Improved Visual-Semantic Alignment for Zero-Shot Object Detection

Shafin Rahman,[1,2] **Salman Khan,**[3,1] **Nick Barnes**[1,2]
[1]College of Engineering and Computer Science, Australian National University
[2]Data61, Commonwealth Scientific and Industrial Research Organisation
[3]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
firstname.lastname@anu.edu.au

Abstract

Zero-shot object detection is an emerging research topic that aims to recognize and localize previously 'unseen' objects. This setting gives rise to several unique challenges, e.g., highly imbalanced positive vs. negative instance ratio, proper alignment between visual and semantic concepts and the ambiguity between background and unseen classes. Here, we propose an end-to-end deep learning framework underpinned by a novel loss function that handles class-imbalance and seeks to properly align the visual and semantic cues for improved zero-shot learning. We call our objective the *'Polarity loss'* because it explicitly maximizes the gap between positive and negative predictions. Such a margin maximizing formulation is not only important for visual-semantic alignment but it also resolves the ambiguity between background and unseen objects. Further, the semantic representations of objects are noisy, thus complicating the alignment between visual and semantic domains. To this end, we perform metric learning using a *'Semantic vocabulary'* of related concepts that refines the noisy semantic embeddings and establishes a better synergy between visual and semantic domains. Our approach is inspired by the embodiment theories in cognitive science, that claim human semantic understanding to be grounded in past experiences (seen objects), related linguistic concepts (word vocabulary) and the visual perception (seen/unseen object images). Our extensive results on MS-COCO and Pascal VOC datasets show significant improvements over state of the art.[1]

1 Introduction

Zero shot learning (ZSL) is considered the 'holy-grail' among transfer learning problems. The goal is to reason about objects that have never been seen before. Traditional ZSL literature only focuses on '*recognizing*' unseen objects. Since real-world objects only appear as a part of a complete scene, the newly introduced zero shot object detection (ZSD) task (Rahman, Khan, and Porikli 2018b) considers a more practical setting where the goal is to simultaneously '*locate and recognize*' unseen objects. A successful ZSD system can help pave the way for lifelong learning machines that intelligently discover new objects and incrementally enhance their knowledge.

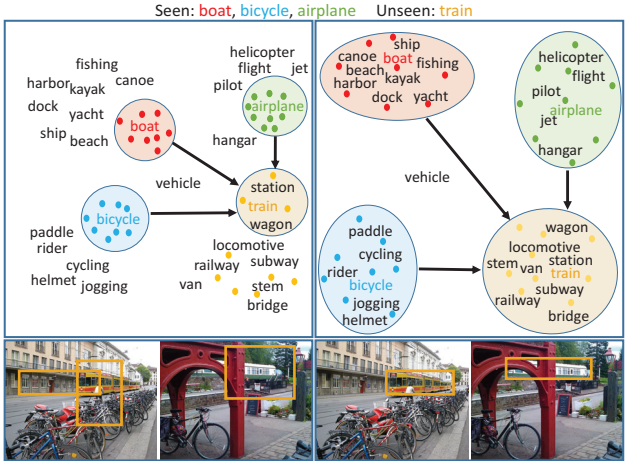

Figure 1: (*Top Left*) Traditional ZSD approaches align visual features (solid dots) to their corresponding semantics (e.g., boat, airplane) without considering the related semantic concepts (black text). It results in a fragile description of an unseen class (train) and causes confusion with background and seen classes (bottom left). (*Top Right*) Our approach automatically attends to related semantics from an external vocabulary and reshapes the semantic embedding so that visual features are well-aligned with related semantics. Moreover, it maximizes the inter-class separation that avoids confusion between unseen and background (bottom right).

Recently, the initial attempts on ZSD have been reported (Bansal et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018; Rahman, Khan, and Porikli 2018b; Zhu et al. 2018; Li et al. 2019). We note that these efforts suffer from limitations that lead to poor visual-semantic alignment, e.g, (*a*) lack of an end-to-end trainable pipeline (Bansal et al. 2018), (*b*) confusion between background and unseen classes (Bansal et al. 2018; Zhu et al. 2018), (*c*) inability to update semantic embeddings based on visual-semantic relationships (Bansal et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018), (*d*) object localization is not directly influenced by the seman-

[1]Code and evaluation protocols available at: https://github.com/salman-h-khan/PL-ZSD_Release

tic information (Bansal et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018; Rahman, Khan, and Porikli 2018b; Zhu et al. 2018), (*e*) inability to predict bounding boxes specific to unseen objects (Rahman, Khan, and Porikli 2018b; Bansal et al. 2018), (*f*) reliance on pre-trained weights tuned on datasets containing unseen objects (Bansal et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018).

In this work, we propose an integrated deep learning framework for ZSD that addresses the above-mentioned challenges. Our approach focuses on learning the complex interplay between visual and semantic domains such that the unseen objects (alongside the seen ones) can be accurately detected and localized. Towards this goal, we introduce a novel loss function called 'Polarity loss' that directly maximizes the margin between positive and negative detections. Our loss function helps distinguish between background and unseen classes, improves visual-semantic alignment (through maximally separating class predictions) while inherently tackling the class-imbalance problem. Despite the advantages of Polarity loss, the unsupervised semantic embeddings (e.g., word2vec) are noisy and complicate the visual-semantic alignment. To address this problem, we introduce a new vocabulary metric that learns to improve the semantic embeddings to better align them with the visual concepts. We integrate both novel components in a fully trainable single-shot object detector that offers high efficiency and superior performance.

