

# Multi-label Zero-shot Classification by Learning to Transfer from External Knowledge

He Huang<sup>1</sup>

hehuang@uic.edu

Yuanwei Chen<sup>2</sup>

liaoyuan.zhangfei@alibaba-inc.com

Wei Tang<sup>1</sup>

tangw@uic.edu

Wenhao Zheng<sup>2</sup>

zwh149850@alibaba-inc.com

Qing-Guo Chen<sup>2</sup>

qingguo.cqg@alibaba-inc.com

Philip Yu<sup>1</sup>

psyu@uic.edu

<sup>1</sup> Department of Computer Science,  
University of Illinois at Chicago,  
Chicago, USA

<sup>2</sup> Alibaba Group,  
Hangzhou, China

---

## Abstract

Multi-label zero-shot classification aims to predict multiple unseen class labels for an input image. It is more challenging than its single-label counterpart. On one hand, the unconstrained number of labels assigned to each image makes the model more easily overfit to those seen classes. On the other hand, there is a large semantic gap between seen and unseen classes in the existing multi-label classification datasets. To address these difficult issues, this paper introduces a novel multi-label zero-shot classification framework by learning to transfer from external knowledge. We observe that ImageNet is commonly used to pretrain the feature extractor and has a large and fine-grained label space. This motivates us to exploit it as external knowledge to bridge the seen and unseen classes and promote generalization. Specifically, we construct a knowledge graph including not only classes from the target dataset but also those from ImageNet. Since ImageNet labels are not available in the target dataset, we propose a novel PosVAE module to infer their initial states in the extended knowledge graph. Then we design a relational graph convolutional network (RGCN) to propagate information among classes and achieve knowledge transfer. Experimental results on two benchmark datasets demonstrate the effectiveness of the proposed approach.

## 1 Introduction

Recent deep learning methods are able to outperform humans in image classification tasks by leveraging supervised learning and large-scale datasets such as ImageNet [25]. However, machines still cannot compete with humans when the task requires generalizing learned concepts to novel concepts. For example, a person who has only seen horses but not zebras

can still recognize a zebra if he/she is told that “a zebra is like a horse but has black and white strips”. To evaluate a model’s ability to generalize what it has learned during training to totally new concepts in testing, the task of zero-shot learning (ZSL) [0, 3, 12, 26, 33] divides a set of classes into two disjoint sets: a set of *seen* classes and a set of *unseen* classes. Images from *seen* classes are provided during training, while images of both *seen* and *unseen* classes are used for testing.

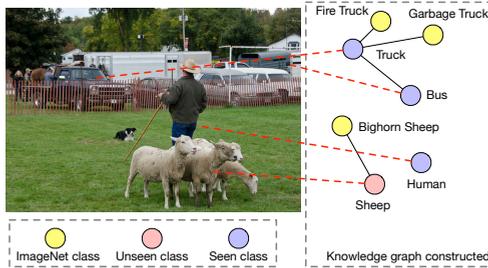


Figure 1: Illustration of combining ImageNet classes into multi-label zero-shot image classification on MS-COCO [18]. Without the yellow nodes from ImageNet [25], it would be hard to predict the unseen class *Sheep* since other seen classes are not semantically related to it. However, with the knowledge that the current image has a high probability on the ImageNet class *Bighorn Sheep* and the word embeddings of both *Sheep* and *Bighorn Sheep*, the model can better predict the unseen class by referring to the ImageNet classes.

Although zero-shot learning for single-label classification has been well studied [12, 22, 26, 28, 32, 35], multi-label zero-shot learning (ML-ZSL) [0, 16, 20, 37] is a less explored area. An image is associated with potentially multiple seen and unseen classes, but only labels of seen classes are provided during training. The multi-label zero-shot learning problem is more difficult than its single-label counterpart for two reasons. First, since there is no constraint on the number of labels assigned to an image and only labels of seen classes are provided during training, the model can easily bias towards the seen classes and ignore the unseen ones. Second, the datasets used for multi-label zero-shot learning are generally much more challenging than those used for traditional zero-shot learning. While the datasets used for zero-shot learning (ZSL) usually consist of closely related classes such as different kinds of birds (e.g., *Baird Sparrow* and *Chipping Sparrow* in CUB [29]), the datasets for multi-label classification contain high-level concepts that are less related to each other (e.g., *truck* and *sheep* in MS-COCO [18]). The semantic gap between seen and unseen classes in multi-label datasets makes it difficult for the model to generalize knowledge learned from seen classes to predict unseen classes.

