# Transductive Learning for Zero-Shot Object Detection

Shafin Rahman*†, Salman Khan‡* and Nick Barnes*†
*Australian National University, †Data61-CSIRO, ‡Inception Institute of AI
firstname.lastname@anu.edu.au

## Abstract

*Zero-shot object detection (ZSD) is a relatively unexplored research problem as compared to the conventional zero-shot recognition task. ZSD aims to detect previously unseen objects during inference. Existing ZSD works suffer from two critical issues: (a) large domain-shift between the source (seen) and target (unseen) domains since the two distributions are highly mismatched. (b) the learned model is biased against unseen classes, therefore in generalized ZSD settings, where both seen and unseen objects co-occur during inference, the learned model tends to misclassify unseen to seen categories. This brings up an important question: How effectively can a transductive setting[1] address the aforementioned problems? To the best of our knowledge, we are the first to propose a transductive zero-shot object detection approach that convincingly reduces the domain-shift and model-bias against unseen classes. Our approach is based on a self-learning mechanism that uses a novel hybrid pseudo-labeling technique. It progressively updates learned model parameters by associating unlabeled data samples to their corresponding classes. During this process, our technique makes sure that knowledge that was previously acquired on the source domain is not forgotten. We report significant 'relative' improvements of 34.9% and 77.1% in terms of mAP and recall rates over the previous best inductive models on MSCOCO dataset.*

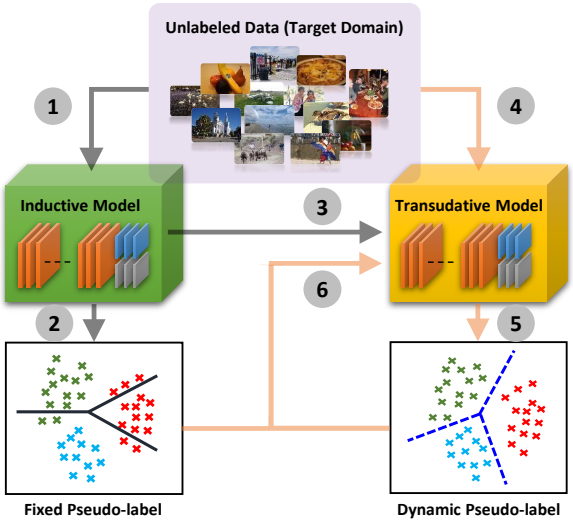


Figure 1: We propose a self-learning approach based on pseudo-labeling for *transductive ZSD*. (1) Unlabeled data is fed to the inductive model to (2) generate fixed pseudo-labels. (3) The Transductive model is initialized with an inductive model. (4) Unlabeled data is fed to the transductive model to generate (5) dynamic labels. (6) Fixed and dynamic labels are then fed to the transductive model to interactively update it (4-5-6). The initial decision boundary of an inductive model (solid black line) is updated to a modified decision boundary (blue dashed line) after transductive learning.

## 1. Introduction

The availability of large-scale annotated datasets and high capacity deep networks have paved the way for rapid progress in supervised learning tasks. As a result, deep CNNs are now performing as well as humans on specialized tasks of visual recognition and fine-grained categorization [17, 33]. However, in several domains, acquiring large-scale annotations is not viable due to the requirement of expert knowledge or simply due to scarcity of visual samples in the real world (e.g., rare species). Zero-shot learning (ZSL) addresses such scenarios where we do not have any visual examples for the unseen classes during training [23, 30]. Traditional ZSL approaches have been limited to recognition (classification) setting.

Zero-shot object detection (ZSD) is a recently-introduced problem that aims to simultaneously locate and categorize unseen object classes. Compared to the recognition task, ZSD is far more challenging due to the ill-posed nature and inherent complexity of localizing totally unseen categories. The problem is compounded when we consider a generalized ZSD setting, which assumes both seen and unseen objects can co-occur during inference. Existing efforts [1, 3, 24, 38, 22] to address the ZSD problem explore

[1]In a transductive ZSD setting, unlabeled test examples are available during model training.

an *inductive* setting, which considers only labeled examples in the source domain for training. In practice, there exists a large domain-gap between the source (seen objects) and target (unseen) domains. To circumvent this gap, a transductive setting for ZSL assumes that a part of the unlabeled target domain samples are available during training.

Given the challenging nature of the ZSD problem, it is of great interest to study how transductive settings can help in dealing with domain-shift [4] and model-bias [2] problems. In this work, we provide the first solution to transductive ZSD and generalized ZSD problems. The transductive learning paradigm allows a method to take advantage of unlabeled test data. The main insight used in our approach is that the learning acquired on the seen classes can be used to resolve ambiguities in the unlabeled target domain images. We progressively assign pseudo-labels to the unlabeled data, which are then used to update model parameters without forgetting the previously acquired learning on the source domain. Fig. 1 illustrates an overview of our approach.

Our main contributions are as follows: **(1)** We propose a single-stage object detector for transductive zero-shot learning that learns to optimally combine semantic and visual domain cues. **(2)** To leverage unlabeled target domain data, our solution introduces a novel pseudo-labeling strategy that dynamically associates unlabeled samples with their respective classes. **(3)** To retain concepts previously learned on the source domain, we propose a fixed pseudo-labeling objective. **(4)** Our experiments demonstrate that the novel pseudo-labeling strategy effectively reduces both domain-shift and model-bias against unseen classes, leading to new state of the art on ZSD. We obtain 3.77% and 20.9% absolute boost in mAP and recall rate which translate to relative gains of 34.9% and 77.1%, respectively, on the challenging MSCOCO dataset.

