	51.5	52.7	56.1

6. Conclusions

In this work we propose FS-DETR, a novel transformer based few-shot architecture, that is simple, powerful and flexible. FS-DETR outperforms all prior training-free methods, thus achieving a new state-of-the-art. In addition to the outstanding results presented in Sec. 4, the proposed method can simultaneously predict arbitrary number of classes, using variable-shots per class, in a single forward pass. These results, in combination with the methods formulation, clearly demonstrate not only its performance improvements but also its high flexibility. Therefore, the visual prompting framework proposed FS-DETR can uniquely satisfy the outlined FSOD system *desiderata* (a) and (b), while also making big improvements toward satisfying (c).

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In *CVPR*, pages 14605–14615, 2022. 2, 6, 8

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 4, 5

[4] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-DETR: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 2, 6, 7

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 5, 7

[6] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 5

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[8] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-RPN and multi-relation detector. In *CVPR*, 2020. 3, 6, 7, 8

[9] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with model calibration. In *ECCV*, 2022. 3, 6, 8

[10] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *CVPR*, 2021. 3

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[12] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *CVPR*, 2019. 2

[13] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *ICCV*, 2021. 2, 3, 5, 6, 7, 8

[14] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster R-CNN: Towards accurate few-shot object detection with attentive feature alignment. In *AAAI*, volume 36, pages 780–789, 2022. 3, 6, 7, 8

[15] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *CVPR*, pages 5321–5330, 2022. 3, 6, 7, 8

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[17] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NuerIPS*, 2019. 3

[18] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, 2021. 6, 7, 8

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 2

[20] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019. 5, 6, 7, 8

[21] Jinxiang Lai, Siqian Yang, Wenlong Liu, Yi Zeng, Zhongyi Huang, Wenlong Wu, Jun Liu, Bin-Bin Gao, and Chengjie Wang. tSF: Transformer-Based Semantic Filter for Few-Shot Learning. In *ECCV*, pages 1–19. Springer, 2022. 2, 3, 6, 7, 8

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 2

[23] Aoxue Li and Zhenguo Li. Transformation invariant few-shot object detection. In *CVPR*, 2021. 2, 6, 7, 8

[24] Bowen Li, Chen Wang, Pranay Reddy, Seungchan Kim, and Sebastian Scherer. Airdet: Few-shot detection without fine-tuning for autonomous exploration. In *ECCV*, pages 427–444. Springer, 2022. 1, 2, 3, 6, 8

[25] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *CVPR*, 2021. 3, 6, 7

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014. 5

[28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and

predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 4

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[31] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *ICCV*, 2021. 2

[32] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. DeFRCN: Decoupled Faster R-CNN for few-shot object detection. In *ICCV*, 2021. 1, 3, 6, 7, 8

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. 2

[35] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 5

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017. 3

[37] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: Few-shot object detection via contrastive proposal encoding. In *CVPR*, 2021. 1, 3

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 5

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2, 5

[41] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 29, 2016. 3

[42] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020. 2, 5, 6, 7, 8

[43] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019. 2, 7, 8

[44] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *ICCV*, 2021. 1, 3

[45] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *ECCV*, 2020. 7

[46] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. 2, 6, 7, 8

[47] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta R-CNN: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019. 2, 7, 8

[48] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P Xing. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3, 6, 7, 8

[49] Shan Zhang, Naila Murray, Lei Wang, and Piotr Koniusz. Time-rEversed diffusioN tEnsor Transformer: A New TENET of Few-Shot Object Detection. In *ECCV*, 2022. 3, 6, 7

[50] Shan Zhang, Lei Wang, Naila Murray, and Piotr Koniusz. Kernelized few-shot object detection with efficient integral aggregation. In *CVPR*, pages 19207–19216, June 2022. 3, 6, 7, 8

[51] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *CVPR*, 2021. 3, 6, 7, 8

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[53] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *CVPR*, 2021. 3, 6, 7, 8

[54] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2