Some studies [0, 3, 16, 19, 20, 24, 37] try to solve the multi-label zero-shot learning problem by leveraging word embeddings [37] or constructing structured knowledge graphs [16]. Although these methods empirically work well, none of them utilizes external knowledge to close the gap between semantically distinct seen and unseen classes. By contrast, humans learn new concepts by connecting them with those that they have already learned before, and thus can transfer knowledge from previous experience to solve new tasks. This ability is called *transfer learning*. The most popular way to use transfer learning in various computer vision tasks is to extract image features from classifiers (e.g., ResNet [11]) pretrained on ImageNet [25]. However, this naive approach does not make use of the semantic information of the 1K ImageNet classes, which can be helpful in recognizing unseen classes in zero-shot

learning, especially when none of the seen classes is closely related to the unseen ones.

In this paper, we propose to build a knowledge graph with not only the classes from the target multi-label dataset (e.g., *MS-COCO* [13]) but also classes from the external single-label dataset (e.g., *ImageNet* [2]). Then classification can be performed via graph reasoning methods such as graph convolutional networks (GCNs) [15]. Our intuition is that the knowledge of which ImageNet classes are more related to the image can help close the semantic gap between the seen and unseen classes. For example, as illustrated in Figure 1, the unseen class *Sheep* in the image is hard to predict since it is semantically very different from all seen classes in the dataset. But if we include the 1K ImageNet classes into the knowledge graph, there will be a class *Bighorn Sheep* which is semantically related to *Sheep*. Given the predicted probability distribution on the ImageNet classes, if the *Bighorn Sheep* class has a high probability, we can infer that the image contains the *Sheep* class.

After incorporating the ImageNet classes as nodes in the knowledge graph of the target dataset, it is still nontrivial to infer their states. A simple solution is to use the predicted probability distribution on ImageNet classes as pseudo labels, and then we can treat the ImageNet classes the same as those in the target multi-label dataset. However, this simple approach is problematic. Since the ImageNet classes are not labeled in the multi-label dataset, treating them the same as the real target labels may confuse the network and lead to inferior learning. To address this issue, we propose a novel PosVAE module based on a conditional variational auto-encoder [7] to model the posterior probabilities of the ImageNet classes. Then the states of ImageNet nodes are obtained via inference through the PosVAE network. The states of nodes from the target dataset are obtained by another semantically conditioned encoder which takes as input the concatenation of the image features and the class embedding of each node. The states of all nodes are passed into a relational graph convolutional network (RGCN), which propagates information among classes and generates predictions for each class in the target dataset.

Our contributions are summarized as follows:

- To the best of our knowledge, this is the first attempt to transfer the semantic knowledge from the 1K ImageNet classes to help solve the multi-label zero-shot learning problem. This is achieved by adding the ImageNet classes as additional nodes to extend the knowledge graph constructed by labels in the target multi-label dataset. By contrast, previous methods only utilize pretrained ImageNet classifiers as feature extractors.
- We design a PosVAE module, which is based on a conditional variational auto-encoder [7], to infer the states of ImageNet nodes in the extended knowledge graph, and a relational graph convolutional network (GCN) to generate predictions for seen and unseen classes.
- We conduct extensive experiments and demonstrate that the proposed approach can effectively transfer knowledge from the ImageNet classes and improve the multi-label zero-shot classification performance. We also show that it outperforms previous methods.

## 2 Related Work

**Multi-label Classification.** Recently, some work [4, 5, 14, 10, 31, 33] propose different ways to learn the correlations among labels. Marino *et al.* [14] first detect one label and then sequentially predict more related labels of the input image. Wang *et al.* [30] use a recurrent neural network (RNN) to capture the dependencies between labels, while Zhu *et al.* [33] utilize spatial attention to better capture spatial and semantic correlations of labels. Chen *et al.* [5] construct a knowledge graph from labels and use the co-occurrence probabilities of labels as weights of the edges in the graph, and apply a graph convolutional network (GCN) [15]

to perform multi-label classification. Chen *et al.* [9] use the pretrained word embeddings of labels as semantic queries on the image feature map and apply spatial attention to obtain semantically attended features for each label. Although these methods consider label dependencies and work well on traditional multi-label classification, they are not directly applicable in the multi-label zero-shot setting because the label co-occurrence statistics is not available for unseen classes, and only utilizing the statistics for seen labels can cause severe bias towards seen classes. Thus we need a different way to learn the relations between labels in the multi-label zero-shot setting.