## 2. Related Work

**Transductive zero-shot learning:** To alleviate the domain shift problem in ZSL, transductive settings have been proposed. Rohrbach *et al*. [27] explored the manifold structure of unseen classes by graph-based label propagation. [5] extended the label propagation with a multi-view hypergraph. Several approaches adopt a joint learning framework to train on labeled and unlabeled data separately [7, 11, 36, 32]. Such training can be in semantic space [7], visual space [11] or a latent space [36, 32]. Few other efforts attempt to refine visual-semantic embeddings iteratively with unlabeled unseen data [35, 12]. A domain-invariant projection is learnt in [37] that maps visual features to semantic embeddings and then reconstructs back the same visual feature. Recently, [28] described a transductive unbiased embedding to improve generalized ZSL performance. All past works in the transductive literature deal with only '*object recognition*', which is an fundamental but easier problem. In this paper, we study transductive setting for the challenging '*zero-shot detection*' problem.

**Pseudo-annotation for ZSL:** In the literature, pseudo-annotation has been used for ZSL in two different scenarios. *Firstly*, given the unseen class names available during training, these approaches try to learn the cluster structure of the unseen world. Typically, this is achieved by building a classifier for unseen classes by selecting pseudo-samples from seen images [6] or by generating pseudo-instances [31, 19, 20]. The main goal is to convert ZSL to a traditional supervised learning problem. *Secondly*, the pseudo-labels are assigned to unlabeled target data during transductive settings of ZSL/GZSL [7, 29, 34]. The goal is to convert ZSL to a domain adaptation problem. These approaches try to match the distribution of training and test data. Since we consider a transductive setting, current work follows the second scenario but in the context of the ZSD task. Different from previous works, we adopt a hybrid pseudo-labeling approach that combines fixed and dynamic updates to obtain more accurate detections in a transductive setting.

**Zero-shot object detection (ZSD):** The traditional object detection task has been well-explored, e.g., two-stage detectors like FasterRCNN [26], RFCN [10] and single-stage detectors like SSD [18], YOLO [25] and RetinaNet [15]) have been proposed. In comparison, ZSD has emerged as a relatively new research area [1, 3, 24, 38, 22]. Among them, [3, 38] build their architecture on YOLO, [24] on FasterRCNN and [22] on RetinaNet. [1] proposed a background-aware approach for ZSD based on EdgeBox-style object proposals without relying on any end-to-end framework. However, none of the ZSD methods ever considered a transductive setting of this problem. In this paper, we attempt to address this problem in a fully trainable pipeline. We built on top of the RetinaNet architecture for ZSD proposed in [22] as it reports the best performance in this area.

## 3. Transductive Zero-Shot Detection

Given a limited amount of seen data, ZSL aims to generalize to a highly diverse set of unseen objects. In reality, the data distribution of unseen (target) is significantly different to that of seen (source). This problem is called the '*domain-shift problem*' and poses a significant challenge to generalization of a ZSL approach [5]. To address this, we adopt the transductive setting for ZSL, i.e., using unlabeled test data during training.

During the training stage, a ZSL model observes only seen instances that makes the trained model biased towards only seen classes. In generalized zero-shot learning (GZSL), where both seen and unseen examples appear during the inference stage, this behavior causes serious problems [28]. In most of the cases, a biased trained

model predicts only seen categories irrespective of the input. To address this issue, we propose a pseudo-labeling scheme that not only maximizes the prediction score of the pseudo-ground truth class but also maximizes unseen scores in transductive settings. Remarkably, while previous works on ZSL only addressed zero shot recognition (ZSR) tasks to solve the problems above, we focus on a more challenging zero-shot detection (ZSD) task. Next, we elaborate the differences between our and previously considered settings and highlight the challenges it presents.

**Transductive ZSR vs. ZSD:** These two tasks are fundamentally different. *Firstly*, during training with unlabeled data in transductive ZSR, as only one object is present per image, a model knows which images are coming from seen and which from unseen data. This provides an important supervision signal during training. However, in ZSD, one image can contain multiple seen or unseen objects. For example, MSCOCO [16] contains 7.7 object instances per image. Therefore in a transductive ZSD, we know a test image may contain one or more unseen objects, but during training, both seen and unseen object annotations (label and bounding box) for test data are not present. *Secondly*, training of transductive ZSR often follows iterative joint learning by considering seen and unseen data separately [7, 11, 36, 32]. Such approaches generally work on top of fixed deep features and are not end-to-end trainable. In contrast, for transductive ZSD, we believe an end-to-end model can improve the performance due to the complexity of joint classification and localization.

**Transductive GZSR vs GZSD:** Using unseen data as unlabeled during training creates a problem for Generalized ZSR (GZSR) in transductive settings. This is because a GZSR method has a high-level supervision signal showing which objects are unseen (a single category is present in each image). Therefore a GZSR approach precisely knows which ones are seen objects and which examples belong to the unseen distribution. Due to this reason, existing transductive ZSR methods are not extendable to the GZSD setting. Song *et al*. [28] identify this problem and address it by dividing the unlabeled data in two halves to use one half in training and the other for testing. In this manner, although seen/unseen level supervision is available for the first half, the model does not exactly know the seen/unseen label for the test set.

In this paper, we deal with transductive GZSD in a way that no seen/unseen level supervision is available during training. Furthermore, a key challenge for ZSD is how to differentiate between background bounding boxes and unseen ones during training. As we explain next, our approach uses a hybrid pseudo-labeling strategy to approach this problem.

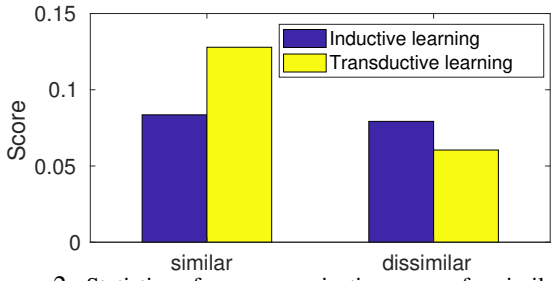

Figure 2: Statistics of average projection scores for similar and dissimilar unseen classes. Transductive learning provides higher and lower projection scores for similar and dissimilar classes respectively than inductive learning. Moreover, the gap between similar and dissimilar projection scores is increased from inductive to transductive learning.

