Meta-ZSDETR: Zero-shot DETR with Meta-learning

Lu Zhang[1] Chenbo Zhang[1] Jiajia Zhao[2] Jihong Guan[3] Shuigeng Zhou[1]*
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, China
[2]Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, China
[3]Department of Computer Science & Technology, Tongji University, China
{l_zhang19,sgzhou}@fudan.edu.cn, cbzhang21@m.fudan.edu.cn
zhaojiajia1982@gmail.com, jhguan@tongji.edu.cn

Abstract

Zero-shot object detection aims to localize and recognize objects of unseen classes. Most of existing works face two problems: the low recall of RPN in unseen classes and the confusion of unseen classes with background. In this paper, we present the first method that combines DETR and meta-learning to perform zero-shot object detection, named Meta-ZSDETR, where model training is formalized as an individual episode based meta-learning task. Different from Faster R-CNN based methods that firstly generate class-agnostic proposals, and then classify them with visual-semantic alignment module, Meta-ZSDETR directly predict class-specific boxes with class-specific queries and further filter them with the predicted accuracy from classification head. The model is optimized with meta-contrastive learning, which contains a regression head to generate the coordinates of class-specific boxes, a classification head to predict the accuracy of generated boxes, and a contrastive head that utilizes the proposed contrastive-reconstruction loss to further separate different classes in visual space. We conduct extensive experiments on two benchmark datasets MS COCO and PASCAL VOC. Experimental results show that our method outperforms the existing ZSD methods by a large margin.

1. Introduction

Object detection [34] is one of the most fundamental tasks in computer vision. Most existing object detection methods require huge amounts of annotated training data, which is expensive and time-consuming to acquire. Meanwhile, in reality novel categories constantly emerge, and there is seriously lack or even nonexistent of visual data of those novel categories for model training, such as endangered species in the wild. The above issues motivates the investigation of zero-shot object detection, which aims to localize and recognize objects of unseen classes.

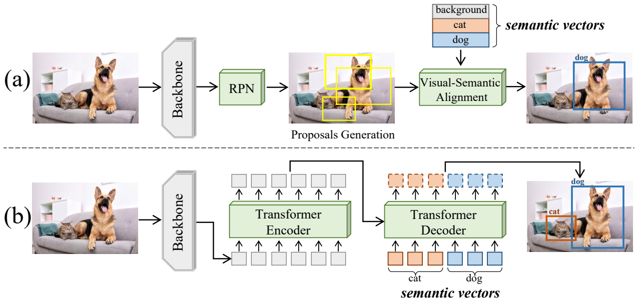


Figure 1. Zero-shot object detection: Faster R-CNN based methods vs. Meta-ZSDETR. (a): Faster R-CNN based methods firstly generate class-agnostic proposals, and then classify them with different visual-semantic alignment modules. (b): Meta-ZSDETR directly predict class-specific boxes with class-specific queries and further filter them with classification head.

A mainstream framework of the existing works that are based on Faster R-CNN, is illustrated in Fig. 1(a), where the RPN remains unchanged and the RoI classification head is replaced with different visual-semantic alignment modules, such as mapping to the same embedding space to calculate similarity between proposals and semantic vectors [2, 40, 41, 32, 44, 12, 25, 18, 7], synthesizing visual features from semantic vectors [17, 14, 42, 47, 27] etc.

However, we observe that the existing methods are suboptimal, due to their obvious inherent shortcomings: i) The proposals from RPN are often not reliable enough to cover all unseen classes objects in an image because of lacking training data, which has also been identified by a recent study [19]. ii) The confusion between background and un-

*correspondence author

seen classes is an intractable problem. Although many previous works have tried to tackle it [46, 14, 2], the results are still unsatisfactory.

Recently, object detection frameworks based on the Transformer have gained widespread popularity, such as DETR [5], Deformable DETR [48], etc. Such architectures are RPN-free and background-free, i.e., they do not involve RPN and background class, which are naturally conducive to building zero-shot object detection methods. However, how to build a ZSD method based on DETR detectors poses new challenges. An intuitive idea is to replace DETR's classification head with a zero-shot classifier based on cosine similarity [43]. However, such a method simply treats DETR as a large RPN for proposals generation, the overall framework is essentially the same as previous works.

In this paper, we present the *first* method that fully explores DETR detectors and meta-learning to perform zero-shot object detection, named **Meta-ZSDETR**, which can solve the two problems mentioned above that have plagued the field of ZSD for many years, and achieves the state-of-the-art performance. The comparison of Meta-ZSDETR with previous methods is shown in Fig. 1. Different from the previous works that firstly generate class-agnostic proposals and then classify them with visual-semantic alignment module, our method utilizes semantic vectors to guide both proposal generation and classification, which greatly improves the recall of unseen classes. Meanwhile, there is no background class in DETR detectors, which means the confusion between background and unseen classes is no more existent.

In order to detect unseen classes, we formalize the training process as an individual episode based meta-learning task. In each episode, we randomly sample an image I and a set of classes $\mathcal{C}_\pi$, which contains the positive classes that appear in I and negative classes that do not appear. The meta-learning task is to make the model learn to detect all positive classes of $\mathcal{C}_\pi$ on image I. Through the meta-learning task, the training and testing can be unified, i.e., in the model testing, we need only to employ the unseen classes as the set $\mathcal{C}_\pi$. To enable the model to detect an arbitrary class set, we firstly fuse each object query with a projected semantic vector from the class set $\mathcal{C}_\pi$, which transfers the query from class-agnostic to class-specific. Then, the decoder takes the class-specific query as input and predicts the locations of class-specific boxes, together with the probabilities that the boxes belong to the fused class. To achieve the above goal, we propose meta-contrastive learning, where all predictions are split into three different types and different combinations of them are chosen to optimize three different heads, i.e., the regression head to generate the locations of class-specific boxes, the classification head to predict the accuracy of generated boxes, and the contrastive head to separate different classes in visual space for performance improving with a contrastive-reconstruction loss. The bipartite matching and loss calculation are performed in a class-by-class manner, and the final loss is averaged over all classes in the sampled class set $\mathcal{C}_\pi$.