**Multi-label Zero-shot Learning** is a less explored problem compared to single-label zero-shot learning [12, 22, 26, 28, 54]. Fu *et al.* [7] enumerate all possible combinations of labels and thus transform the problem into single-label zero-shot learning. However, as the number of labels increases, the number of label combinations will increase exponentially, which makes this method inapplicable. Mensink *et al.* [20] learn classifiers for unseen classes as weighted combinations of seen classifiers learned with co-occurrence statistics of seen labels. Zhang *et al.* [57] consider the projected image features as a projection vector which ranks relevant labels higher than irrelevant ones by calculating the inner product between the projected image features and label word embeddings. Gaure *et al.* [8] further assume that the co-occurrence statistics of both seen and unseen labels are available and design a probabilistic model to solve the problem. Ren *et al.* [24] cast the problem into a multi-instance learning [56] framework and learn a joint latent space for both image features and label embeddings. Lee *et al.* [16] construct a knowledge graph for both seen and unseen labels, and learn a gated graph neural network (GGNN) [17] to propagate predictions among labels. To our knowledge, none of the previous work considers the large semantic gap between seen and unseen labels, and we are the first to exploit the label space of a single-label image classification dataset as external knowledge and transfer it for zero-shot multi-label image classification.

## 3 Approach

### 3.1 Problem Definition and Notations

The multi-label zero-shot classification problem is defined as follows. Given a multi-label classification dataset, the label space  $\mathcal{Y}$  is split into seen labels  $\mathcal{Y}^s$  and unseen labels  $\mathcal{Y}^u$  such that  $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$  and  $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$ . During training, a set of image-label pairs is available, where the labels  $\mathbf{y}$  of an image  $\mathbf{x}$  come from only the seen classes:  $\{\mathbf{x}, \mathbf{y} | \mathbf{y} \subseteq \mathcal{Y}^s\}$ . During testing, the images may have labels from both seen and unseen classes:  $\{\mathbf{x}, \mathbf{y} | \mathbf{y} \subseteq \mathcal{Y}\}$ . The goal is to learn a model  $f : \mathbf{x} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$ , given only seen classes  $\mathcal{Y}^s$  during training. In this work, we propose to use an external set of labels  $\mathcal{Y}^a$  from ImageNet [25] where  $\mathcal{Y}^a \cap \mathcal{Y} = \emptyset$ . For an arbitrary class  $y$ , its pretrained GloVe [23] embedding is denoted by  $\mathbf{s}_y$ , and all class embeddings are aggregated into a single matrix  $\mathbf{S} \in \mathbb{R}^{(|\mathcal{Y}|+|\mathcal{Y}^a|) \times d_s}$ , where  $d_s$  is the dimension of embeddings.

### 3.2 Overview of the Proposed Method

The overall pipeline of the proposed model is illustrated in Figure 2. An input image is first fed into a pretrained ImageNet classifier to extract its image feature and a probability distribution  $\mathbf{p}^a$  over the ImageNet classes  $\mathcal{Y}^a$ . We first construct a graph whose nodes are classes from the target dataset. We calculate the pair-wise Wu-Palmer (WUP) [52] similarities on

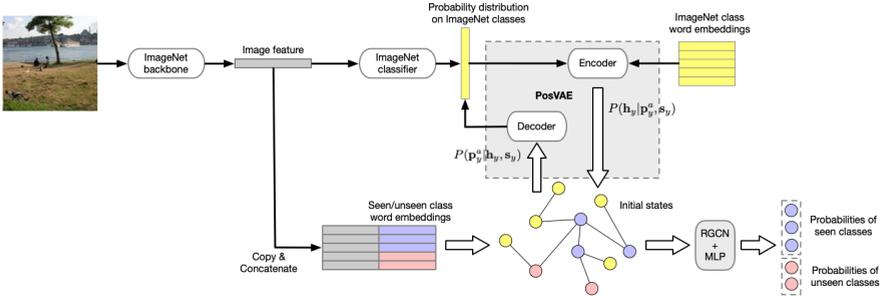


Figure 2: Pipeline of the proposed model. When constructing the graph, we calculate the WUP [52] similarity between each pair of nodes (one node per class) and use a threshold to decide whether there is an edge between the two nodes. Since WUP similarity is invariant to the order of nodes, edges are bidirectional. We hide the self-connections here for clarity.

WordNet [21] and then use a threshold to determine whether an edge exists between a pair of nodes. Self-connections are also included. During training, only seen classes  $\mathcal{Y}^s$  are included into the training graph, while during testing both seen and unseen classes ( $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$ ) are included in the testing graph. The classes  $\mathcal{Y}^a$  from ImageNet [25] are also added to the training and testing graphs, and edges are added in the same way as described above. The overlapping classes between ImageNet and the target dataset have been removed from  $\mathcal{Y}^a$ . All aggregated edges form an adjacency matrix  $\mathbf{A} \in \mathbb{R}^{(|\mathcal{Y}^s|+|\mathcal{Y}^a|) \times (|\mathcal{Y}^s|+|\mathcal{Y}^a|)}$  for training, and an adjacency matrix  $\mathbf{A} \in \mathbb{R}^{(|\mathcal{Y}|+|\mathcal{Y}^a|) \times (|\mathcal{Y}|+|\mathcal{Y}^a|)}$  for testing (with little abuse of notation).