### 3.1. Our Approach

**Problem Formulation:** Suppose, we have S seen and U unseen classes with a total of C = S+U classes. For each class, we have an associated $d$-dimensional semantic vector acquired in a supervised (manual attributes) or unsupervised manner (e.g., word2vec, GloVe). We represent the set of all semantic vectors using $\mathbf{W} = [\mathbf{W}_{\mathrm{S}}, \mathbf{W}_{\mathrm{U}}] \in \mathbb{R}^{d \times \mathrm{C}}$, where $\mathbf{W}_{\mathrm{S}} \in \mathbb{R}^{d \times \mathrm{S}}$ and $\mathbf{W}_{\mathrm{U}} \in \mathbb{R}^{d \times \mathrm{U}}$ are the collection of seen and unseen semantic vectors, respectively. We have a set $\mathcal{X}_{tr}$ consisting of $N_{tr}$ training images where each image contains one or more seen objects. Each seen object has a ground-truth label $y_{tr}$ and bounding box coordinates $\mathbf{b}_{tr}$. Similarly, we have $N_{ts}$ images in the test set $\mathcal{X}_{ts}$, where each image can have one or more objects from both seen and unseen categories. For each object in the test set, we denote the ground-truth label as $y_{ts}$ and the true bounding box as $\mathbf{b}_{ts}$.

Given the semantics $\mathbf{W}$, the set $\mathcal{X}_{tr}$ along with ground-truth labels $\mathcal{Y}_{tr}$ and the test image set $\mathcal{X}_{ts}$, we address the following two problems: (a) **Transductive ZSD:** Predict category labels $y_{ts}$ and object locations $\mathbf{b}_{ts}$ for only '*unseen*' classes present in the set $\mathcal{X}_{ts}$. (b) **Transductive GZSD:** Predict category labels $y_{ts}$ and object locations $\mathbf{b}_{ts}$ for both '*seen and unseen*' classes present in the set $\mathcal{X}_{ts}$.

Below, we first outline the inductive ZSD setting (Sec. 3.1.1) that acts as a precursor to our transductive ZSD approach (Sec. 3.1.2).

#### 3.1.1 Inductive ZSD

Given an input image $I$, an object detector model generates $K$ anchor boxes $\{b_i\}_{i=1}^{K}$. We represent $D$-dimensional visual feature vectors for each box, $b$ as $\mathbf{f} \in \mathbb{R}^{D}$. The classification branch of the detector generates prediction scores, $\mathbf{p}$ as follows:

$$\mathbf{p} = \sigma(\mathbf{f}^{T}\mathbf{U}\mathbf{W}) \tag{1}$$

where, $\mathbf{U} \in \mathbb{R}^{D \times d}$ are learnable parameters and $\sigma$ represents a sigmoid/softmax activation. The above relation incorporates semantic information (word vectors) within deep networks that is necessary to perform zero-shot learning. The learned projection $\mathbf{U}$ helps in aligning the feature vector $\mathbf{f}$ with the word vector of its corresponding seen class, $\mathbf{w}_y \in \mathbf{W}_\text{S}$. Another advantage of such prediction scoring is that it treats visual to semantic ($\mathbf{f}^T\mathbf{U}$ onto $\mathbf{W}$) and semantic to visual ($\mathbf{UW}$ onto $\mathbf{f}$) domain projection in an identical manner. We visualize the scores in Fig. 2. One can use these scores while calculating standard focal loss to train the detector in an end-to-end manner [15]:

$$\text{FL}(p, y) = -\alpha_t(1-p_t)^\gamma \log p_t,\ p_t = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise.} \end{cases}$$

where, $p \in \mathbf{p}$ represents an individual score, $\alpha$ and $\gamma$ are focal loss hyper-parameters. Depending on the considered setting, unseen word vectors may or may not be present during training. Therefore, for clarity, we present seen and unseen prediction scores as $\mathbf{s} = \sigma(\mathbf{f}^T\mathbf{U}\mathbf{W}_\text{S})$ and $\mathbf{u} = \sigma(\mathbf{f}^T\mathbf{U}\mathbf{W}_\text{U})$ respectively. For later discussion, $s$ and $u$ represent individual scores in $\mathbf{s}$ and $\mathbf{u}$, respectively.

#### 3.1.2 Transductive ZSD

The above scheme deals with conventional zero-shot detection. In transductive learning, data for unseen classes is available without any corresponding annotations. Therefore, after a detector is trained on all available seen data, we propose an intelligent pseudo-labeling scheme for these extra unlabeled samples that can provide a valuable supervisory signal for appropriate model training.

Our proposed approach has two complimentary components, namely **fixed** and **dynamic** pseudo-labeling. The first component aims to retain the previously acquired knowledge on seen classes and use it to disentangle 'seen' objects from the 'unseen' in the given unlabeled set. To this end, it only assigns seen class pseudo-labels. The second component aims to dynamically update the features and classifier based on the unlabeled dataset. In this pursuit, it assigns both 'seen' and 'unseen' object labels that continue updating as the learning progresses. In this manner, the model starts with the easily classified samples to update its knowledge about the unseen and gradually builds on initial concepts to improve its performance.

Our unlabeled set can contain examples of both seen and unseen classes, which makes ours a more challenging setting since we do not explicitly know which samples are unseen. To address this challenging problem, we propose fixed and dynamic pseudo-labeling techniques given the training is already done on only seen data. Next, we explain both pseudo-labeling approaches in detail.