In summary, our major contributions are as follows:

- We present the first method that explores DETR and meta-learning to perform zero-shot object detection, which formalizes the training as an individual episode based meta-learning task and ingeniously tackles the two problems that plague ZSD for years.
- We propose to train the decoder to directly predict class-specific boxes with class-specific queries as input, under the supervision of our meta-contrastive learning that contains three different heads.
- We conduct extensive experiments on two benchmark datasets MSCOCO and PASCAL VOC to evaluate the proposed method Meta-ZSDETR. Experimental results show that our method outperforms the existing ZSD methods.

2. Related work

2.1. Zero-shot learning

Zero-shot learning (ZSL) aims to classify images of unseen classes that do not appear during training. There are two main streams in ZSL: embedding based methods and generative based methods. The key idea of embedding based methods is to learn an embedding function that maps the semantic vectors and visual features into the same embedding space, where the visual features and semantic vectors can be compared directly [1, 4, 10, 21, 37]. Generative based methods aim to synthesize unseen visual features with variational autoencoder [20] and generative adversarial networks [39], which convert the ZSL into a fully supervised way [6, 13, 36, 38].

2.2. Zero-shot object detection

Zero-shot object detection (ZSD) has received a great deal of research interest in recent years. Most of ZSD methods are built on Faster R-CNN [11], YOLO [33] and RetinaNet [23]. The process of these methods can be summarized as: generating class-agnostic proposals and classifying proposals into seen/unseen and background classes. The main difference of these methods is that different visual-semantic alignment methods are used to complete the classification of proposals. These methods can be divided into two categories: mapping the semantic vectors and visual features to the same embedding space to calculate similarity [2, 40, 41, 32, 44, 12, 25, 18, 7] and synthesizing visual features from semantic vectors [17, 14, 42, 47, 27]. Although previous works have paid great efforts, there are some problems that still have no satisfactory solution, such as the low recall of class-agnostic RPN for unseen classes and the confusion between background and unseen classes. These problems may be caused by the incompatibility of ZSD task and proposals-based architecture such as Faster R-CNN.

Different from previous works, Meta-ZSDETR is the first work built on Deformable DETR with meta-learning, where the semantic vectors are guided for class-specific boxes generation, instead of class-agnostic proposals in previous works, resulting in a higher unseen recall and precision. Meanwhile, since there is no background class in DETR detectors, the confusion between background and unseen classes is non-existent.

3. Method

3.1. Problem definition

Zero-shot object detection (ZSD) aims to detect objects of unseen classes with model trained on the seen classes. Formally, the class space $\mathcal{C}$ in ZSD is divided into seen classes $\mathcal{C}^s$ and unseen classes $\mathcal{C}^u$, where $\mathcal{C} = \mathcal{C}^s \cup \mathcal{C}^u$ and $\mathcal{C}^s \cap \mathcal{C}^u = \varnothing$. The training set contains objects of seen classes, where each image I is provided with ground-truth class labels and bounding boxes coordinates. While the test set may contain only unseen objects (ZSD setting) or both seen and unseen classes (GZSD setting). During the training and testing, the semantic vectors $\mathcal{W} = \{\mathcal{W}^s, \mathcal{W}^u\}$ is provided for both seen and unseen classes.

3.2. Revisit standard DETR in object detection

To begin with, we review the pipeline of a standard DETR in generic object detection, which contains the following steps: set prediction, optimal bipartite matching and loss calculation.

3.2.1 Set prediction

For an image I, the global representation x_I is extracted by the backbone f_ϕ and Transformer encoder g_ψ successively, which can be expressed as:

$$x_I = g_\psi(f_\phi(I)) \tag{1}$$

Then, the decoder g_θ infers N object predictions $\hat{\mathcal{Y}}$, where N is determined by the number of object queries $\mathcal{Q}$ that serve as learnable positional embedding:

$$\hat{\mathcal{Y}} = g_\theta(x_I, \mathcal{Q}) \tag{2}$$

where $\hat{\mathcal{Y}} = \{(\hat{c}_i, \hat{b}_i)\}_{i=1}^N$ and $\mathcal{Q} = \{q_i\}_{i=1}^N$. For each object query q_i, the decoder g_θ will output a prediction box, which contains two parts: the predicted class $\hat{c}_i$ and predicted box location $\hat{b}_i$.

3.2.2 Optimal bipartite matching

The optimal bipartite matching is to find the minimal-cost matching between the predictions $\hat{\mathcal{Y}} = \{(\hat{c}_i, \hat{b}_i)\}_{i=1}^N$ and ground-truth boxes $\mathcal{Y} = \{(c_i, b_i)\}_{i=1}^N$ (padded with no object $\varnothing$). Therefore, we search for a permutation of N elements $\sigma \in \mathfrak{S}_N$ with lowest cost:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i=1}^{N} \left[\mathcal{L}_{cls}(c_i, \hat{c}_{\sigma_i}) + \mathcal{L}_{loc}(b_i, \hat{b}_{\sigma_i}) \right] \tag{3}$$

where $\mathcal{L}_{cls}(c_i, \hat{c}_{\sigma_i})$ and $\mathcal{L}_{loc}(b_i, \hat{b}_{\sigma_i})$ are matching cost for class prediction and box location with index σ_i, respectively. Bipartite matching produces one-to-one assignments, where each prediction $(\hat{c}_i, \hat{b}_i)$ is assigned to either a ground-truth box (c_i, b_i) or $\varnothing$ (no object). The permutation for lowest cost is calculated with Hungarian algorithm.

3.2.3 Hungarian loss

Hungarian loss is a widely used loss function in DETR, which takes the following form:

$$\mathcal{L}_{Hug} = \sum_{i=1}^{N} \left[\mathcal{L}_{cls}(c_i, \hat{c}_{\hat{\sigma}_i}) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{loc}(b_i, \hat{b}_{\hat{\sigma}_i}) \right] \tag{4}$$

where $\hat{\sigma}$ is the optimal assignment computed in Eq.(3). $\mathcal{L}_{cls}$ is the loss for classification, which usually takes the form of focal loss [23] or cross-entropy loss. $\mathcal{L}_{loc}$ is the location loss and usually contains l_1 loss and GIoU loss [35].