Since the target dataset does not have ground-truth labels of the ImageNet classes  $\mathcal{Y}^a$ , we need to treat  $\mathcal{Y}$  and  $\mathcal{Y}^a$  differently when inferring their states in the graph. For nodes in  $\mathcal{Y}^s \cup \mathcal{Y}^u$ , we infer their initial states by concatenating their corresponding word embeddings and the extracted image features. Then a multi-layer perceptron (MLP) network is applied to obtain the node features  $\mathbf{H}^d \in \mathbb{R}^{|\mathcal{Y}^s| \times d_h}$  for training and  $\mathbf{H}^d \in \mathbb{R}^{|\mathcal{Y}| \times d_h}$  for testing, where  $d_h$  is the dimension of node features. Meanwhile, for nodes corresponding to  $\mathcal{Y}^a$ , we design a PosVAE module to estimate the posterior  $P(\mathbf{h}_y | \mathbf{p}_y^a, \mathbf{s}_y)$  for each  $y \in \mathcal{Y}^a$ , where  $\mathbf{h}_y$  is the node state of class  $y$  and  $\mathbf{p}_y^a$  is predicted probability of the class  $y \in \mathcal{Y}^a$ . Then we sample from this posterior distribution to infer the initial node states for all classes in  $\mathcal{Y}^a$  denoted as  $\mathbf{H}^a \in \mathbb{R}^{|\mathcal{Y}^a| \times d_h}$ . We will discuss the detailed design of PosVAE in Section 3.3. Then we can obtain the initial node states for all the nodes in the graph by concatenation:  $\mathbf{H} \in \mathbb{R}^{(|\mathcal{Y}^s|+|\mathcal{Y}^a|) \times d_h}$  for training and  $\mathbf{H} \in \mathbb{R}^{(|\mathcal{Y}|+|\mathcal{Y}^a|) \times d_h}$  for testing.

With the initial node states  $\mathbf{H}$ , the adjacency matrix  $\mathbf{A}$ , and the embedding matrix  $\mathbf{S}$ , we design a Relational Graph Convolutional Network (RGCN) to propagate information among the nodes and refine their features, which will be discussed in Section 3.4. The output features of RGCN are fed into another MLP network followed by Sigmoid functions to predict the existence of each label in the target dataset ( $\mathcal{Y}^s$  for training and  $\mathcal{Y}$  for testing).

### 3.3 Variational Inference for Node States of $\mathcal{Y}^a$

For each ImageNet class  $y$ , we aim to estimate its corresponding initial node state  $\mathbf{h}_y$  in the graph from its probability  $\mathbf{p}_y^a$  predicted by pretrained ImageNet classifier and its embedding  $\mathbf{s}_y$ . The objective is to estimate the posterior  $P(\mathbf{h}_y | \mathbf{p}_y^a, \mathbf{s}_y)$ . Since we do not have the exact form of the posterior, we resort to variational inference and use another learned distribution  $Q(\mathbf{h}_y | \mathbf{p}_y^a, \mathbf{s}_y)$  to approximate the true posterior  $P(\mathbf{h}_y | \mathbf{p}_y^a, \mathbf{s}_y)$ . In order to ensure a good approx-

imation, we aim to minimize the KL-divergence between  $Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)$  and  $P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)$ :

$$\mathcal{L}_{VAE} = \frac{1}{|\mathcal{Y}^a|} \sum_{y \in \mathcal{Y}^a} D_{KL}[Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y) || P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)] \quad (1)$$

$$= \frac{1}{|\mathcal{Y}^a|} \sum_{y \in \mathcal{Y}^a} \mathbb{E}_{\mathbf{h}_y \sim P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)} [\log Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y) - \log P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)] \quad (2)$$

Similar to the conditional variational auto-encoder [27], the above objective can be further derived as:

$$\mathcal{L}_{VAE} = -\frac{1}{|\mathcal{Y}^a|} \sum_{y \in \mathcal{Y}^a} [\mathbb{E}_{\mathbf{h}_y \sim P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)} [\log P(\mathbf{p}_y^a|\mathbf{h}_y, \mathbf{s}_y) - \log Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y) + \log P(\mathbf{h}_y)] - P(\mathbf{p}_y^a)] \quad (3)$$

$$= -\frac{1}{|\mathcal{Y}^a|} \sum_{y \in \mathcal{Y}^a} [\mathbb{E}_{\mathbf{h}_y \sim P(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y)} [\log P(\mathbf{p}_y^a|\mathbf{h}_y, \mathbf{s}_y)] - D_{KL}[Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y) || P(\mathbf{h}_y)]], \quad (4)$$

where the term  $P(\mathbf{p}_y^a)$  is neglected since it does not depend on  $\mathbf{h}_y$  during optimization.