**(a) Fixed pseudo-labeling:** Based on inductive learning described in Sec. 3.1.1, we apply the trained detector on unlabeled test data to detect seen objects. We use the detected seen labels and bounding boxes as pseudo-labels and keep this labeling as fixed throughout the transductive training. Since our proposed transductive ZSD setting does not consider labeled seen data (in the unlabeled test set), such pseudo-labeling can serve as a ground-truth label during the transductive training. One may argue that a fixed pseudo-labeling will hinder the training process, and an optimal pseudo-labeling should be adaptive during training i.e., it must continually update during the learning process. While a dynamic sub-component of our approach will be introduced in the next section, we note that it alone does not work and a fixed pseudo-labeling is a vital component of our transductive formulation. In fact, our fixed pseudo-labeling scheme helps us preserve the initial learning acquired by the model on seen examples where ground-truths were known. Therefore, this labeling scheme seeks to achieve learning without forgetting [13] and the fixed labels work as a distillation term [9].

As mentioned above, after completing the inductive phase of learning (Sec. 3.1.1), we perform fixed pseudo-labeling on unlabeled test data to improve our learned model. Then, we initialize our transductive model with the weights of the pre-trained inductive model. At each iteration, we calculate fixed pseudo-labeling loss. Suppose, $\hat{y}$ is the fixed pseudo-label of seen bounding boxes. During transductive training, we can calculate a fixed pseudo-labeling based focal loss as follows:

$$L_f = -\alpha_t(1-\hat{s}_t)^\gamma \log \hat{s}_t,\ \hat{s}_t = \begin{cases} s, & \text{if } \hat{y} = 1 \\ 1-s, & \text{otherwise.} \end{cases} \quad (2)$$

Fixed pseudo-labeling assigns only seen pseudo-labels to images as the inductive training did not observe any unseen information (both image and word vectors). Therefore, during the transductive training, we want to update the fixed seen pseudo labels as well as assign newly available unseen pseudo-labels in a dynamic way. In this pursuit, we propose dynamic pseudo-labeling which is introduced next.

**(b) Dynamic pseudo-labeling:** We propose a dynamic pseudo-labeling technique based on seen and unseen prediction scores, that keeps progressively updating in different iterations. It has three components respectively for seen prediction ($L_d(s)$), unseen prediction ($L_d(u)$) and unseen prediction maximization ($L'_d(u)$),

$$L_d = L_d(s) + L_d(u) + L'_d(u). \quad (3)$$

In each iteration, if a seen prediction $s$ gets a score higher than a pre-defined threshold ($t_h$), we assign a dynamic pseudo-label to the corresponding seen class. The loss as-

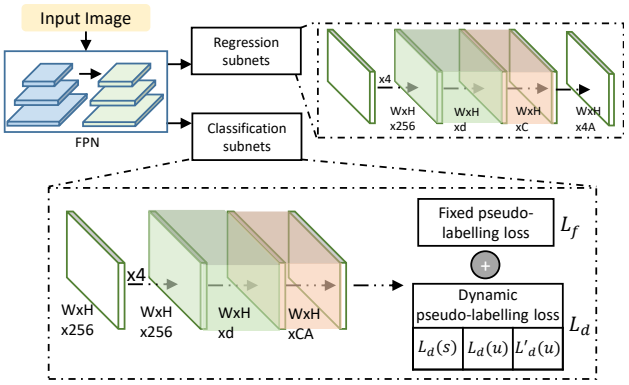


Figure 3: Network architecture. Green and red layers represent $\mathbf{U}$ and $\mathbf{W}$ of Eq. 1.

sociated with the seen pseudo-label is given by,

$$L_d(s) = -\alpha_t(1-s_t)^\gamma \log s_t,\ s_t = \begin{cases} s, & \text{if } s > t_h \\ 1-s, & \text{otherwise.} \end{cases}$$

Similarly, in the same iteration, if an unseen prediction $u$ gets a score higher than $t_h$, we assign a dynamic pseudo-label to the corresponding unseen class. The loss associated with the unseen pseudo-label is given by,

$$L_d(u) = -\beta_t(1-u_t)^\eta \log u_t,\ u_t = \begin{cases} u, & \text{if } u > t_h \\ 1-u, & \text{otherwise.} \end{cases}$$

The underlying intuition behind dynamic pseudo-labeling is to leverage the training so far to steadily improve the detection of unlabeled data. Note that our transductive training begins with a pre-trained model on seen data. Therefore, such pseudo-labeling is not random but important for further training. Additionally, as the pre-training is based on purely seen data, prediction scores become biased towards seen classes i.e. seen scores are relatively higher than unseen ones. To avoid such biased predictions, we propose a regularization term in the loss function that seeks to directly maximize the unseen predictions.

$$L'_d(u) = -\beta_t(1-u_t)^\eta \log u \tag{4}$$

We note that pushing unseen predictions towards higher values in fact avoids unseen classes being mapped to seen classes [28]. We add all these three parts together to calculate dynamic pseudo-label based loss. We merge $L_d(u)$ and $L'_d(u)$ together in Eq. 5 because both work on the same prediction score $u$.

$$L_d = -\alpha_t(1-s_t)^\gamma \log s_t - \beta_t(1-u_t)^\eta \log(uu_t). \tag{5}$$

**Overall transductive loss:** Our final loss for transductive training is a combination of both fixed $(L_f)$ and dynamic $(L_d)$ pseudo-labeling loss terms. A hyper-parameter $\lambda \in [0,1]$ controls the trade-off between both loss terms.

$$L = \lambda L_f + (1-\lambda)L_d. \tag{6}$$

The $L_d$ and $L_f$ in Eq. 6 are given by Eq. 5 and Eq. 2, respectively. Note that we use the same hyper-parameters $\alpha$ and $\gamma$ for focal loss calculation on the seen prediction score and $\beta$ and $\eta$ for the unseen scores. We illustrate the overall process in Algorithm 1.