Challenge: Since the standard DETR can only locate the boxes and predict the classes of objects in training set, it is unable to detect unseen classes. In this paper, we utilize the meta-learning to make the model learn to detect objects according to the inputed semantic vectors, so that the model has the ability to detect objects of any category, as long as the semantic vector of the corresponding category is input.

3.3. Framework

We present the framework of Meta-ZSDETR in Fig. 2, which is based on Deformable DETR. Meta-ZSDETR follows the paradigm of meta-learning. The training is performed by episode based meta-learning task. In each episode, we randomly sample an image I and a class set $\mathcal{C}_\pi$. The meta-learning task of each episode is to make the model learn to detect all appeared classes in $\mathcal{C}_\pi$ on image I. Specifically, the image feature is firstly extracted by backbone and Transformer encoder as in Eq.(1). In order for the decoder to detect categories in $\mathcal{C}_\pi$, we add the projected semantic vectors of classes $\mathcal{C}_\pi$ to the object queries, making the queries class-specific. Then, the decoder takes the queries as input and predicts the class-specific boxes directly. To achieve this, the model is optimized with meta-contrastive learning, which contains a regression head to generate the

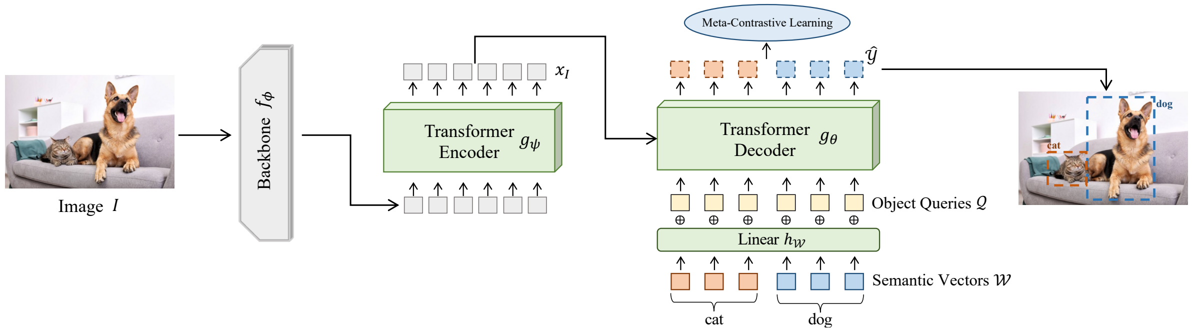

Meta-Contrastive Learning
$\hat{y}$
x_I
Image I
Backbone f_ϕ
Transformer Encoder g_ψ
Transformer Decoder g_θ
Object Queries $\mathcal{Q}$
Linear $h_\mathcal{W}$
Semantic Vectors $\mathcal{W}$
cat
dog
cat
dog

Figure 2. The framework of Meta-ZSDETR. In each episode, a class set $\mathcal{C}_\pi$ and an image I are sampled. The meta-learning task is to make the model learn to detect all appeared classes in $\mathcal{C}_\pi$. Firstly, the image feature x_I is extracted with backbone and encoder. Then, the projected semantic vectors are added to object queries, making them class-specific. Finally, the decoder g_θ will takes the queries as input and directly predict class-specific boxes. To achieve the this, we train our model with proposed meta-contrastive learning.

coordinates of class-specific boxes, a classification to predict the accuracy of generated boxes and a contrastive head that utilize the proposed contrastive-reconstruction loss to further separate different classes in visual space.

3.4. Meta-ZSDETR with class-specific queries

To enable the model to detect any unseen class, we fuse the object queries with class semantic information, and make the model learn to predict the bounding boxes for the fused classes. Such a process is carried out in each meta-learning task.

Specifically, in each episode, we randomly sample a class set $\mathcal{C}_\pi$ and an image I, where the $\mathcal{C}_\pi$ satisfies $\mathcal{C}_\pi \subseteq \mathcal{C}^s$ and each element is unique. Meanwhile, $\mathcal{C}_\pi = \mathcal{C}_\pi^+ \cup \mathcal{C}_\pi^-$, where $\mathcal{C}_\pi^+$ is the positive classes that appeared in image I and $\mathcal{C}_\pi^-$ is the randomly sampled negative classes that do not appear in I for contrast. Meanwhile, we denote the size of $\mathcal{C}_\pi$ as $L(\mathcal{C}_\pi)$, where $L(\cdot)$ is the operation that calculate size. The positive rate $\frac{L(\mathcal{C}_\pi^+)}{L(\mathcal{C}_\pi)}$ is set to λ_π, which is a hyper-parameter.

Then, the corresponding semantic vectors $\mathcal{W}_\pi$ of class set $\mathcal{C}_\pi$ are projected from semantic space to the visual space with a linear layer $h_\mathcal{W}$:

$$\widetilde{\mathcal{W}_\pi} = h_\mathcal{W}(\mathcal{W}_\pi) \tag{5}$$

where $\widetilde{\mathcal{W}_\pi}$ is the projected semantic vectors of class set $\mathcal{C}_\pi$. Since $L(\widetilde{\mathcal{W}_\pi}) \ll N$, i.e. the number of semantic vectors is smaller than the number of object queries $\mathcal{Q}$, we expand $\widetilde{\mathcal{W}_\pi}$ by duplicating each element in $\widetilde{\mathcal{W}_\pi}$ for T times, which satisfies $L(\widetilde{\mathcal{W}_\pi}) \cdot T \geqslant N$ and $L(\widetilde{\mathcal{W}_\pi}) \cdot (T-1) < N$. For redundant elements more than N, we drop them.