We assume  $\mathbf{h}_y$  follows an isotropic Gaussian distribution  $Q(\mathbf{h}_y|\mathbf{p}_y^a, \mathbf{s}_y) \sim \mathcal{N}(\mu, \sigma^2 \mathbf{I})$  and  $P(\mathbf{h}_y) \sim \mathcal{N}(0, 1)$ . By using the reparameterization trick [24], the initial node states  $\mathbf{h}_y$  can be derived as:

$$\mathbf{z} \sim \mathcal{N}(0, 1); (\mu, \log \sigma) = \text{Encoder}(\mathbf{p}_y^a, \mathbf{s}_y); \mathbf{h}_y = \mu + \exp(\log \sigma / 2) \odot \mathbf{z}, \quad (5)$$

where the Encoder network is a stack of MLP layers and  $\odot$  is element-wise multiplication. The Encoder first projects  $\mathbf{p}_y^a$  into the same dimension space as  $\mathbf{s}_y$ . Then the two vectors are concatenated and fed into subsequent MLP layers to generate the mean  $\mu \in \mathbb{R}^{d_h}$  and log standard deviation  $\log \sigma \in \mathbb{R}^{d_h}$ . In order to calculate the log-likelihood term  $\log P(\mathbf{p}_y^a|\mathbf{h}_y, \mathbf{s}_y)$ , we assume a Gaussian distribution on  $\mathbf{p}_y^a$ , and use the Decoder network to project the concatenation of  $\mathbf{h}_y$  and  $\mathbf{s}_y$  back to  $\mathbf{p}_y^a$ . Then maximizing the log-likelihood term is approximately equal to minimizing the mean-square error (MSE) between the reconstructed probability and the real one.

### 3.4 Relational Graph Convolutional Network

The Graph Convolutional Network (GCN) [15] was first introduced as a method to perform semi-supervised classification. Its core idea is to design a message passing mechanism for graph data so that the node features can be updated recursively and become more suitable to the desired task. Different from the conventional convolution that operates in the Euclidean space like images, a GCN works on graph-structure data that contain nodes and edges but without fixed geometric layout. Given the states  $\mathbf{H}^l$  of the current layer  $l$  and the adjacency matrix  $\mathbf{A}$ , the states of the  $(l+1)$ -th layer can be calculated as:

$$\mathbf{H}^{l+1} = \delta(\hat{\mathbf{A}}\mathbf{H}^l\mathbf{W}^l), \quad (6)$$

where  $\mathbf{W}^l$  is the learnable weight matrix of the  $l$ -th layer,  $\hat{\mathbf{A}}$  is the normalized version of  $\mathbf{A}$  [15], and  $\delta(\cdot)$  is a non-linear activation function such as ReLU.

In traditional multi-label classification models, the  $\mathbf{A}$  matrix is often represented by the co-occurrence probabilities between labels [4, 5]. However, in the zero-shot learning setting, we do not have this information. Thus we set  $\mathbf{A} \in \mathbb{R}^{(|\mathcal{Y}^s|+|\mathcal{Y}^a|) \times (|\mathcal{Y}^s|+|\mathcal{Y}^a|)}$  as binary adjacency

matrix and learn a new matrix  $\hat{\mathbf{W}} \in \mathbb{R}^{(|\mathcal{Y}^s|+|\mathcal{Y}^a|) \times (|\mathcal{Y}^s|+|\mathcal{Y}^a|)}$  to indicate the weight of each edge ( $\mathbf{A}, \hat{\mathbf{W}} \in \mathbb{R}^{(|\mathcal{Y}^s|+|\mathcal{Y}^a|) \times (|\mathcal{Y}^s|+|\mathcal{Y}^a|)}$  for testing). In order to learn the weight matrix, we propose the relational graph convolutional network (RGCN) as follows. Specifically, the weight of the edge between a pair of classes  $(u, v)$  is calculated by first concatenating their word embeddings  $(\mathbf{s}_u, \mathbf{s}_v)$  together and feeding them into an MLP network to generate a scalar which represents the weight. In this way, we are able to calculate the weight between seen and unseen classes. An entry  $w_{uv}$  of the weight matrix  $\hat{\mathbf{W}}$  is derived by:

