OntoZSL: Ontology-enhanced Zero-shot Learning

Yuxia Geng
gengyx@zju.edu.cn
Zhejiang University
Hangzhou, China

Jiaoyan Chen
jiaoyan.chen@cs.ox.ac.uk
University of Oxford
Oxford, United Kingdom

Zhuo Chen
zhuo.chen@zju.edu.cn
Zhejiang University
Hangzhou, China

Jeff Z. Pan
https://knowledge-representation.org/j.z.pan/
University of Edinburgh
Edinburgh, United Kingdom

Zhiquan Ye
yezq@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Zonggang Yuan
yuanzonggang@huawei.com
NAIE CTO Office,
Huawei Technologies Co., Ltd.
Nanjing, China

Yantao Jia
jamaths.h@163.com
Poisson Lab,
Huawei Technologies Co., Ltd.
Beijing, China

Huajun Chen*
huajunsir@zju.edu.cn
College of Computer Science & HIC,
Zhejiang University
AZFT Knowledge Engine Lab

ABSTRACT

Zero-shot Learning (ZSL), which aims to predict for those classes that have never appeared in the training data, has arisen hot research interests. The key of implementing ZSL is to leverage the prior knowledge of classes which builds the semantic relationship between classes and enables the transfer of the learned models (e.g., features) from training classes (i.e., seen classes) to unseen classes. However, the priors adopted by the existing methods are relatively limited with incomplete semantics. In this paper, we explore richer and more competitive prior knowledge to model the inter-class relationship for ZSL via ontology-based knowledge representation and semantic embedding. Meanwhile, to address the data imbalance between seen classes and unseen classes, we developed a generative ZSL framework with Generative Adversarial Networks (GANs).

Our main findings include: (i) an ontology-enhanced ZSL framework that can be applied to different domains, such as image classification (IMGC) and knowledge graph completion (KGC); (ii) a comprehensive evaluation with multiple zero-shot datasets from different domains, where our method often achieves better performance than the state-of-the-art models. In particular, on four representative ZSL baselines of IMGC, the ontology-based class semantics outperform the previous priors e.g., the word embeddings of classes by an average of 12.4 accuracy points in the standard ZSL across two example datasets (see Figure 4).

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**.

*Corresponding author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia
© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450042

KEYWORDS

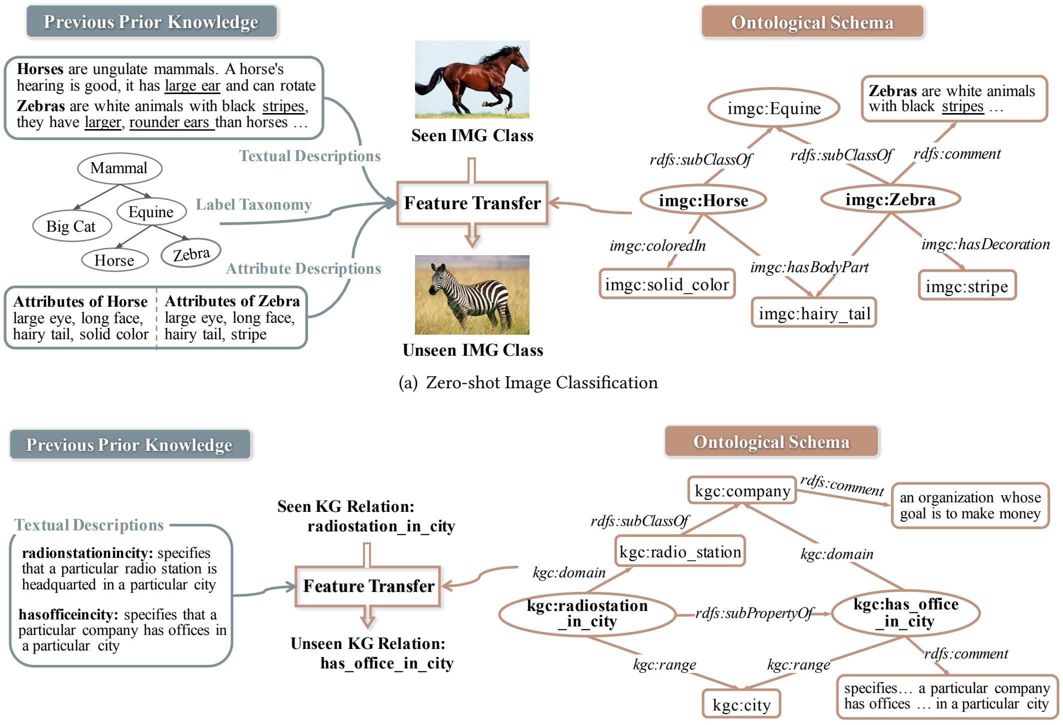
Zero-shot Learning, Ontology, Generative Adversarial Networks, Image Classification, Knowledge Graph Completion

ACM Reference Format:
Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Jeff Z. Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2021. OntoZSL: Ontology-enhanced Zero-shot Learning. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450042

1 INTRODUCTION

Machine learning often operates on a closed world assumption: it trains the model with a number of labeled samples and makes predictions with classes that have appeared in the training stage (i.e., seen classes). For those newly emerging classes, hundreds of samples are needed to be collected and labeled. However, it is impractical to always annotate enough samples and retrain the model for all the emerging classes. Targeting such a limitation, Zero-shot Learning (ZSL) was proposed to handle these novel classes without seeing their training samples (i.e., unseen classes). Over the past few years, ZSL has been introduced in a wide range of machine learning tasks, such as image classification [11, 24, 25, 46], relation extraction [26] and knowledge graph completion [34, 48].