**Algorithm 1:** Transductive zero-shot detection

**Input:** $N_{tr}$, $N_{ts}$, $\mathcal{X}_{tr}$, $y_{tr}$, $\mathbf{b}_{tr}$, $\mathcal{X}_{ts}$, $\mathbf{W}_\mathrm{S}$, $\mathbf{W}_\mathrm{U}$
**Output:** A trained model $\mathscr{M}_{tns}$ to find $y_{ts}$, $\mathbf{b}_{ts}$ for all $\mathcal{X}_{ts}$
**Inductive training phase**
$\mathscr{M}_{ind} \leftarrow$ Train an inductive model using only seen data: $N_{tr}$, $\mathcal{X}_{tr}$, $y_{tr}$, $\mathbf{b}_{tr}$, $\mathbf{W}_\mathrm{S}$
**Transductive training phase**
Initialize inductive model, $\mathscr{M}_{tns} \leftarrow \mathscr{M}_{ind}$
$\hat{y}_{ts} \leftarrow$ Use $\mathscr{M}_{ind}$ assign fixed pseudo-labels to unseen test images, $\mathcal{X}_{ts}$
**repeat**
  **for** $\forall I \in \mathcal{X}_{ts}$ **do**
    Calculate fixed pseudo-labeling loss $L_f$
    Calculate dynamic pseudo-labeling loss $L_d$
    Calculate overall transductive loss using 6
    Back-propagate and update $\mathscr{M}_{tns}$
**until** convergence;
**Return:** Using $\mathscr{M}_{tns}$ find $y_{ts}$, $\mathbf{b}_{ts}$ for all $\mathcal{X}_{ts}$

### 3.2. Training and Inference

**Network Architecture:** We choose a variant of the popular RetinaNet architecture [15, 22] with Feature Pyramid Network (FPN) [14] as the backbone, keeping ResNet50 [8] as a feature generator, to perform our transductive training. The overall architecture is shown in Fig. 3. An input image is passed through a ResNet50 [8] to generate a convolutional feature pyramid. Then FPN performs bottom-up and top-down processing to construct a rich and multi-scale discriminative feature space. Each pyramid level is then connected to two branches: classification and box regression subnets. Similar to the original recommendation, our anchors are at {1:2, 1:1, 2:1} aspect ratios with sizes $\{2^0, 2^{1/3}, 2^{2/3}\}$, totaling to A=9 anchors per level. If an anchor box gets an overlap $> 0.5$ in terms of intersection-over-union (IoU) with the ground-truth bonding box, we consider it as a valid object box prediction.

**Plugging semantics into RetinaNet:** We modify the penultimate layer of all branches to incorporate word vectors as mentioned in Eq. 1. In the **classification subnet**, initially four $3 \times 3$ convolution layers with ReLU are applied.

The output after this operation is a set of image features $\{\mathbf{f}_i \in \mathbb{R}^d\}$ for all $W \times H$ locations in an image, where $W$ and $H$ represent the height and width of the convolutional feature map. Then, we add another $3 \times 3$ convolution layer with $d \times A$ filters. The trainable weights of this layer implement $\mathbf{U}$ of Eq. 1. After that, we place a non-trainable custom layer having word vectors as fixed weights followed by a sigmoid activation to produce prediction scores. The last two layers can be summarized as Eq. 1. In the **regression subnet**, we again apply a similar strategy to plug semantics in to the pipeline. After producing the convolutional features map, we add a $3 \times 3$ convolution layer with $d$ filters. Then, a custom layer with non-trainable word vectors as weights are used to produce $\mathrm{S} + \mathrm{U}$ dimensional outputs. Finally, another convolution layer with $4A$ filters is used to generate bounding box parameters for each anchor at each spatial location. As suggested in [15], the classification and regression subnets do not share any parameters. During inductive training, we learn the network using the sum of the losses from the classification and regression subnets. The regression subnet branch is trained with the standard $L_1$ smooth loss. During transductive training, we calculate the loss from the classification subnet only because we assign pseudo-labels to the predictions of anchor boxes scores. We normalize each part of the loss by the total number of positive boxes during fixed and dynamic pseudo-labeling.

**Inference:** After a forward pass with a test image $I_u$, the classification and regression subnets produce class labels and bounding boxes, respectively, for all anchor boxes. Suppose, $\mathbf{f}_u$ is the image feature for an anchor box. We calculate seen scores using the following equation,

$$\mathbf{s} = \sigma(\mathbf{f}_u^T \mathbf{U} \mathbf{W}_{\mathrm{S}}). \tag{7}$$

For unseen scores, we apply the following equation:

$$\mathbf{p} = \sigma(\mathbf{f}_u^T \mathbf{U} \mathbf{W}), \qquad \mathbf{u} = \mathbf{p}' \mathbf{W}'^T \mathbf{W}_{\mathrm{U}}{}^T \tag{8}$$

Where, $\mathbf{p}'$ denotes top-T (e.g. T=5) predicted scores in $\mathbf{p}$ and $\mathbf{W}'$ is the corresponding word vectors of top predictions. We select the 100 top scoring bounding boxes and apply a Non-Maximal Suppression (NMS) with IoU=0.5 on the selected boxes. Finally, boxes that score higher than a specified threshold are chosen as the final detection.

Our proposed transductive solution has no additional parameters to train in comparison to inductive solution. After finishing the inductive training, we perform a few more epochs of training with unlabeled test data. Therefore the overall training time is relatively higher, but the inference time performance remains the same as the inductive case.

# 4. Experiments

## 4.1. Setup

**Dataset:** We have used the challenging MSCOCO-2014 dataset to test our approach. In the ZSD literature, two different types of seen/unseen split settings are available: the 48/17 and the 65/15 seen/unseen split by Bansal *et al.* [1] and Rahman *et al.* [22], respectively. In this paper, we choose the Rahman *et al.* [22] setting over [1] because it considers all 80 object classes of MSCOCO. The training set includes 62,300 images containing 51,782 bounding boxes from 65 seen classes. The test set for ZSD and GZSD includes 10,098 images having 16,388 bounding boxes. Besides, to test traditional detection task on seen classes, it provides a list of 38,096 images. To relate seen and unseen classes, we use 300-dimensional word2vec vectors [21].