$$w_{uv} = \text{MLP}(\mathbf{s}_u, \mathbf{s}_v), \quad (7)$$

where  $w_{uv} \in [0, 1]$ , and  $(\mathbf{s}_u, \mathbf{s}_v) \in \mathbb{R}^{d_s}$ . Elements on the diagonal of  $\hat{\mathbf{W}}$  are set to 1 to include self-connections. After obtaining the weight matrix  $\hat{\mathbf{W}}$ , we calculate  $\hat{\mathbf{A}} = \text{normalize}(\hat{\mathbf{W}} \odot \mathbf{A})$ , where  $\odot$  is element-wise multiplication. After obtaining the new matrix  $\hat{\mathbf{A}}$ , the rest of RGCN is the same as the traditional GCN [15]. After obtaining the node states  $\mathbf{H}^s \in \mathbb{R}^{|\mathcal{Y}^s| \times d_h}$  of each node corresponding to the seen classes  $\mathcal{Y}^s$ , we apply a final MLP layer to the node states to calculate the probability of each label, *i.e.*,  $\mathbf{p}^s = \text{MLP}(\mathbf{H}^s), \mathbf{p}^s \in \mathbb{R}^{|\mathcal{Y}^s|}$ , and  $L$  is the last layer of RGCN. We train the relational graph convolutional network (RGCN) with class-wise binary cross-entropy loss:

$$\mathcal{L}_{RGCN} = -\frac{1}{|\mathcal{Y}^s|} \sum_{y \in \mathcal{Y}^s} [(1 - \mathbf{I}(y)) \log(1 - \mathbf{p}_y^s) + \mathbf{I}(y) \log \mathbf{p}_y^s], \quad (8)$$

where  $\mathbf{I}(y)$  is an indicator function that outputs 1 if the input image contains label  $y$ , otherwise 0. The whole model is trained by minimizing the joint loss function:

$$\mathcal{L}_{all} = \mathcal{L}_{RGCN} + \lambda \mathcal{L}_{VAE}, \quad (9)$$

where  $\lambda$  is a hyper-parameter to balance the two terms.

## 4 Experiments

### 4.1 Datasets and Setting

**Datasets.** In our experiments, we use two multi-label datasets, *i.e.*, MS-COCO [18] and NUS-WIDE [6]. MS-COCO [18] is a large-scale dataset commonly used for multi-label image classification, object detection, instance segmentation and image captioning. We adopt the 2014 challenge which contains 79,465 training images and 40,137 testing images respectively (after removing images without labels). Among its 80 classes, we randomly select 16 unseen classes and make sure that the unseen classes do not overlap with the ImageNet classes [23]. The 16 unseen classes are ('bicycle', 'boat', 'stop sign', 'bird', 'backpack', 'frisbee', 'snowboard', 'surfboard', 'cup', 'fork', 'spoon', 'broccoli', 'chair', 'keyboard', 'microwave', 'vase'). NUS-WIDE [6] is a web-crawled dataset with 54,334 training images and 42,486 testing images (after removing images without labels). Among the 81 classes of NUS-WIDE [6], we randomly select 16 unseen classes and make sure that the unseen classes do not overlap with the ImageNet classes [23]. The 16 unseen classes are ('airport', 'cars', 'food', 'fox', 'frost', 'garden', 'mountain', 'police', 'protest', 'rainbow', 'sun', 'tattoo', 'train', 'water', 'waterfall', 'window').

**Graph construction.** We calculate the WUP [62] similarity between each pair of classes, and an edge is added to the graph if the corresponding WUP similarity is greater than a

Table 1: Performance comparison on MS-COCO [13] and NUS-WIDE [6]

	MS-COCO				NUS-WIDE			
	Seen		Unseen		Seen		Unseen	
Model	mAP	miAP	mAP	miAP	mAP	miAP	mAP	miAP
visual-semantic	58.30	59.87	18.60	28.08	<b>25.79</b>	34.64	<b>12.73</b>	29.01
fast0tag	44.60	60.20	19.10	32.90	25.10	<b>36.09</b>	10.41	34.58
SKG	<b>64.02</b>	<b>64.22</b>	18.03	33.68	22.89	32.95	8.89	33.77
RGCN	53.67	60.43	18.97	33.65	24.95	31.23	9.98	33.61
RGCN-XL	34.10	49.18	17.32	35.43	23.66	30.95	9.86	34.31
RGCN-PosVAE	51.54	57.13	<b>20.52</b>	<b>37.14</b>	24.67	32.09	11.09	<b>36.78</b>

specified threshold (0.5 in all of our experiments). When adding the ImageNet [25] classes into either the MS-COCO [13] or NUS-WIDE [6] graph, we exclude any class existing in the target dataset from ImageNet to ensure the additional nodes are distinct from the nodes in the two target datasets.