Our main contributions are:

- An end-to-end single-shot ZSD framework based on a novel loss function called '*Polarity loss*' to address object-background imbalance and achieve maximal separation between positive and negative predictions.
- Using an external vocabulary of words, our approach learns to associate semantic concepts with both seen and unseen objects. This helps to resolve confusion between unseen classes and background, and to appropriately reshape the noisy word embeddings.
- A new seen-unseen split on the MS-COCO dataset that respects practical considerations such as diversity and rarity among unseen classes.
- Extensive experiments on the old and new splits for MS-COCO and Pascal VOC which give absolute gains of 9.3 and 7.6 in mAP over (Bansal et al. 2018) and (Demirel, Cinbis, and Ikizler-Cinbis 2018), respectively.

2 Related work

The ZSL literature is predominated by classification approaches that focus on single (Fu et al. 2018) or multi-label (Rahman and Khan 2018) recognition. The ZSD problem has recently been investigated by (Rahman, Khan, and Porikli 2018b; Zhu et al. 2018; Bansal et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018; Li et al. 2019). These methods can detect unseen objects using box annotations of seen objects. Among them, (Bansal et al. 2018) proposed a feature based approach where object proposals are generated by edge-box (Zitnick and Dollár 2014), and (Rahman, Khan, and Porikli 2018b; Zhu et al. 2018; Demirel, Cinbis, and Ikizler-Cinbis 2018) modified the object detection frameworks (Ren et al. 2017; Redmon and Farhadi 2017) to adapt ZSD settings. In another work, (Li et al. 2019) use textual description of seen/unseen classes to obtain semantic representations that are used for zero-shot object detection. The generalization of this problem is called generalized-ZSD (GZSD) which aims to detect both seen and unseen objects together (Bansal et al. 2018). A transductive approach to leverage unlabelled target-domain data is proposed in (Rahman, Khan, and Barnes 2019). Different from above works, we propose a new loss formulation that can greatly benefit single stage zero-shot object detectors.

3 Max-margin cross-entropy

We first introduce the proposed Polarity Loss that builds upon Focal loss (Lin et al. 2018) for generic object detection. Focal loss only promotes correct prediction, whereas a sound ZSL system should also learn to minimize projections on representative vectors for negative classes. Our proposed loss jointly maximizes projection on correct classes and minimizes the alignment with incorrect ones. This approach effectively allows distinction between background *vs.* unseen classes and promotes better alignment between visual and semantic concepts. Below, we provide a brief background and then propose the Polarity loss.

3.1 Balanced Cross-Entroy vs. Focal loss

Consider a binary classification task where $y \in \{0, 1\}$ denotes the ground-truth class and $p \in [0, 1]$ is the prediction probability for the positive class (i.e., $y = 1$). The standard binary cross-entropy (CE) formulation gives:

$$\text{CE}(p, y) = -\alpha_t \log p_t, \quad p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise.} \end{cases} \tag{1}$$

where, α is a loss hyper-parameter representing inverse class frequency and the definition of α_t is analogous to p_t. In the object detection case, the object vs. background ratio is significantly high (e.g., 10^{-3}). Using a weight factor α is a traditional way to address this strong imbalance. However, being independent of the model's prediction, this approach treats both well-classified (easy) and poorly-classified (hard) cases equally. It favors easily classified examples to dominate the gradient and fails to differentiate between easy and hard examples. To address this problem, Lin *et al.* (Lin et al. 2018) proposed 'Focal loss' (FL):

$$\text{FL}(p, y) = -\alpha_t (1 - p_t)^\gamma \log p_t \tag{2}$$

where, $\gamma \in [0, 5]$ is a loss hyper-parameter that dictates the slope of cross entropy loss (a large value denotes higher slope). The term $(1 - p_t)^\gamma$ enforces a high and low penalty for hard and easy examples respectively. In this way, FL simultaneously addresses object vs. background imbalance and easy vs. hard examples difference during training.

Shortcomings: In zero-shot learning, it is highly important to align visual features with semantic word vectors. This alignment requires the training procedure to (1) push visual features close to their ground-truth embedding vector and (2) push them away from all negative class vectors. FL can only perform (1) but cannot enforce (2) during the training of ZSD. Therefore, although FL is well-suited for traditional seen object detection, but not for the ZSD scenario.

3.2 Polarity Loss definition

To address the above-mentioned shortcomings, we propose a margin maximizing loss formulation that is particularly suitable for ZSD. This formulation is generalizable and can work with loss functions other than Eqs. 1 and 2. However, for the sake of comparison with the best model, we base our analysis on the state of the art FL.

Multi-class Loss: Consider that a given training set $\{\mathbf{x}, \mathbf{y}\}_i$ contains N examples belonging to C object classes plus an additional background class. For the multi-label prediction case, the problem is treated as a sum of individual binary cross-entropy losses where each output neuron decides whether a sample belongs to a particular object class or not. Assume, $\mathbf{y} = \{y^i \in \{0,1\}\} \in \mathbb{R}^C$ and $\mathbf{p} = \{p^i \in [0,1]\} \in \mathbb{R}^C$ denote the ground-truth label and prediction vectors respectively, and the background class is denoted by $\mathbf{y} = \mathbf{0} \in \mathbb{R}^C$. Then, the FL for a single box proposal is:

$$\mathscr{L} = \sum_i -\alpha_t^i (1-p_t^i)^\gamma \log p_t^i. \tag{3}$$

Polarity Loss: Suppose, for a given bounding box feature containing an ℓ^{th} object class, p^ℓ represents the prediction value for the ground-truth object class, *i.e.*, $y^\ell = 1$, see Table 1. Note that $p^\ell = 0$ for the background class (where $y^i = 0; \forall i$). Ideally, we would like to maximize the predictions for ground-truth classes and simultaneously minimize prediction scores for all other classes. We propose to explicitly maximize the margin between predictions for positive and negative classes to improve the visual-semantic alignment for ZSD (see Fig. 2). This leads to a new loss function that we term as 'Polarity Loss' (PL):

$$\mathscr{L}_{PL} = \sum_i f_p(p^i - p^\ell)\text{FL}(p^i, y^i), \tag{4}$$

where, f_p is a monotonic penalty function. For any prediction, p^i where $\ell \neq i$, the difference $p^i - p^\ell$ represents the disparity between the true class prediction and the prediction for the negative class. The loss function enforces a large negative margin to push predictions p^i and p^ℓ further apart. Thus, for an object anchor case, the above objective enforces $p^\ell > p^i$, while for background case $0 > p^i$ *i.e.*, all p^i's are pushed towards zero (since $p^\ell = 0$).