Then, the projected semantic vectors $\widetilde{\mathcal{W}_\pi}$ is added to object queries $\mathcal{Q}$ as follows:

$$\mathcal{Q}_\pi = \mathcal{Q} \oplus \widetilde{\mathcal{W}_\pi} \tag{6}$$

where $\mathcal{Q}_\pi = \{q_i^\pi\}_{i=1}^N$ is the class-specific queries that will be inputed into the Transformer decoder g_θ with image feature x_I to generate predictions:

$$\hat{\mathcal{Y}} = g_\theta(x_I, \mathcal{Q}_\pi) \tag{7}$$

$\hat{\mathcal{Y}} = \{(\hat{\delta}_i, \hat{b}_i)\}_{i=1}^N$ is the set of predictions, where $\hat{b}_i$ is predicted box location generated with query q_i^π and $\hat{\delta}_i$ is the probability of box $\hat{b}_i$ belonging to the fused class, i.e. the class of semantic vector that fused to query q_i^π. Meanwhile, different from the standard DETR that the classification head determines the class of predicted boxes, the class of predicted box $\hat{b}_i$ in Meta-ZSDETR is class-specific and is determined by the class of corresponding query and $\hat{\delta}_i$ only has one dimension to represent the probability of $\hat{b}_i$ belongs to the fused class.

3.5. Meta-contrastive learning

In order for the regression head of decoder g_θ to generate more accurate class-specific box coordinate $\hat{b}_i$ and classification head to have a stronger discriminative ability of further judging the location accuracy of generated $\hat{b}_i$, we propose the meta-contrastive learning to train the heads of decoder g_θ, i.e. a regression head to generate class-specific boxes, a classification head to filter inaccurate boxes, and moreover a contrastive head to further separate different classes in visual space, which will improve the performance of both seen classes and unseen classes.

Meta-contrastive learning performs the matching and optimization in a class-by-class manner. As shown in Fig. 3, for each class $c_j^\pi \in \mathcal{C}_\pi$, the decoder takes queries of class c_j^π as input, and the predictions are split into three types: 1) The positive predictions that are assigned to GT box of class c_j^π by class-specific bipartite matching. 2) The negative predictions that are assigned to any other classes than c_j^π. 3) The negative predictions that belong to background.

Then, we takes different combinations of three types

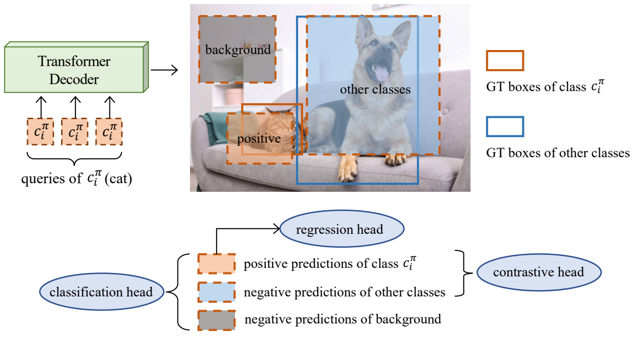

Transformer Decoder
c_i^π
c_i^π
c_i^π
queries of c_i^π (cat)
background
other classes
positive
GT boxes of class c_i^π
GT boxes of other classes
regression head
classification head
positive predictions of class c_i^π
negative predictions of other classes
negative predictions of background
contrastive head

Figure 3. All predictions of class c_j^π are split into three different types and different combinations of them are utilized to train three different heads.

of predictions to train three heads: a) For the classification head, we take all predictions for the training and give them corresponding positive/negative class-specific targets for class c_j^π. b) For the regression head, since the regression head aims to generate class-specific boxes of class c_j^π, we only utilize the positive predictions for optimization. c) For the contrastive head, since our intention is to separate different classes in visual space, we only use positive predictions of class c_j^π and negative predictions of other classes for contrastive-reconstruction loss.

3.5.1 Class-specific bipartite matching

Our meta-contrastive learning performs the matching in a class-by-class manner. For class $c_j^\pi \in \mathcal{C}_\pi$, the predictions generated by queries of class c_j^π are selected for bipartite matching, which is denoted as $\hat{\mathcal{Y}}_{c_j^\pi} = \{(\hat{\delta}_{\tau_i}, \hat{b}_{\tau_i})\}_{i=1}^{T_\tau}$, where $\{\tau_i\}_{i=1}^{T_\tau}$ are the indexes for queries of class c_j^π in N queries. Since each query is duplicated for T or $T-1$ times, we denote the length uniformly as T_τ.

Then, the ground-truth matching targets are revised for each class $c_j^\pi \in \mathcal{C}_\pi$. The class-specific matching labels are denoted as $\Delta_{c_j^\pi} = \{\delta_i^{c_j^\pi}\}_{i=1}^{T_\tau}$ (padded with no objects $\varnothing$), which satisfy:

$$\delta_i^{c_j^\pi} = \begin{cases} 1, & c_i = c_j^\pi \\ 0, & c_i \neq c_j^\pi \end{cases} \tag{8}$$

where c_i is the origin class label for ground-truth box b_i. The revised matching labels are generated according to whether c_i is equal to c_j^π. Then, the matching targets $\mathcal{Y}_{c_j^\pi} = \{(\delta_i^{c_j^\pi}, b_i)\}_{i=1}^{T_\tau}$ is utilized for bipartite matching of class c_j^π, where a permutation of T_τ elements $\sigma \in \mathfrak{S}_{T_\tau}$ with lowest cost is search as:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_{T_\tau}}{\arg\min} \sum_{i=1}^{T_\tau} \left[\mathcal{L}_{cls}(\delta_i^{c_j^\pi}, \hat{\delta}_{\tau_{\sigma_i}}) + \mathcal{L}_{loc}(b_i, \hat{b}_{\tau_{\sigma_i}}) \right] \tag{9}$$

where $\hat{\sigma}$ is the optimal assignment between matching targets $\mathcal{Y}_{c_j^\pi}$ and predictions $\hat{\mathcal{Y}}_{c_j^\pi}$. $\mathcal{L}_{cls}$ and $\mathcal{L}_{loc}$ are the same as that in Eq.(3).