**Evaluation protocol.** We adopt two evaluation metrics, *i.e.*, *mean average precision* (**mAP**) which evaluates the model performance on the class level, and *mean image average precision* (**miAP**) which evaluates the model performance on the image level. We calculate the **mAP** and **miAP** for two sets of labels, *i.e.*, **seen** classes  $\mathcal{Y}^s$  and **unseen** classes  $\mathcal{Y}^u$ .

**Baselines.** We use three models from previous work and some variants of our proposed method as baselines. **Visual-semantic** [1] was proposed for single-label zero-shot classification, which projects the visual features of images to the semantic space of classes and use the class embeddings as classifiers. Here we change the loss function from multi-class cross-entropy to binary cross-entropy applied to each class individually. **Fast0Tag** [6] uses class embeddings as projection vectors to rank the closeness between images and classes, and applies a triplet-based ranking loss to train the model. **SKG** [16] constructs a structural knowledge graph from labels and applies GGNN [17] to infer the existence of each label. It also uses WUP similarities on WordNet [21] to construct the graph as we do. **RGCN** is a basic version of our proposed model which does not include external nodes from ImageNet [25]. **RGCN-XL** is another variant of our proposed model, where the ImageNet [25] nodes are also included in the graph but treated the same as the other nodes in the target dataset. For ImageNet nodes, we predict their probabilities in the same way as the other classes in the target dataset, but train them with MSE to reconstruct the probability distribution of the pre-trained ImageNet classifier. **RGCN-PosVAE** is our full model which uses PosVAE to infer the hidden states of ImageNet classes and applies RGCN to propagate information among different classes.

**Implementation details.** For SKG [16] we use the code provided by the authors and modify it to fit our pipeline, and we implement all other baselines. We use ResNet-101 [18] pretrained on ImageNet [25] as the feature extractor. We use the 300-dimension pretrained GloVe [23] vectors to represent class embeddings. We implement the encoder and decoder of PosVAE as MLP networks with a hidden layer of size 256, while the relation network that computes  $w_{uv}$  is implemented as a two-layer MLP network with a hidden size 256. We use two layers of RGCN with the feature dimension 256. We use an Adam [19] optimizer and set the initial learning rate as  $5e^{-4}$  and decay it by 0.1 when the loss plateaus.  $\lambda$  is set to 1 for our model. The random seed for all experiments are fixed as 42.

## 4.2 Results

The main results are shown in Table 1. As we can see from the last row of the table, our proposed method achieves the highest miAP on unseen classes in both datasets. We achieve the highest unseen mAP on MS-COCO and second highest unseen mAP on NUS-WIDE [6]. Although our model has 1.64% lower unseen mAP than visual-semantic [10] on NUS-WIDE [6], our unseen miAP is 7.77% higher than visual-semantic [10]. For the MS-COCO [18] dataset, our model performs better than SKG [7] with an obvious margin of about 3.5% unseen miAP and 2.5% unseen mAP. On NUS-WIDE [6], our RGCN-PosVAE outperforms the second best Fast0Tag [57] by around 2% unseen miAP and 0.6% unseen mAP.

Fast0tag [57] achieves higher seen and unseen miAP on both datasets than visual-semantic [10] while visual-semantic [10] performs better on seen mAP than Fast0tag [57] on both datasets. SKG [16] performs better than the previous two baselines on MS-COCO [18], but is slightly worse than Fast0tag [57] on the NUS-WIDE [6]. The RGCN baseline, which is the same as our full model but without incorporating ImageNet nodes into the graph, still outperforms Fast0Tag [57] on MS-COCO, and have comparable result with SKG [16] on NUS-WIDE. By comparing the seen and unseen class performance of each model, we can see that other baselines generally achieve better performance on seen classes than our proposed model, but have lower performance on unseen classes, which indicates that our proposed model is less likely to overfit seen classes and better able to generalize to unseen classes than others.

The basic RGCN model achieves higher unseen mAP than SKG [16] on MS-COCO and slightly lower unseen miAP on NUS-WIDE [6]. But their results are still comparable, which shows that our proposed RGCN itself is already a very strong baseline. RGCN-XL, which is an extended RGCN with ImageNet nodes, achieves higher unseen miAP than RGCN, which shows the potential of utilizing the semantic information of ImageNet classes, although in a simple fashion. It can also be noted that RGCN-XL has much lower performance on seen classes on MS-COCO. The reason may be that the brute-force way of incorporating ImageNet classes will make the predictions on seen classes harder, since the input images do not actually have those ImageNet classes while the model is trying to treat them the same as the classes in the target dataset when initializing their node states. Meanwhile, the proposed RGCN-PosVAE model, which infers the initial states of ImageNet classes using variational inference, achieves higher scores on both seen and unseen classes than RGCN-XL, which demonstrates the advantages of using the proposed method to initialize the hidden states of ImageNet classes.