Our Penalty Function: f_p should necessarily be a 'monotonically increasing' function. It offers a small penalty if the gap $p^i - p^\ell$ is low and a large penalty if the gap is high. This constraint enforces that $p^i < p^\ell$. In this paper, we implement f_p with β parameterized sigmoid function:

$$f_p(p^i - p^\ell) = \frac{1}{1 + \exp(-\beta(p^i - p^\ell))} \tag{5}$$

For the case when $p^i = p^\ell$, the FL part guides the loss because f_p becomes a constant. We choose a sigmoid form for f_p because the difference $(p^i - p^\ell) \in [-1, 1]$ and f_p can be bounded by $[0, 1]$, similar to α_t or $(1 - p_t)$ factor of FL. Note that, it is not compulsory to stick with this particular choice of f_p.

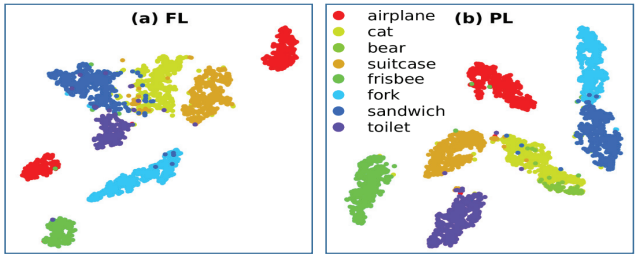

Figure 2: tSNE plot of visual features from 8 unseen classes projected onto semantic space using (a) FL & (b) PL. FL pushes visual features close to their ground-truth. Thus, intra-class distances are minimized but inter-class distances are not considered. This works well for seen class separation, but is not optimal for unseen classes because inter-class distances must be increased to ensure unseen class separability. Our PL ensures this requisite.

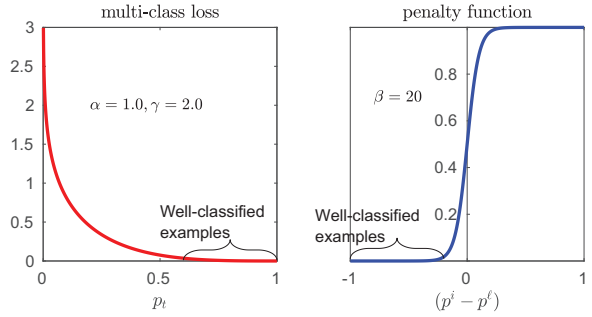

Figure 3: Plot of (left) multi-class loss (right) penalty function.

(a) Object case: $p^\ell = .8$

p^i	.1	.8	.9
y^i	0	1	0
p_t^i	.9	.8	.1
$p^i - p^\ell$	-.7	0	.1
loss	L	L	H

(b) Background case: $p^\ell = 0$

p^i	.1	.8	.9
y^i	0	0	0
p_t^i	.9	.2	.1
$p^i - p^\ell$	.1	.8	.9
loss	L	H	H

Table 1: A toy example. Intermediate computations for Polarity Loss are shown. Low (L) values are shown in green while High (H) values are shown in red. A mismatch between (p^i and y^i) + a close match between (y^i and y^ℓ) results in a high loss.

Final Objective: The final form of the loss is:

$$\mathscr{L}_{PL}(\mathbf{p}, \mathbf{y}) = \sum_i \frac{-\alpha_t^i (1-p_t^i)^\gamma \log p_t^i}{1 + \exp(-\beta(p^i - p^\ell))}, \text{ where,}$$

$$p_t^i = \begin{cases} p^i, & \text{if } y^i = 1 \\ 1 - p^i, & \text{otherwise} \end{cases} \qquad p^\ell = p^i [\![y^i = 1]\!], \tag{6}$$

where, $[\![\cdot]\!]$ denotes the Iverson bracket.

A Toy Example: We explain the proposed loss with a toy example in Table 1 and Fig. 3. When an anchor box belongs to an '*object*' and $p_t^i \geq .5$ (high) then $p^i - p^\ell \leq 0$ (low).

From Fig. 3, both a multi-class loss and the penalty function find low loss which eventually calculates a low loss. Similarly, when $p_t^i < .5$ (low), $p^i - p^\ell > 0$ (high), which evaluates to a high loss. When an anchor belongs to '*background*', $p^i - p^\ell \geq 0$ and a high p^i results in a high value for both multi-class loss and the penalty function and vice versa. In this way, the penalty function always supports multi-class loss based on the disparity between the current prediction and ground-truth class's prediction.