3.5.2 Loss function

Based on $\hat{\sigma}$, predictions in $\hat{\mathcal{Y}}_{c_j^\pi}$ are split into three types mentioned above, i.e. positive and two types of negative ones, and loss function of class c_j^π for three heads are calculated as follows:

$$\mathcal{L}_{c_j^\pi} = \sum_{i=1}^{T_\tau} \left[\mathcal{L}_{cls}(\delta_i^{c_j^\pi}, \hat{\delta}_{\tau_{\hat{\sigma}_i}}) + \mathbb{1}_{(c_i = c_j^\pi)} \mathcal{L}_{loc}(b_i, \hat{b}_{\tau_{\hat{\sigma}_i}}) \right] + \mathcal{L}_{cont} \tag{10}$$

where the loss $\mathcal{L}_{c_j^\pi}$ for class c_j^π is composed of classification loss $\mathcal{L}_{cls}$, regression loss $\mathcal{L}_{loc}$ and contrastive-reconstruction loss $\mathcal{L}_{cont}$.

Classification loss. $\mathcal{L}_{cls}$ takes all predictions as input and makes the model learn to distinguish whether a predicted box belongs to c_j^π. The predicted box has the label $\delta_i^{c_j^\pi} = 1$ if and only if it is assigned to a ground-truth box with class label c_j^π. We implement $\mathcal{L}_{cls}$ with focal loss [23].

Regression loss. Since our intention is to generate class-specific boxes, i.e. input decoder with query of class c_j^π and output box of class c_j^π, we only select the ground-truth box with class label c_j^π for optimization, i.e. $\delta_i^{c_j^\pi} = 1$, making the predicted boxes to be closer to GT boxes with class label c_j^π. The $\mathcal{L}_{loc}$ is implement with l_1 loss and GIoU loss [35].

Contrastive-reconstruction loss. In ZSD tasks, the original visual space is not well-structured due to the lack of discriminative information, many previous works solved it by introducing a reconstruction loss to guide the distribution of visual features. Here, we combine the reconstruction loss and contrastive loss to bring a higher intra-class compactness and inter-class separability of the visual structure.

In particular, we project the last hidden features of decoder to the semantic space, where the projected hidden features of positive predictions are constraint to be as close as possible to ω_j^π, which is the semantic vector of class c_j^π, and the negative ones are constraint to be as far as possible to ω_j^π. Formally, we denote the last hidden feature of optimal box $(\hat{\delta}_{\tau_{\hat{\sigma}_i}}, \hat{b}_{\tau_{\hat{\sigma}_i}})$ that matched to GT box (c_i, b_i) as $z_{\tau_{\hat{\sigma}_k}}$ and the $\mathcal{L}_{cont}$ is formulated as:

$$\mathcal{L}_{cont} = \frac{1}{N_{pos}} \sum_{i=1}^{T_\tau} \mathbb{1}_{(c_i = c_j^\pi)} \mathcal{L}_{cont}(z_{\tau_{\hat{\sigma}_i}}) \tag{11}$$

$$\mathcal{L}_{cont}(z_{\tau_{\hat{\sigma}_i}}) = -\log \frac{\exp[h_\rho(z_{\tau_{\hat{\sigma}_i}}) \cdot \omega_j^\pi / \kappa]}{\sum_{k=1}^{T_\tau} \mathbb{1}_{(c_k \neq \varnothing)} \exp[h_\rho(z_{\tau_{\hat{\sigma}_k}}) \cdot \omega_j^\pi / \kappa]} \tag{12}$$

where h_ρ is a linear layer that project the hidden feature to the semantic space. N_{pos} is the number of positive predictions of class c_j^π. κ is a temperature hyper-parameter as in InfoNCE [28]. The optimization of the above loss function increases the instance-level similarity between projected hidden features of positive predictions with semantic vector ω_j^π and space the negative ones. As a result, visual features of the same class will form a tighter cluster.

Total loss function. We compute the loss function with Eq.(10) for each class $c_j^\pi \in \mathcal{C}_\pi$, separately. For negative classes $\mathcal{C}_\pi^-$, only classification loss is calculated. Finally, the loss of current episode is averaged over all classes in sampled class set $\mathcal{C}_\pi$:

$$\mathcal{L} = \frac{1}{L(\mathcal{C}_\pi)} \sum_{j=1}^{L(\mathcal{C}_\pi)} \mathcal{L}_{c_j^\pi} \tag{13}$$

where $L(\mathcal{C}_\pi)$ is the number of classes in $\mathcal{C}_\pi$. We utilize $\mathcal{L}$ for model optimization.

4. Experiments

4.1. Datasets and splits

Following previous works [40, 17], we takes two benchmark datasets for evaluation: PASCAL VOC 2007+2012 [9] and MS COCO 2014 [24].

Datasets: PASCAL VOC contains 20 classes of objects for object detection. More specifically, the PASCAL VOC 2,007 is composed of 2,501 training images, 2,510 validation images, and 5,011 test images. The PASCAL VOC 2012 dataset comprises 5,717 training images and 5,823 validation images, without test images released. MS COCO 2014 is a benchmark dataset designed for object detection and semantic segmentation tasks, which contains 82,783 training and 40,504 validation images from 80 categories. For PASCAL VOC and MS COCO, we adopt the FastText [26] to extract the semantic vectors following [14, 17].

Seen/unseen splits: All splits follow the previous setting [40, 17]. For PASCAL VOC, we use a 16/4 split proposed in [7]. For MS COCO, we follow the same procedures described in [2, 29, 17] to take 2 different splits, i.e. 48/17 and 65/15. For all datasets, the images that contains unseen classes in the training set are removed to guarantee that unseen objects will not be available during training.

4.2. Evaluation protocols

We adopt the widely-used evaluation protocols proposed in [2, 7]. For PASCAL VOC, mAP with IoU threshold 0.5 is used to evaluate the performance. For MS COCO, mAP and recall@100 with three different IoU threshold (i.e. 0.4,0.5 and 0.6) are utilized for evaluation. For GZSD setting that contains both seen and unseen classes, the performance is evaluated by Harmonic Mean (HM).