## 4.3 Ablation Study

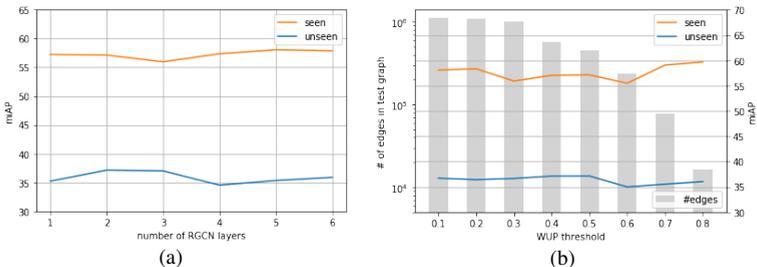


Figure 3: Ablation study

**Effect of the number of GCN layers.** We first investigate how the number of RGCN

layers affects our model’s performance on the MS-COCO dataset [18]. We try different numbers of layers from 1 up to 6, and plot the seen and unseen classes miAP in Figure 3(a). As we can see, the miAP performance on seen classes does not vary a lot with different numbers of RGCN layers. On the other hand, two or three layers of RGCN have similar performance on unseen classes, while the version with only one layer has miAP about 2% lower than the highest. After increasing the number of layers to 4, 5 and 6, the model’s performance on unseen classes drops, which indicates that the model starts to become more overfitted to seen classes.

**Effect of the WUP similarity threshold.** We also investigate how the threshold of WUP similarity can affect our model’s performance. We tune the threshold from 0.1 to 0.8, and show the result as well as the number of edges generated by each threshold in Figure 3(b). As we can see, although setting a low threshold will include some noisy edges that may be misleading, our model still achieves a good performance on unseen miAP, with less than 1% decrease from the best one. As the threshold becomes larger than 0.5, we can see that there is an obvious drop of performance on unseen miAP. This is because the number of edges is much smaller when the threshold is too large, and thus the model cannot make good reference to seen classes when predicting unseen classes. Overall, our model’s performance on seen and unseen classes is still relatively robust with respect to the WUP [52] similarity threshold when the threshold is less than 0.6.

## 5 Conclusion

In this paper, we tackle the multi-label zero-shot learning (ML-ZSL) problem by first pointing out the potential semantic gap between seen and unseen classes, and propose to incorporate external ImageNet classes to help predict the seen can unseen classes. Specifically, we construct a knowledge graph consisting both classes from ImageNet and the target dataset (e.g., MS-COCO [18] and NUS-WIDE [6]), and design a PosVAE network to infer the node states of ImageNet classes, and learn a relational graph convolutional network (RGCN) which calculates the propagation weights between each pair of classes given their GloVe [23] embeddings. Experiments show that our proposed method has a clear advantage of predicting unseen classes. For future research, it will be interesting to apply generative adversarial nets [9, 10] to generate images of unseen classes.

## Acknowledgements

This work is supported in part by NSF under grants III-1526499, III-1763325, III-1909323, and SaTC-1930941.

## References

- [1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

- [3] Binghui Chen and Weihong Deng. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2750–2759, 2019.
- [4] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 522–531, 2019.
- [5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.
- [6] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–9, 2009.
- [7] Yanwei Fu, Yongxin Yang, Tim Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-label zero-shot learning. *BMVC*, 2014.
- [8] Abhilash Gaure, Aishwarya Gupta, Vinay Kumar Verma, and Piyush Rai. A probabilistic framework for zero-shot multi-label learning. In *The Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 1, page 3, 2017.
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] He Huang, Philip S Yu, and Changhu Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*, 2018.
- [12] He Huang, Changhu Wang, Philip S. Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [16] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1576–1585, 2018.

- [17] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.
- [18] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.
- [19] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016.
- [20] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2441–2448, 2014.
- [21] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [22] Ashish Mishra, M Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. *arXiv preprint arXiv:1709.00663*, 2017.
- [23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [24] Zhou Ren, Hailin Jin, Zhe Lin, Chen Fang, and Alan L Yuille. Multiple instance visual-semantic embedding. In *BMVC*, 2017.
- [25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- [26] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019.
- [27] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.
- [28] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

- [30] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.
- [31] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017.
- [32] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.
- [33] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [34] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
- [35] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9384–9393, 2019.
- [36] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.
- [37] Yang Zhang, Boqing Gong, and Mubarak Shah. Fast zero-shot image tagging. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5985–5994. IEEE, 2016.
- [38] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5513–5522, 2017.