Polarity Loss Properties: The PL has two intriguing properties. *(a) Word-vectors alignment:* For ZSL, generally visual features are projected onto the semantic word vectors. A high projection score indicates proper alignment with a word-vector. The overall goal of training is to achieve good alignment between the visual feature and its corresponding word-vector and an inverse alignment with all other word-vectors. In our proposed loss, FL$(\cdot)$ and f_p perform the direct and inverse alignment respectively. Fig. 2 shows visual features before and after this alignment. *(b) Class imbalance:* The penalty function f_p follows a trend similar to α_t and $(1-p_t)^\gamma$. It means that f_p assigns a low penalty to well-classified/easy examples and a high penalty to poorly-performed/hard cases. It greatly helps in tackling class imbalance for single stage detectors where negative boxes heavily outnumber positive detections.

4 Vocabulary metric learning

Apart from proper visual-semantic alignment and class imbalance, a significant challenge for ZSD is the inherent noise in the semantic space. In this paper, we propose a new 'vocabulary metric learning' approach to improve the quality of word vectors for ZSL tasks. For brevity of expression, we restrict our discussion to the case of classification probability prediction or bounding box regression for a single anchor. Suppose, the visual feature of that anchor, $\mathbf{a}$ is $\phi(\mathbf{a}) = \mathbf{f}$, where ϕ represents the detector network. The total number of seen classes is S and a matrix $W_s \in \mathbb{R}^{\mathrm{S}\times d}$ denotes all the d-dimensional word vectors of S seen classes arranged row-wise. The detector network ϕ is augmented with FC layers towards the head to transform the visual feature $\mathbf{f}$ to have the same dimension as the of word vectors, i.e., $\mathbf{f} \in \mathbb{R}^d$. In Fig. 4, we describe several ways to learn the alignment function between visual features and semantic information. We elaborate these further below.

4.1 Learning with Word-vectors

For the traditional detection case, shown in Fig. 4(a), the visual features $\mathbf{f}$ are transformed with a learnable FC layer $W_d \in \mathbb{R}^{\mathrm{S}\times d}$, followed by a sigmoid/softmax activation (σ) to calculate S prediction probabilities, $\mathbf{p}_d = \sigma(W_d\mathbf{f})$. This approach works well for traditional object detection, but it is not suitable for the zero-shot setting as the transformation W_d cannot work with unseen object classes.

A simple extension of the traditional detection framework to the zero-shot setting is possible by replacing trainable weights of the FC layers, W_d, by the non-trainable seen word vectors W_s (Fig. 4(b)). Keeping this layer frozen, we allow projection of the visual feature $\mathbf{f}$ to the word embedding space to calculate prediction scores $\mathbf{p}_s$:

$$\mathbf{p}_s = \sigma(W_s\mathbf{f}) \tag{7}$$

This projection aligns visual features with the word vector of the corresponding true class. The intuition is that rather than directly learning a prediction score from visual features (in Fig 4(a)), it is better to learn a correspondence between the visual features with word vectors before the prediction.

Challenges with Basic Approach: Although the configuration described in Fig. 4(b) delivers a basic solution to zero-shot detection, it suffers from several limitations. (1) *Fixed Representations:* With a fixed embedding W_s, the network cannot update the semantic representations and has limited flexibility to properly align visual and semantic domains. (2) *Limited word embeddings:* The word embedding space is usually learned using billions of words from unannotated texts which results in noisy word embeddings. Understanding the semantic space with only S word vectors is therefore unstable and insufficient to model visual-semantic relationships. (3) *Unseen-background confusion:* In ZSD, one common problem is that the model confuses unseen objects with background since it has not seen any visual instances of unseen classes (Bansal et al. 2018).

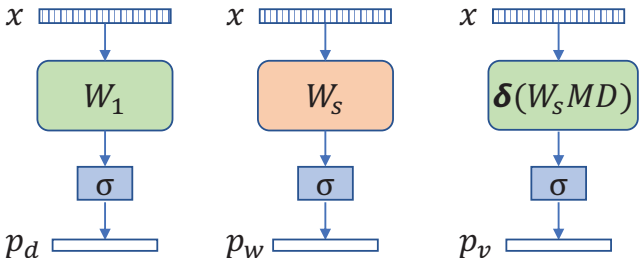

Figure 4: (a) Traditional basic approach with learnable W_d, (b) Inserting word vectors as a fixed embedding W_s, (c) learnable word vectors with vocabulary metric $\delta(W_sMD)$.

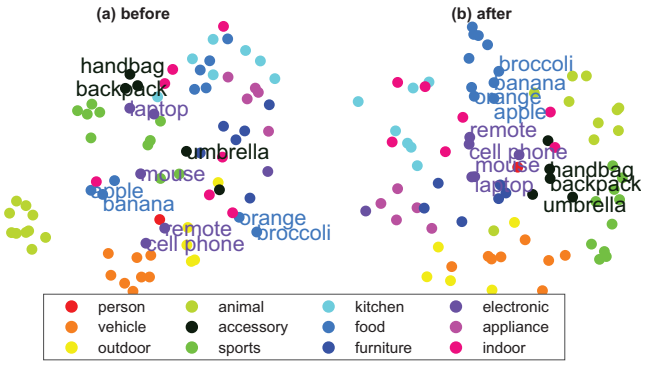

Figure 5: 2D tSNE (Van Der Maaten 2014) embedding of word2vec: (a) before (b) after modification based on vocabulary metric with our loss. Word-vectors are more evenly distributed in (b) than (a). Also, visually similar classes for example, apple/banana/orange/broccoli, cell phone/remote/laptop/mouse and handbag/backpack/umbrella are embedded more closely in (b) than (a). Super-category annotations are used for visualization only, not during our training.