4.3. Implementation details

We build Meta-ZSDETR on Deformable DETR [48] with ResNet-50 [16] as backbone. The number of queries N is set to 900. For the sampled class set $\mathcal{C}_\pi$, the positive classes consist of the classes that appear in the image and the negative classes is sampled from $\mathcal{C}^s$. The positive rate λ_π is set to 0.5. The number of Transformer encoder layers and decoder layers is set to 6. The temperature hyperparameter κ in Eq.(12) is set to 0.2. We train our model for total 500,000 iterations with batch size 16, i.e. each iteration contains 16 episodes in parallel. Following Deformable DETR, different coefficients are utilized to weight different loss functions, where 1.0 is used for classification loss, 5.0 is used for l_1 loss of regression head, 2.0 is used for GIoU loss of regression head and 1.0 is used for contrastive-reconstruction loss. More details can refer to our code.

Method	ZSD	GZSD		
		S	U	HM
SAN [31]	59.1	48.0	37.0	41.8
HRE [8]	54.2	62.4	25.5	36.2
PL [30]	62.1	-	-	-
BLC [45]	55.2	58.2	22.9	32.9
SU [15]	64.9	-	-	-
Robust-Syn [17]	65.5	47.1	49.1	48.1
ContrastZSD [40]	65.7	63.2	46.5	53.6
Meta-ZSDETR	**70.3**	**67.6**	**56.3**	**61.4**

Table 1. The results of mAP in PASCAL VOC with IoU=0.5 under ZSD and GZSD settings. Here, “S” denotes seen classes, “U” denotes unseen classes and “HM” denotes harmonic mean.

Method	car	dog	sofa	train	mAP
SAN [31]	56.2	85.3	62.6	26.4	57.6
HRE [8]	55.0	82.0	55.0	26.0	54.5
PL [30]	63.7	87.2	53.2	44.1	62.1
BLC [45]	43.7	86	60.8	30.1	55.2
SU [15]	59.6	**92.7**	62.3	45.2	64.9
Robust-Syn [17]	60.1	93.0	59.7	49.1	65.5
ContrastZSD [40]	65.5	86.4	63.1	47.9	65.7
Meta-ZSDETR	**69.0**	92.4	**65.7**	**54.1**	**70.3**

Table 2. Class-wise AP and mAP on unseen classes in PASCAL VOC under ZSD setting.

4.4. Comparison with existing methods

4.4.1 PASCAL VOC

We present the results of PASCAL VOC in Tab. 1, where we can see that our method performs best among all existing methods under both ZSD and GZSD settings, and lift the mAP in PASCAL VOC to a higher level.

Specifically, in ZSD setting, Meta-ZSDETR achieves 70.3 mAP and outperform the second-best model ContrastZSD [40] by a large margin of 4.6 mAP, which is the

Method	Split	Recall@100			mAP
		IoU=0.4	IoU=0.5	IoU=0.6	IoU=0.5
DSES [3]	48/17	40.2	27.2	13.6	0.5
TD [22]	48/17	45.5	34.3	18.1	-
PL [30]	48/17	-	43.5	-	10.1
BLC [45]	48/17	51.3	48.8	45.0	10.6
ZSDTR [43]	48/17	51.8	48.5	44.5	10.4
Robust-Syn [17]	48/17	58.1	53.5	47.9	13.4
ContrastZSD [40]	48/17	56.1	52.4	47.2	12.5
Meta-ZSDETR	48/17	**62.3**	**59.8**	**54.2**	**15.1**
PL [30]	65/15	-	37.7	-	12.4
BLC [45]	65/15	57.2	54.7	51.2	14.7
SU [15]	65/15	54.4	54.0	47.0	19.0
ZSDTR [43]	65/15	63.8	60.3	56.5	13.2
Robust-Syn [17]	65/15	65.3	62.3	55.9	19.8
ContrastZSD [40]	65/15	62.3	59.5	55.1	18.6
Meta-ZSDETR	65/15	**69.1**	**66.7**	**59.0**	**22.5**

Table 3. ZSD performance of Recall@100 and mAP with different IoU thresholds on MS COCO dataset.

first time to boost the performance of ZSD setting on PASCAL VOC to over 70 mAP.

For GZSD setting, our method also achieves SOTAs in all three metrics, i.e. mAP on seen classes, unseen classes and harmonic mean, which brings about 4.4, 9.8 and 7.8 points improvement, respectively. It is worth noting that the improvement of our method on unseen classes in GZSD setting is extremely large, which proves that our method has a strong generalization on unseen classes, and can alleviate the problem that the unseen classes tend to be misclassified into seen classes to a certain extent. We also report class-wise mAP in ZSD setting in Tab. 2, where our method achieves the best performance on 3 classes.

4.4.2 MS COCO

We perform experiments on MS COCO, where the results of ZSD setting is shown in Tab. 3 and the results of GZSD setting is shown in Tab. 4. We can see that Meta-ZSDETR achieves the best results in all metrics under all settings.

For ZSD setting, we can see that mAP of our method in 48/17 and 65/15 splits outperforms the second-best by a margin of 1.7 and 2.7 mAP, respectively, which demonstrates that our method generalizes well to unseen classes. Meanwhile, we can see that Recall@100 decrease as the IoU increases in all methods. Compared with other methods, Meta-ZSDETR has a smaller drop, which is benefit from that our decoder can generate more accurate boxes with class semantic information as input.

For GZSD setting, our method achieves SOTAs in both seen and unseen classes. The harmonic means of mAP under 48/17 and 65/15 splits are improved from 20.4 to 22.5, and from 26.0 to 29.5, demonstrating the effectiveness and superiority of our method. Meanwhile, the Recall@100 also improves due to the powerful class-specific boxes generation capabilities. We also report the class-wise AP in 65/15 split of MS COCO, which can be found in our supplementary material.