**Evaluation:** To evaluate ZSD, [1] and [22] proposed to use recall@100 and mean average precision (mAP) with IoU=$0.5$ respectively. We report overall results on both evaluation metrics. However, for validation and ablation studies, we use mAP only because recall does not penalize wrong bounding box prediction. For GZSD, we report the harmonic-mean (HM) of seen and unseen performance.

**Implementation details:** We re-scale each image to make its smallest side 800px. During training, we ignore bounding boxes with IOU within $[.4, .5)$ and we consider those boxes with IOU within $[0, 0.4)$ as background. We first train a traditional RetinaNet architecture for 50 epochs (10K iterations/epoch) with only 65 seen classes and corresponding annotations. Using this pre-trained model, we perform an inductive training on the same data for 50 epochs (10K iterations/epoch). Finally, we conduct our proposed transductive learning for three epochs (30K iterations). In each iteration, we process only one image at a time. As we use 10,098 unlabeled images, the transductive learning observes each unlabeled image three times. During transductive training, we only train the classification branch by freezing the rest of the network. We also report comparison when the rest of the network is also tuned, that results in lower performance. We use Adam optimizer with a learning rate $10^{-5}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We implement our method in the *Keras* library.

**Validation experiments:** The hyper-parameters of our methods are $\alpha, \gamma, \beta, \eta, \lambda$ and $t_h$. Among them, $\alpha$ and $\gamma$ are focal loss [15] hyper-parameters. We use $\alpha = 0.25$ and $\gamma = 2$ as suggested in [15] for all of our experiments. For tuning the rest of hyper-parameters, we used the validation set comprising of images with seen objects for the traditional detection task. We report the validation performance in the supplementary material.

## 4.2. Main Results

**Compared methods:** We compare our results with in-

| Metric | Method | Seen/ Unseen | ZSD | GZSD | | |
|---|---|---|---|---|---|---|
| | | | | seen | unseen | HM |
| mAP | SB [1] | 48/17 | 0.70 | - | - | - |
| | DSES [1] | 48/17 | 0.54 | - | - | - |
| | FL-48 [22] | 48/17 | 5.91 | 36.57 | 2.64 | 4.93 |
| | FL-65 [22] | 65/15 | 10.80 | 37.56 | 10.80 | 16.77 |
| | FL-80 [22] | 65/15 | 10.73 | **40.60** | 10.28 | 16.40 |
| | Baseline | 65/15 | 12.40 | 29.52 | 11.91 | 16.97 |
| | Ours | 65/15 | **14.57** | 28.79 | **14.05** | **18.89** |
| RE | SB [1] | 48/17 | 24.39 | - | - | - |
| | DSES [1] | 48/17 | 27.19 | 15.02 | 15.32 | 15.17 |
| | FL-48 [22] | 48/17 | 18.67 | 42.21 | 17.60 | 24.84 |
| | FL-65 [22] | 65/15 | 22.18 | 40.29 | 22.14 | 28.57 |
| | FL-80 [22] | 65/15 | 22.25 | **59.19** | 19.43 | 29.25 |
| | Baseline | 65/15 | 48.06 | 54.89 | 33.38 | 41.52 |
| | Ours | 65/15 | **48.15** | 54.14 | **37.16** | **44.07** |

Table 1: Overall performance in mAP and recall (RE). For fair comparison, we compare our method with focal loss and no external information is used in the case of [22]. We get 'relative' improvements of 34.9% and 77.1% in terms of mAP and RE over the best inductive model.

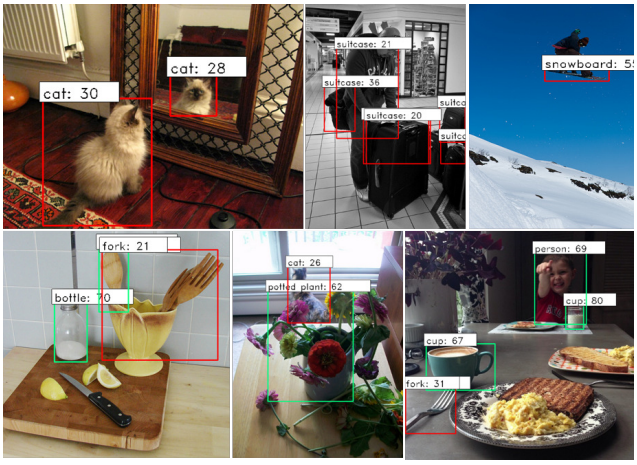


Figure 4: Qualitative results of ZSD (*top row*) and GZSD (*bottom row*). Red and green bounding boxes represent unseen and seen classes respectively.

ductive methods (SB, DSES, FL-48, FL-65 and FL-80) and a transductive baseline. SB and DSES are not end-to-end trainable since they use proposals drawn from EdgeBox [39] for ZSD. FL-48, FL-65, and FL-80 are the inductive approaches. FL-80 observes unseen word vectors (in addition to seen), but FL-48/FL-65 only observes 48/65 seen vectors based on the split settings. The transductive baseline method uses FL-65 as a pre-trained model and continues transductive learning without considering unseen word vectors i.e. $L_d(u) = 0$ and $L'_d(u) = 0$.