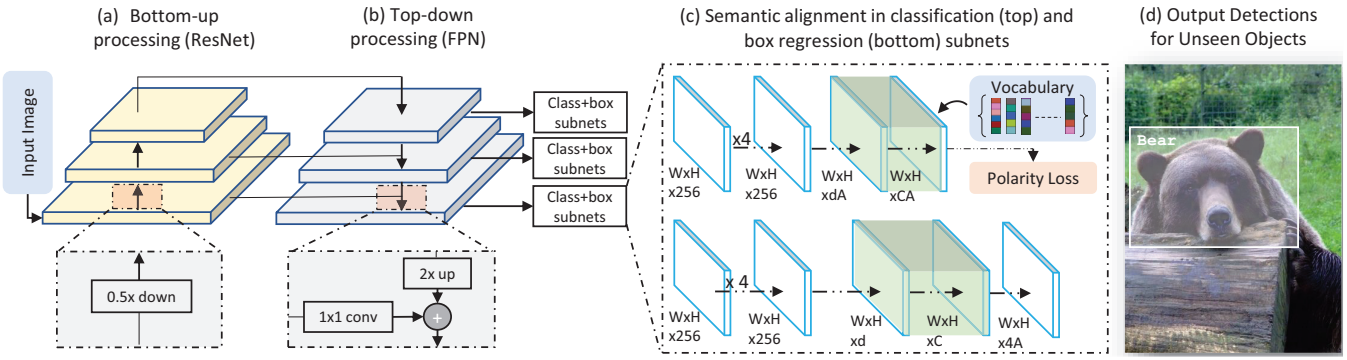

Figure 6: Network architecture for ZSD. The green colored layer implements Eq. 7 (Our-PL-word) or 8 (Our-PL-vocab).

4.2 Learning with vocabulary metric

To address the above gaps, we propose to learn a more expressive and flexible semantic domain representation. Such a representation can lead to a better alignment between visual features and word vectors. Precisely, we propose a vocabulary metric method summarized in Fig. 4(c) that takes advantage of the word vectors of a pre-defined vocabulary, $D \in \mathbb{R}^{v \times d}$ (where v is the number of words in the vocabulary). By relating the given class semantics with dictionary atoms, the proposed approach provides an inherent mechanism to update class-semantics optimally for ZSD. Now, we calculate the prediction score as follows:

$$\mathbf{p}_v = \sigma(\delta(W_s M D)\mathbf{f}). \tag{8}$$

Here, $M \in \mathbb{R}^{d \times v}$ represents the learnable parameters which connect seen word vectors with the vocabulary and $\delta(.)$ is a $\tanh$ activation function. M can be interpreted as learned attention over the dictionary. With such an attention, the network can understand the semantic space better and learn a rich representation because it considers more linguistic examples (vocabulary words) inside the semantic space. Simultaneously, it helps the network to update the word embedding space for better alignment with visual features. Further, it reduces unseen-background confusion since the network can relate visual features more accurately with a diverse set of linguistic concepts. We visualize word vectors before and after the update in Fig. 5 (a) and (b) respectively.

Here, we emphasize that the previous attempts to use such an external vocabulary have their respective limitations. For example, (Al-Halah, Tapaswi, and Stiefelhagen 2016) considered a limited set of attributes while (Al-Halah and Stiefelhagen 2017) used several disjoint training stages. These approaches are therefore not end-to-end trainable. Further, they only investigate the recognition problem.

Regression branch with semantic: Eq. 8 allows our network to predict seen class probabilities at the classification branch directly using semantic information from the vocabulary metric. Similarly, we also apply such semantics in the regression branch with some additional trainable FC layers. In our experiments, we show that adding semantics in this manner leads to further improvement in ZSD. It shows that the predicted regression box can benefit from the semantic information that improves the overall performance.

5 Training and Inference

Single-stage Detector: Our proposed ZSD framework is specially designed to work with single-stage detectors. The primary motivation is the direct connection between anchor classification and localization that ensures a strong feedback for both tasks. For this study, we choose a recent unified single architecture, RetinaNet (Lin et al. 2018) to implement our proposed method. RetinaNet is the best detector known for its high speed (on par with single-stage detectors) and top accuracy (outperforming two-stage detectors). In Fig. 6, we illustrate the overall architecture of the model. In addition to a novel loss formulation, we also perform modifications to the RetinaNet architecture to link visual features (from ResNet50 (He et al. 2016)) with semantic information. To adapt this network to ZSL setting, we perform simple modifications in both classification and box regression subnets to consider word-vectors (with vocabulary metric) during training (see Fig. 6).

Training: We train the classification subnet branch with our proposed loss defined in Eq. 6. Similar to (Lin et al. 2018), to address the imbalance between hard and easy examples, we normalize the total classification loss (calculated from $\sim$100k anchors) by the total number of object/positive anchor boxes rather than the total number of anchors. We use standard smooth L_1 loss for the box-regression subnet branch. The total loss is the sum of the loss of both branches.

Inference: For seen object detection, a simple forward pass predicts both confidence scores and bounding boxes in the classification and box-regression subnetworks respectively. Note that we only consider a fixed number (e.g., 100) of boxes from RPN having confidence greater than 0.05 for inference. Moreover, we apply Non-Maximum Suppression (NMS) with a threshold of 0.5 to obtain final detections. We select the final detections that satisfy a seen score-threshold (t_s). To detect unseen objects, we use the following equation, followed by an unseen score-thresholding with a relatively lower value ($t_u < t_s$)[2]:

$$\mathbf{p}_u = W_u W_s^T \sigma(\delta(W_s M D)\mathbf{f}) \tag{9}$$

[2]Empirically, we found t_s=0.3 and t_u=0.1 generally work well. t_u is kept smaller to counter classifier bias towards unseen classes due to the lack of visual examples during training.

where, $W_u \in \mathbb{R}^{\mathrm{U}\times d}$ contains unseen class word vectors. For generalized zero-shot object detection (GZSD), we simply consider all detected seen and unseen objects together. In our experiments, we report performances for traditional seen, zero-shot unseen detection and GZSD. One can notice that our architecture predicts a bounding box for every anchor which is independent of seen classes. It enables the network to predict bounding boxes dedicated to unseen objects. Previous attempts like (Rahman, Khan, and Porikli 2018b) detect seen objects first and then attempts to classify those detections to unseen objects based on semantic similarity. By contrast, our model allows detection of unseen bounding boxes that are different to those seen.