Method	Split	Recall@100			mAP		
		S	U	HM	S	U	HM
PL [30]	48/17	38.2	26.3	31.2	35.9	4.1	7.4
BLC [45]	48/17	57.6	46.4	51.4	42.1	4.5	8.2
ZSDTR [43]	48/17	74.3	48.4	60.5	48.5	5.6	9.5
Robust-Syn [17]	48/17	59.7	58.8	59.2	42.3	13.4	20.4
ContrastZSD [40]	48/17	65.7	52.4	58.3	45.1	6.3	11.1
Meta-ZSDETR	48/17	**74.3**	**59.0**	**65.8**	**48.7**	**14.6**	**22.5**
PL [30]	65/15	36.4	37.2	36.8	34.1	12.4	18.2
BLC [45]	65/15	56.4	51.7	53.9	36.0	13.1	19.2
SU [15]	65/15	57.7	53.9	55.8	36.9	19.0	25.1
ZSDTR [43]	65/15	69.1	59.5	61.1	40.6	13.2	20.2
Robust-Syn [17]	65/15	58.6	61.8	60.2	37.4	19.8	26.0
ContrastZSD [40]	65/15	62.9	58.6	60.7	40.2	16.5	23.4
Meta-ZSDETR	65/15	**71.1**	**65.4**	**68.1**	**45.9**	**21.7**	**29.5**

Table 4. GZSD performance of Recall@100 and mAP with IoU=0.5 on MS COCO dataset.

$\mathcal{L}_{loc}$	$\mathcal{L}_{cls}$	$\mathcal{L}_{cont}$	Seen	Unseen
✓			39.9	14.5
✓	✓		44.8	20.6
✓		✓	40.6	15.3
✓	✓	✓	**45.9**	**21.7**

Table 5. Ablation study of different combinations of loss functions.

4.5. Ablation study

We analyze the effects of various components in Meta-ZSDETR. Unless otherwise specified, the experiments are carried out on MS COCO with 65/15 split under GZSD setting and use mAP with IoU=0.5 as metric.

Effects of different loss functions. Here, we analyze the effects of three loss functions in meta-contrastive learning. We utilize different combinations of regression loss $\mathcal{L}_{loc}$, classification loss $\mathcal{L}_{cls}$ and contrastive-reconstruction loss $\mathcal{L}_{cont}$ to optimize the model, and show the results in Tab. 5. Since the regression loss is necessary, we keep it for all combinations. For model without classification loss $\mathcal{L}_{cls}$, we do not perform the boxes filter and directly use the class-specific boxes predicted from regression head, where the scores are generated randomly. As we can see, if the model is only trained with regression head to gen-

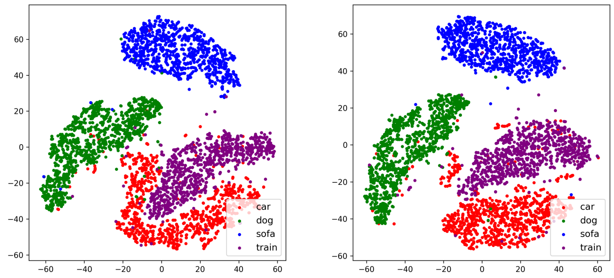
Heads	$\hat{\mathcal{Y}}_{pos}$	$\hat{\mathcal{Y}}_{other}$	$\hat{\mathcal{Y}}_{bg}$	Seen	Unseen
Classification Head	✓	✓		43.7	17.9
	✓		✓	44.8	19.0
	✓	✓	✓	**45.9**	**21.7**
Regression Head	✓			**45.9**	**21.7**
	✓	✓		42.1	16.5
Contrastive Head	✓			45.1	21.0
	✓	✓		**45.9**	**21.7**
	✓	✓	✓	45.4	21.3

Table 6. Ablation study on using different combinations of predictions to train three heads.

erate class-specific boxes, it can achieve a mAP of 14.5 in unseen classes, which is relatively low, but also surpasses many previous methods. Adding the classification loss will greatly boost the performance of unseen classes to 20.6 mAP, which thanks to the powerful discriminative ability of the classification head that can filter inaccurate boxes. Meanwhile, the contrastive-reconstruction loss can improve the performance with and without the classification head about 1 point. Finally, the combination of three losses achieves the best performance.

Study for training of different heads. As describe, based on the class-specific bipartite matching for class c_j^π, all predictions are split into three different types: the positive predictions $\hat{\mathcal{Y}}_{pos}$ assigned to GT boxes of c_j^π, the negative predictions $\hat{\mathcal{Y}}_{other}$ assigned to other classes and the negative predictions $\hat{\mathcal{Y}}_{bg}$ that belong to background. Here, we study different combinations of them to train three heads and the results are shown in Tab. 6. We can see that: 1) For classification head, since it aims to filter all kinds of negative predictions, using all predictions to train it can achieve the best performance. 2) For regression head, if we train it with GT boxes of all classes, i.e. using $\hat{\mathcal{Y}}_{pos}$ and $\hat{\mathcal{Y}}_{other}$, the regression head will degenerate into a class-agnostic RPN, which will greatly reduce the recall of unseen classes, thus lead to a lower mAP. 3) For contrastive head, on one hand, if we only use $\hat{\mathcal{Y}}_{pos}$ for training, it will degenerate into a reconstruction loss, which has been widely used in previous works and it will bring a 0.4 mAP improvement in unseen classes compared with the version without it. On the other hand, compared with using all predictions, removing background predictions will make the contrastive head focus on distinguishing $\hat{\mathcal{Y}}_{other}$ and inputed semantic vectors of class c_j^π, thus brings more improvement.

Visualization for contrastive-reconstruction loss. Here, we study the influence of contrastive-reconstruction loss on visual space by visualizing the distribution of hidden features with t-SNE. We visualize the last hidden features of decoder in unseen classes of PASCAL VOC. The result is shown in Fig. 4. As we can see, our contrastive-reconstruction loss can further separate different classes in visual space and bring a higher intra-class compactness and inter-class separability of the visual structure.



Without Contrastive Head

With Contrastive Head

Figure 4. The t-SNE visualization of the last hidden layer of decoder. We can see that the contrastive head can separate different classes in visual space.