**Analysis:** We present the overall results in Table 1. In 48/17 split settings, FL-48 [22] outperforms SB/DSES [1] with a large margin in mAP. Being dependent on external proposals, SB/DSES suffers significantly. However, SB/DSES achieves a high recall because the recall metric

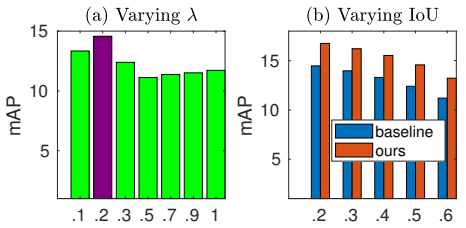


Figure 5: (a) Effect of varying $\lambda$ with our method, (b) mAP scores with different IoU.

does not penalize for wrong bounding box predictions. It shows that the end-to-end models in [22] are better than feature based models in [1]. In 65/15 split settings, FL-65 performs marginally better in mAP than FL-80 in both ZSD and GZSD tasks. The reason is FL-80 considers unseen vectors to calculate predictions scores in the last layer, but it does not perform any processing on unseen scores. It makes the model biased towards seen classes which results in a decrease in mAP. However, with the recall based metric, we notice an opposite scenario because it ignores the effect of false positives. Our transductive baseline beats all inductive methods as it uses unlabeled data by considering fixed and dynamic pseudo-labeling on only seen classes. Finally, our proposed model outperforms the transductive baseline in both ZSD and GZSD tasks because it uses both seen and unseen pseudo-labeling in the loss function. As recall is a less comprehensive measure than mAP, the improvement on recall is higher than mAP based evaluation. However, one can notice transductive methods lose some performance in GZSD-seen to achieve a balance between seen and unseen scores. In Fig. 5(a), we vary $\lambda$ to see the impact of fixed and dynamic pseudo-labeling. We notice that our experimentally validated $\lambda = .2$ brings an ideal balance between both both pseudo-labeling methods. In Fig. 5(b), we illustrate the comparison of the baseline and our method with different IoU settings. For more strict IOU thresholds, the performance of both approaches gradually decreases. In Table 2, we compare per-class AP of unseen classes between the inductive and our proposed transductive approach. Here, we notice our proposed method achieves higher mAP in most of the unseen classes than the inductive method. We have also shown some qualitative results in Fig. 4.

### 4.3. Ablation Studies

**Dynamic pseudo-labeling:** Our proposed transductive ZSD method works with fixed and dynamic pseudo-labeling techniques. We argue that the fixed part is the most important in this approach because it tries to retain the knowledge obtained from inductive training. The addition of dynamic pseudo-labeling tries to improve the inductive performance and reduce domain-shift leveraging the unlabeled data. It has three components: $L_d(s)$, $L_d(u)$ and $L'_d(u)$. For the ablation study in Table 3, we explore different combinations of

| **Method** | **Overall** | airplane | train | parking meter | cat | bear | suitcase | frisbee | snow-board | fork | sand-wich | hot dog | toilet | mouse | toaster | hair drier |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inductive | 10.80 | 6.23 | 50.01 | 2.61 | 34.58 | 0.0 | 10.93 | **13.62** | 20.91 | 10.96 | 9.57 | 0.77 | 0.64 | 0.14 | 1.04 | 0.0 |
| Ours | **14.57** | **19.75** | **63.40** | **3.65** | **43.18** | **3.68** | **13.78** | 12.81 | **24.24** | **12.61** | **9.65** | **5.99** | **1.54** | **2.26** | **2.03** | 0.0 |

Table 2: Per-class AP of unseen classes in MSCOCO dataset.

| $L_d$ | ZSD | GZSD | | |
|---|---|---|---|---|
| | | seen | unseen | HM |
| Baseline, $L_d(s)$ | 12.40 | 29.52 | 11.91 | 16.97 |
| $L_d(u)$ | 10.97 | 29.69 | 9.77 | 14.70 |
| $L'_d(u)$ | 9.47 | **30.65** | 7.98 | 12.67 |
| $L_d(u) + L'_d(u)$ | 12.41 | 29.32 | 10.97 | 15.96 |
| $L_d(s) + L'_d(u)$ | 12.60 | 28.65 | 12.05 | 16.96 |
| $L_d(s) + L_d(u)$ | 13.22 | 28.94 | 12.78 | 17.73 |
| $L_d(s) + L_d(u) + L'_d(u)$ | **14.57** | 28.79 | **14.05** | **18.89** |

Table 3: mAP for using different dynamic pseudo-labelling.

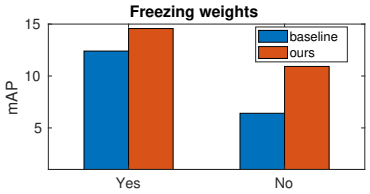


Figure 6: *Freezing effect:* When the classification subnet is trained keeping the rest of the network fixed, our approach performs better.

these components keeping the same fixed labeling. Among these components, $L_d(s)$ works on seen, while $L_d(u)$ and $L'_d(u)$ on unseen prediction scores. As an individual component, $L_d(s)$ does not use any unseen predictions, thus it achieved an improvement over inductive learning (10.8 to 12.4 for ZSD). However, $L_d(u)$ or $L'_d(u)$ could not work well alone as a dynamic component because $L_d(u)$ still suffers from the model-bias problem of inductive learning and $L'_d(u)$ tries to solve the bias but cannot pseudo-label the anchors. Therefore, the combination $L_d(s) + L_d(u)$ jointly improves the performance to the level of $L_d(s)$. In general, we notice that our transductive learning achieves relatively less mAP when dynamic labeling is based on only seen ($L_d(s)$) or unseen ($L_d(u)/L'_d(u)$) predictions. In contrast, when both seen and unseen predictions are used, we notice a clear improvement in performance. For example, $L_d(s) + L'_d(u)$ and $L_d(s) + L_d(u)$ got 12.60 and 13.22 on ZSD, and 16.96 and 17.73 on GZSD tasks. Our final model outperforms all others because as it takes advantage of all three proposed components $L_d(s), L_d(u)$ and $L'_d(u)$.