Reduced description of unseen: All seen semantics vectors are not necessary to describe an unseen object (Rahman, Khan, and Porikli 2018a). Thus, we only consider the top T predictions, $\mathbf{p}'_v \in \mathbb{R}^T$ from $\sigma(\delta(W_s MD)\mathbf{f})$ and the corresponding seen semantics, $W'_s \in \mathbb{R}^{\mathrm{T}\times d}$ to predict unseen scores. For the reduced case, $\mathbf{p}'_u = W_u W_s'^T \mathbf{p}'_v$. In the experiment, we vary the number of the closest seen T from 5 to S and find that a relatively small value of T (e.g., 5) performs better than using all available $T = \mathrm{S}$ seen word vectors.

6 Experiments

Datasets: We evaluate our method with MS-COCO (2014) (Lin et al. 2014) and Pascal VOC $(2007/12)$ (Everingham et al. 2010). With 80 object classes, MS-COCO includes $82{,}783$ training and $40{,}504$ validation images. For the ZSD task, only unseen class performance is of interest. As the test data labels are not known, the ZSD evaluation is done on a subset of validation data. MS-COCO (2014) has more validation images than any later versions which motivates us to use it. For Pascal VOC, we use the train set of 2007 and 2012 for training and use validation+test set of 2007 for testing.

Issues with existing MS-COCO split: Recently, (Bansal et al. 2018) proposed a split of seen/unseen classes for MS-COCO (2014). It considers $73,774$ training images from 48 seen classes and 6608 test images from 17 unseen classes. The split criteria were the cluster embedding of class semantics and synset WordNet hierarchy (Miller 1995). We identify two practical drawbacks of this split: **(1)** Because all 63 classes are not used as seen, this split does not take full advantage of training images/annotations, **(2)** Because of choosing unseen classes based on wordvector clustering it cannot guarantee the desired diverse nature of the unseen set. For example, this split does not choose any classes from '*outdoor*' super-category of MS-COCO.

Proposed seen/unseen split on MS-COCO: To address these issues, we propose a more realistic split of MS-COCO for ZSD. Following the practical consideration of unseen classes discussed in (Rahman, Khan, and Porikli 2018b) i.e. rarity and diverseness, we follow the following steps: **(1)** We sort classes of each superclass in ascending order based on the total number of instances in the training set. **(2)** For each superclass, we pick 20% rare classes as unseen which results in 15 unseen and 65 seen classes. Note that the superclass information is only used to create a diverse seen/unseen split, and never used during training. **(3)** Being zero-shot, we remove all the images from the training set where at least one unseen class appears to create a training set of $62{,}300$ images. **(4)** For testing ZSD, we select $10{,}098$ images from the validation set where at least one instance of an unseen class is present. The total number of unseen bounding boxes is $16{,}388$. We use both seen and unseen annotation together for this set to perform GZSD. **(5)** We prepare another list of $38{,}096$ images from the validation set where at least one occurrence of the seen instance is present to test traditional detection performance on seen classes. In this paper, we report results on both our and (Bansal et al. 2018) settings. We validate our hyper-parameters on traditional detection task.

Pascal VOC Split: For Pascal VOC 2007/12 (Everingham et al. 2010), we follow the settings of (Demirel, Cinbis, and Ikizler-Cinbis 2018). We use 16 seen and 4 unseen classes from total 20 classes. We utilize 2072 and 3909 train images from Pascal VOC 2007 and 2012 respectively after ignoring images containing any instance of unseen classes. For testing, we use 1402 val+test images from Pascal VOC 2007 where any unseen class appears at least once.

Vocabulary: We choose vocabulary atoms from 5018 Flickr tags in NUS-WIDE (Chua et al.). We only remove MS-COCO class names and tags that have no word vectors. This vocabulary covers a wide variety of objects, attributes, scene types, actions, and visual concepts.

Semantic embedding: For MS-COCO classes and vocabulary words, we use ℓ_2 normalized 300 dimensional unsupervised word2vec (Mikolov et al. 2013), GloVe (Pennington, Socher, and Manning 2014) and FastText (Joulin et al. 2017) vectors obtained from billions of words from unannotated texts like Wikipedia. For Pascal VOC (Everingham et al. 2010) classes, we use average 64 dimension binary per-instance attribute annotation of all training images from aPY dataset (Farhadi et al. 2009). Unless mentioned otherwise, we use word2vec in our experiments.

6.1 Quantitative Results

Compared Methods: We rigorously evaluate our proposed ZSD method on both (Bansal et al. 2018) split $(48/17)$ and our new $(65/15)$ split of MS-COCO. We provide a brief description of all compared methods: **(a)** SB (Bansal et al. 2018): This method extracts pre-trained Inception-ResNet-v2 features from Edge-Box object proposals. It applies a standard max-margin loss to align visual features to semantic embeddings via linear projections. **(b)** DSES (Bansal et al. 2018): In addition to SB, DSES augments extra bounding boxes other than MSCOCO objects. As (Bansal et al. 2018) reported recall performances, we also report recall results (in addition to mAP) to compare with this method. **(c)** Baseline: This method trains an exact RetinaNet model. Thus, it does not use any word vectors during training. To extend this approach to perform ZSD, we apply this formula to calculate unseen scores: $\mathbf{p}'_u{=}W_u W_s'^T \mathbf{p}'_d$ where $\mathbf{p}'_d$ represents top T seen prediction scores for the reduced description of unseen. **(d)** Ours: This method is our final proposal using vocabulary and polarity loss (Fig. 6).