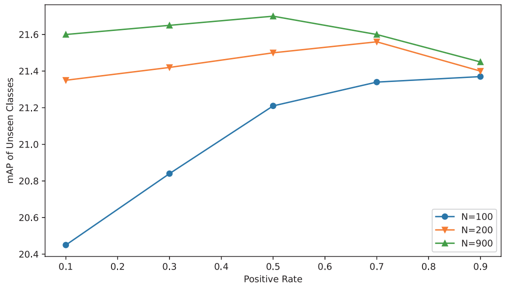


Figure 5. The effect of number of queries and positive rate of $\mathcal{C}_\pi$

Effect of number of queries and positive rate. We study the effect of number of queries N and positive rate λ_π of sampled class set $\mathcal{C}_\pi$. We found that λ_π have different influence under different number of queries N. We change the positive rate λ_π in different settings of N and report the mAP of converged model in unseen classes. In each episode, we control λ_π by sampling different number of negative classes. All models are trained for 500,000 episodes. The results are shown in Fig. 5. As we can see, a larger N tends to have a better performance due to a higher recall, and of course a higher amount of calculation. Meanwhile, when N is small (e.g. 100), a small positive rate will greatly reduce the amount of positive queries for training, thereby reducing the model performance. When N is large (e.g. 900), the number of positive queries is guaranteed and more negative queries are need for the classification head to learn to distinguish among them. Therefore, we can see the best performance is achieved when λ_π is 0.5 and N is 900.

5. Conclusion

In this paper, we present the first work that combine DETR and meta-learning to perform zero-shot object detection, which formalize the training as individual episode based meta-learning task. In each episode, we randomly sample an image and a class set. The meta-learning task is to make the model learn to detect all appeared classes of the sampled class set on the image. To achieve this, we train the decoder to directly predict class-specific boxes with class-specific queries as input, under the supervision of our meta-contrastive learning that contains three different heads. We conduct extensive experiments on the benchmark datasets MSCOCO and PASCAL VOC. Experimental results show that our method outperforms the existing ZSD methods. In the future, we will focus on further performance improvement.

Acknowledgement. Jihong Guan was supported by National Natural Science Foundation of China (NSFC) under grant No. U1936205.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 819–826, 2013. 2

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 1, 2, 6

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018. 7

[4] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 730–746. Springer, 2016. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[6] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 122–131, 2021. 2

[7] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. *arXiv preprint arXiv:1805.06157*, 2018. 1, 2, 6

[8] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. In *BMVC*, 2018. 6

[9] 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*, 88:303–308, 2009. 6

[10] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 584–599. Springer, 2014. 2

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[12] Dikshant Gupta, Aditya Anantharaman, Nehal Mamgain, Vineeth N Balasubramanian, CV Jawahar, et al. A multi-space approach to zero-shot object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1209–1217, 2020. 1, 2

[13] Zongyan Han, Zhenyong Fu, and Jian Yang. Learning the redundancy-free features for generalized zero-shot object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12865–12874, 2020. 2

[14] Nasir Hayat, Munawar Hayat, Shafin Rahman, Salman Khan, Syed Waqas Zamir, and Fahad Shahbaz Khan. Synthesizing the unseen for zero-shot object detection. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1, 2, 6

[15] Nasir Hayat, Munawar Hayat, Shafin Rahman, Salman Khan, Syed Waqas Zamir, and Fahad Shahbaz Khan. Synthesizing the unseen for zero-shot object detection. In *ACCV*, 2020. 6, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] Peiliang Huang, Junwei Han, De Cheng, and Dingwen Zhang. Robust region feature synthesizer for zero-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7622–7631, 2022. 1, 2, 6, 7

[18] Siddhesh Khandelwal, Anirudth Nambirajan, Behjat Siddiquie, Jayan Eledath, and Leonid Sigal. Frustratingly simple but effective zero-shot detection and segmentation: Analysis and a strong baseline. *arXiv preprint arXiv:2302.07319*, 2023. 1, 2

[19] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 7(2):5453–5460, 2022. 1

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[21] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3174–3183, 2017. 2

[22] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *AAAI*, volume 33, pages 8690–8697, 2019. 7

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 3, 5

[24] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[25] Qiaomei Mao, Chong Wang, Shenghao Yu, Ye Zheng, and Yuqi Li. Zero-shot object detection with attributes-based category similarity. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(5):921–925, 2020. 1, 2

[26] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 2017. 6

[27] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 479–495. Springer, 2020. 1, 2

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[29] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11932–11939, 2020. 6

[30] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI*, volume 34, pages 11932–11939, 2020. 6, 7

[31] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*, pages 547–563. Springer, 2018. 6

[32] Shafin Rahman, Salman H Khan, and Fatih Porikli. Zero-shot object detection: Joint recognition and localization of novel concepts. *International Journal of Computer Vision*, 128:2979–2999, 2020. 1, 2

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 3, 5

[36] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019. 2

[37] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 2

[38] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018. 2

[39] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10275–10284, 2019. 2

[40] Caixia Yan, Xiaojun Chang, Minnan Luo, Huan Liu, Xiaoqin Zhang, and Qinghua Zheng. Semantics-guided contrastive network for zero-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 6, 7

[41] Caixia Yan, Qinghua Zheng, Xiaojun Chang, Minnan Luo, Chung-Hsing Yeh, and Alexander G Hauptman. Semantics-preserving graph propagation for zero-shot object detection. *IEEE Transactions on Image Processing*, 29:8163–8176, 2020. 1, 2

[42] Shizhen Zhao, Changxin Gao, Yuanjie Shao, Lerenhan Li, Changqian Yu, Zhong Ji, and Nong Sang. Gtnet: Generative transfer network for zero-shot object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12967–12974, 2020. 1, 2

[43] Ye Zheng and Li Cui. Zero-shot object detection with transformers. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 444–448. IEEE, 2021. 2, 7

[44] Ye Zheng, Ruoran Huang, Chuanqi Han, Xi Huang, and Li Cui. Background learnable cascade for zero-shot object detection. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1, 2

[45] Ye Zheng, Ruoran Huang, Chuanqi Han, Xi Huang, and Li Cui. Background learnable cascade for zero-shot object detection. In *ACCV*, 2020. 6, 7

[46] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2593–2602, 2021. 2

[47] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11693–11702, 2020. 1, 2

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2, 6