**Freezing effect:** As mentioned earlier in Sec. 3.2, during our transductive training, we only fine-tune the classification subnet because our pseudo-labeling process only assigns class labels. In Fig. 6, we report the effect of freezing the rest of the network (i.e., other than the classification branch). We notice that in both baseline and our method's case, this idea helps to improve the performance significantly. Unlike many traditional transductive approaches, our entire learning is based on pseudo-labels. Therefore, allowing the whole network to update its weights can mislead the learning process (as pseudo-labels can be noisy) and therefore result in lower performance.

| Method | ZSD | GZSD | | |
|---|---|---|---|---|
| | (mAP/RE) | seen (mAP/RE) | unseen (mAP/RE) | HM (mAP/RE) |
| FL-80 | 10.36/34.29 | 36.69/39.53 | 10.33/36.62 | 16.12/36.34 |
| Baseline | 11.05/43.20 | 29.82/55.05 | 11.09/30.31 | 16.17/39.09 |
| Ours | **12.87/47.46** | **29.93/55.98** | **12.19/31.22** | **17.32/40.09** |

Table 4: Results with GloVe vectors.

| Method | Avg. | car | dog | sofa | train |
|---|---|---|---|---|---|
| [3] | 54.5 | 55.0 | 82.0 | 55.0 | 26.0 |
| [22] | 62.1 | 63.7 | **87.2** | 53.2 | 44.1 |
| Ours | **66.6** | **64.4** | 77.9 | **70.5** | **53.6** |

Table 5: PASCAL VOC experiment using the split in [3].

**GloVe embedding:** Our method can work equally with other semantics apart from word2vec. In Table 4, we experiment with GloVe as semantic embedding. Our method successfully outperforms the inductive version (FL-80) and the transductive baseline in both ZSD and GZSD tasks with mAP and Recall (RE) based evaluation metrics.

**Beyond MSCOCO:** Using the setup in [3], we perform additional experiments with the Pascal VOC 2007/2012 dataset. In Table 5, we report ZSD mAP of unseen classes with the standard 16/4 split. Our transductive solution successfully outperforms the recent methods of [3] and [22].

## 5. Conclusion

Recently, zero-shot detection has received considerable attention from the research community. To address the domain shift and bias problem of inductive learning models, in this paper, we propose a transductive solution for ZSD. We leverage unlabeled testing data during transductive learning by employing fixed and dynamic pseudo-labeling based loss functions. Unlike the traditional transductive method, we do not use seen/unseen label supervision for unlabeled data. Moreover, most transductive learning-based recognition methods lack the end-to-end trainable solutions. However, our approach is end-to-end trainable with the proposed loss functions. In our experiments on the challenging MSCOCO dataset, we show that our method provides performance gains for both ZSD and GZSD problems.

**Acknowledgment.** This work was supported in part by NH&MRC Project grant #1082358.

## References


[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 52–68, Cham, 2016. Springer International Publishing.

[3] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. In *British Machine Vision Conference (BMVC)*, Sep. 2018.

[4] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, pages 584–599. Springer, 2014.

[5] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2332–2345, Nov. 2015.

[6] Yuchen Guo, Guiguang Ding, Jungong Han, and Yue Gao. Zero-shot learning with transferred samples. *IEEE Transactions on Image Processing*, 26(7):3277–3290, July 2017.

[7] Yuchen Guo, Guiguang Ding, Xiaoming Jin, and Jianmin Wang. Transductive zero-shot recognition via shared model space learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3494–3500. AAAI Press, 2016.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[10] Kaiming He Jian Sun Jifeng Dai, Yi Li. R-FCN: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.

[11] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] Yanan Li, Donghui Wang, Huanhang Hu, Yuetan Lin, and Yueting Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[14] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[16] 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 *ECCV*, pages 740–755. Springer, 2014.

[17] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[19] Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] Jiang Lu, Jin Li, Ziang Yan, and Changshui Zhang. Zero-shot learning by generating pseudo feature representations. *arXiv preprint arXiv:1703.06389*, 2017.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[22] Shafin Rahman, Salman Khan, and Nick Barnes. Polarity loss for zero-shot object detection. *arXiv preprint arXiv:1811.08982*, 2018.

[23] Shafin Rahman, Salman Khan, and Nick Barnes. Deep0tag: Deep multiple instance learning for zero-shot image tagging. *IEEE Transactions on Multimedia*, 2019.

[24] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *Computer Vision – ACCV 2018*, pages 547–563, Cham, 2019. Springer International Publishing.

[25] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[27] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 46–54. Curran Associates, Inc., 2013.

[28] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] Yao-Hung Hubert Tsai Yi-Ren Yeh Tao, Shih-Yen and Yu-Chiang Frank Wang. Semantics-preserving locality embedding for zero-shot learning. In *British Machine Vision Conference (BMVC)*, 2017.

[30] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learninga comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, Sep. 2019.

[31] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] Xing Xu, Fumin Shen, Yang Yang, Jie Shao, and Zi Huang. Transductive visual-semantic embedding for zero-shot learning. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 41–49, New York, NY, USA, 2017. ACM.

[33] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[34] Meng Ye and Yuhong Guo. Progressive ensemble networks for zero-shot recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Yunlong Yu, Zhong Ji, Jichang Guo, and Yanwei Pang. Transductive zero-shot learning with adaptive structural embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4116–4127, 2018.

[36] Yunlong Yu, Zhong Ji, Xi Li, Jichang Guo, Zhongfei Zhang, Haibin Ling, and Fei Wu. Transductive zero-shot learning with a self-training dictionary approach. *IEEE Transactions on Cybernetics*, 48(10):2908–2919, Oct 2018.

[37] An Zhao, Mingyu Ding, Jiechao Guan, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Domain-invariant projection learning for zero-shot recognition. In *Advances in Neural Information Processing Systems 31*, pages 1019–1030. Curran Associates, Inc., 2018.

[38] Pengkai Zhu, Hanxiao Wang, Tolga Bolukbasi, and Venkatesh Saligrama. Zero-shot detection. *arXiv preprint arXiv:1803.07113*, 2018.

[39] Charles Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, pages 391–405, Cham, 2014. Springer International Publishing.