Overall Results: Fig. 7 presents overall performance on ZSD and GZSD tasks across different comparison methods

Method Split in (Bansal et al. 2018) (↓)	Seen / Unseen	ZSD (mAP/RE)	GZSD Seen (mAP/RE)	GZSD Unseen (mAP/RE)	GZSD HM (mAP/RE)
SB (Bansal et al. 2018)	48/17	0.70/24.39	-	-	-
DSES (Bansal et al. 2018)	48/17	0.54/27.19	-/15.02	-/15.32	-/15.17
ZSD-Textual (Li et al. 2019)	48/17	-/34.3	-/-	-/-	-/-
Baseline	48/17	6.99/18.65	40.46/43.69	2.88/17.89	5.38/25.38
Ours	48/17	**10.01/43.56**	35.92/38.24	**4.12/26.32**	**7.39/31.18**
Proposed Split (↓)		mAP/RE	mAP/RE	mAP/RE	mAP/RE
Baseline	65/15	8.48/20.44	**36.96/40.09**	8.66/20.45	14.03/27.08
Ours	65/15	**12.40/37.72**	34.07/36.38	**12.40/37.16**	**18.18/36.76**

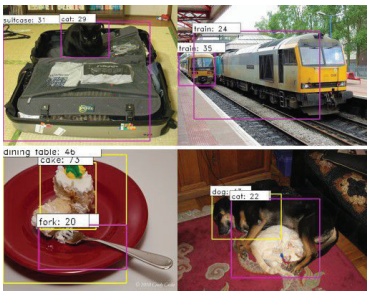

Figure 7: *(left)* Overall performance on MS-COCO. Hyper-parameters are set on the validation set: β=5, IoU=0.5. mAP = mean average precision and RE = recall (@100). The top part shows results on (Bansal et al. 2018) split and the lower part shows results on our proposed split. Ours achieves best performance in terms of mAP on unseen classes. *(Right)* Qualitative examples of ZSD (top row) and GZSD (bottom row). Pink and yellow box represent unseen and seen detections respectively.

Method	Seen	Unseen	aeroplane	bicycle	bird	boat	bottle	bus	cat	chair	cow	d.table	horse	motrobike	person	p.plant	sheep	tvmonitor	*car*	*dog*	*sofa*	*train*
Demirel *et al.*(2018)	57.9	54.5	68.0	**72.0**	**74.0**	48.0	41.0	61.0	48.0	25.0	48.0	**73.0**	**75.0**	71.0	73.0	33.0	59.0	57.0	55.0	82.0	**55.0**	26.0
Ours	**63.5**	**62.1**	**74.4**	71.2	67.0	**50.1**	**50.8**	**67.6**	**84.7**	**44.8**	**68.6**	39.6	74.9	**76.0**	**79.5**	**39.6**	**61.6**	**66.1**	**63.7**	**87.2**	53.2	**44.1**

Table 2: mAP scores of Pascal VOC'07. *Italic* classes are unseen.

γ	α	ZSD	GZSD Seen	GZSD Unseen	GZSD HM
0	1	6.6	31.9	6.6	10.9
0	.75	2.7	27.4	2.7	4.9
0.1	.75	5.4	27.9	5.4	9.0
0.2	.75	7.3	31.4	7.3	11.8
0.5	.50	8.4	30.6	8.4	13.1
1.0	.25	11.6	31.3	11.6	16.9
2.0	.25	**12.6**	33.0	**12.6**	**18.3**
5.0	.25	9.1	**33.6**	9.1	14.3

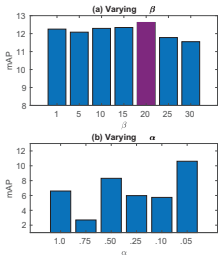

Figure 8: Parameter sensitivity analysis: *(Left)* Varying α and γ with a fixed β=20. *(Right-a)* Impact of varying β, *(Right-b)* varying α with γ=0 to see the behavior of our loss with only balanced CE. Note that the actual hyper-parameters choice is made on a validation set.

Method	ZSD	GZSD Seen	GZSD Unseen	GZSD HM
Baseline	8.48	**36.96**	8.66	14.03
Our-FL-word	10.80	37.56	10.80	16.77
Our-PL-word	12.02	33.28	12.02	17.66
Our-PL-vocab*	**12.62**	32.99	**12.62**	**18.26**

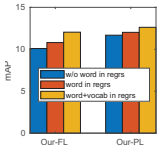

Figure 9: Ablation studies with $\beta = 20$: *(Left)* Comparison of different variant of our approach, best method denoted with $*$. *(Right)* Impact of word-vectors in the regression branch.

with two different seen/unseen split of MS-COCO. In addition to mAP, we also report recall (RE) to compare with (Bansal et al. 2018). With 48/17 settings, our method (and baseline) beats (Bansal et al. 2018) (SB and DSES) in both the ZSD and GZSD by a significantly large margin. Similarly, in 65/15 split, we outperform our baseline by a margin 3.92 mAP (12.40 vs. 8.48) in ZSD task and 4.15 harmonic-mAP in GZSD task (18.18 vs. 14.03). This improvement is the result of end-to-end learning, the inclusion of the vocabulary metric to update word vectors and the proposed loss in our method.

Hyper-parameter Sensitivity Analysis: We study the sensitivity of our model to loss hyper-parameters γ, α and β. First, we vary $\gamma \in [0, 5]$ and $\alpha \in [.25, 1]$ keeping β=20. In Fig. 8 (*left*), we report mAP using different parameter settings for ZSD and GZSD. Our model works best with α=.25 and γ=2.0 which are also the recommended values in FL. We also vary β from 1-30 to see its effect on ZSD in Fig. 8 (*Right-a*). This parameter controls the steepness of the penalty function f_p in Eq. 5. Notably β=20 provides correct steepness to estimate a penalty for incorrect predictions. Our loss can also work reasonably well with balanced CE (i.e., without FL when γ=0). We show this in Fig. 8(*Right-b*). With a low α of 0.05, our method can achieve around 10% mAP. It shows that our penalty function can effectively balance object/background and easy/hard cases.

Ablation Studies: In Fig. 9*(Left)*, we report results on different variants of our method. Our-FL-word: This method is based on the architecture in Fig. 4(b) and trained with focal loss. It uses static word vectors during training. But, it cannot update vectors based on visual features. Our-PL-word: Same architecture as of Our-FL-word but training is done with our proposed polarity loss. Our-PL-vocab: The method uses our proposed framework in Fig. 6 with vocabulary metric learning in the custom layer and is learned with polarity loss. Our observations: **(1)** Our-FL-word works better than Baseline for ZSD and GZSD because the former uses word vectors during training whereas the later does not adopts semantics. By contrast, in GZSD-seen detection cases, Base-

line outperforms Our-FL-word because the use of unsupervised semantics (word vectors) during training in Our-FL-word introduces noise in the network which degrades the seen mAP. **(2)** From Our-FL-word to Our-PL-word unseen mAP improves because of the proposed loss which increases inter-class and reduces intra-class differences. It brings better visual-semantic alignment than FL (Fig. 2). **(3)** Our-PL-vocab further improves the ZSD performance. Here, the vocabulary metric helps the word vectors to update based on visual similarity and allows features to align better with semantics.

Semantics in Box-regression Subnet: Our framework can be trained without semantics in the box-regression subnet. In Fig. 9 *(Right)*, we compare performance with and without using word vectors in the regression branch using FL and our loss. We observe that using word vectors in regression branch helps to improve the performance of ZSD.

Pascal VOC results: To compare with YOLO-based ZSD Demirel *et al.*(2018), we adopt their exact settings with Pascal VOC 2007 and 2012. Note that, their approach used attribute vectors as semantics from (Farhadi et al. 2009). As such attribute vectors are not available for our vocabulary list, we compare this approach with only using fixed attribute vectors inside our network. Our method beats Demirel *et al.*(2018) by a large margin (57.9 vs 63.5 on traditional detection, 54.5 vs 62.1 on unseen detection).

7 Conclusion

In this paper, we propose an end-to-end trainable framework of ZSD which includes a novel loss formulation and a new vocabulary metric. Our proposed polarity loss penalizes an example considering background vs. object imbalance, easy vs. hard cases and inter-class vs. intra-class relations. Moreover, our method learns a vocabulary metric to reshape the semantic embedding space so that word vectors become well-distributed and visually similar classes reside close together in the embedding space. Also, we propose a realistic seen-unseen split on the MS-COCO dataset to evaluate ZSD methods. In our experiments, we have outperformed several recent state-of-the-art methods on both ZSD and GZSD tasks across the MS-COCO and Pascal VOC 2007 datasets.

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

References

Al-Halah, Z., and Stiefelhagen, R. 2017. Automatic discovery, association estimation and learning of semantic attributes for a thousand categories. In *CVPR*.

Al-Halah, Z.; Tapaswi, M.; and Stiefelhagen, R. 2016. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *CVPR*.

Bansal, A.; Sikka, K.; Sharma, G.; Chellappa, R.; and Divakaran, A. 2018. Zero-shot object detection. In *ECCV*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. Nus-wide: A real-world web image database from national university of singapore. In *IVR*. ACM.

Demirel, B.; Cinbis, R. G.; and Ikizler-Cinbis, N. 2018. Zero-shot object detection by hybrid region embedding. In *BMVC*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2):303–338.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *CVPR*, 1778–1785.

Fu, Y.; Xiang, T.; Jiang, Y.; Xue, X.; Sigal, L.; and Gong, S. 2018. Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine* 35(1):112–125.

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

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of tricks for efficient text classification. In *EACL*.

Li, Z.; Yao, L.; Zhang, X.; Wang, X.; Kanhere, S.; and Zhang, H. 2019. Zero-shot object detection with textual descriptions. *AAAI* 33(01):8690–8697.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Lin, T.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2018. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1–1.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *NIPS*. Curran Associates, Inc. 3111–3119.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Rahman, S., and Khan, S. 2018. Deep multiple instance learning for zero-shot image tagging. In *ACCV*.

Rahman, S.; Khan, S.; and Barnes, N. 2019. Transductive learning for zero-shot object detection. In *ICCV*.

Rahman, S.; Khan, S.; and Porikli, F. 2018a. A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing* 27(11):5652–5667.

Rahman, S.; Khan, S.; and Porikli, F. 2018b. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*.

Redmon, J., and Farhadi, A. 2017. Yolo9000: Better, faster, stronger. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6):1137–1149.

Van Der Maaten, L. 2014. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research* 15(1):3221–3245.

Zhu, P.; Wang, H.; Bolukbasi, T.; and Saligrama, V. 2018. Zero-shot detection. *arXiv preprint arXiv:1803.07113*.

Zitnick, C. L., and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *ECCV*, 391–405. Cham: Springer International Publishing.