Inspired by the humans' abilities of recognizing new concepts only from their semantic descriptions and previous recognition experience, ZSL aims to develop models trained on data of seen classes and class semantic descriptions to make predictions on unseen classes. These descriptions, also known as *prior knowledge*, provide a prior about the semantic relationships between seen and unseen classes so that the model parameters (e.g., features) learned from seen classes can be transferred to unseen classes. The majority of ZSL methods [11, 31, 34, 53] consider textual descriptions as the priors. For example, [11, 31] project images into the semantic embedding space pre-trained on textual corpora to classify unseen images. [34] generates relation embeddings for unseen relations from their text descriptions for embedding-based knowledge graph


Previous Prior Knowledge
Horses are ungulate mammals. A horse's hearing is good, it has large ear and can rotate
Zebras are white animals with black stripes, they have larger, rounder ears than horses …
Textual Descriptions
Mammal
Big Cat
Equine
Horse
Zebra
Label Taxonomy
Attribute Descriptions
Attributes of Horse large eye, long face, hairy tail, solid color
Attributes of Zebra large eye, long face, hairy tail, stripe
Seen IMG Class
Feature Transfer
Unseen IMG Class
Ontological Schema
imgc:Equine
Zebras are white animals with black stripes …
rdfs:subClassOf
rdfs:subClassOf
rdfs:comment
imgc:Horse
imgc:Zebra
imgc:coloredIn
imgc:hasBodyPart
imgc:hasDecoration
imgc:solid_color
imgc:hairy_tail
imgc:stripe
Previous Prior Knowledge
Textual Descriptions
radionstationincity: specifies that a particular radio station is headquarted in a particular city
hasofficeincity: specifies that a particular company has offices in a particular city
Seen KG Relation: radiostation_in_city
Feature Transfer
Unseen KG Relation: has_office_in_city
Ontological Schema
kgc:company
rdfs:comment
an organization whose goal is to make money
rdfs:subClassOf
kgc:radio_station
kgc:domain
kgc:domain
kgc:radiostation_in_city
rdfs:subPropertyOf
kgc:has_office_in_city
rdfs:comment
kgc:range
kgc:range
kgc:city
specifies… a particular company has offices … in a particular city

(a) Zero-shot Image Classification

(b) Zero-shot Knowledge Graph Completion with Newly-added Relations

Figure 1: Comparison of previously used prior knowledge [Left] and our proposed ontological schemas [Right] in zero-shot image classification and zero-shot knowledge graph completion tasks.

completion tasks. Attribute descriptions [24, 25, 27] and label taxonomy [22, 44] are also widely adopted to model the semantic relationships between classes for zero-shot image classification.

However, all of these priors are limited with incomplete semantics. As shown in the left of Figure 1, in general, textual descriptions may not provide distinguishing characteristics necessary for ZSL due to the noisy words in text [33]; attribute descriptions focus on the "local" characteristics of objects and easily suffer from the domain shift problem [12]; while label taxonomy merely considers the hierarchical relationships between classes. A more detailed discussion of these shortcomings is provided in Section 3.1.

We expect more expressive and competitive prior knowledge to boost the performance of ZSL. In this study, we propose to utilize ontological schema which is convenient for defining the expressive semantics for a given domain [20]. An ontological schema models the general concepts (i.e., types of things) that exist in a domain and the properties that describe the semantic relationships between concepts. It can also represent a variety of valuable information such as concept hierarchy and meta data (e.g., textual definitions, comments and descriptions of concepts), which facilitate modeling richer prior knowledge for ZSL. The right of Figure 1 shows some snapshots of such an ontological schema.

In this paper, we propose a novel ZSL framework called OntoZSL which not only enhances the class semantics with an ontological schema, but also employs an ontology-based generative model to synthesize training samples for unseen classes. Specifically, an ontology embedding technique is first proposed to learn meaningful vector representations of ZSL classes (i.e., class embeddings) from the ontological schema. A generative model e.g., generative adversarial network (GAN) [14] is then adopted to generate features and synthesize training samples for unseen classes conditioned on the class embeddings, empirically turning the ZSL problem into a standard supervised learning problem.

Many ontology embedding methods [6, 16, 18] have been invented and extended from knowledge graph embedding (KGE) techniques [43]. However, adapting existing KGE methods to encode ontologies for ZSL is problematic due to the inherent graph structure and lexical information that both exist in the ontological schema. That is to say, each concept in the ontological schema may have two types of semantics, one is structure-based and captures the multi-relational structures, while the other is text-based and describes the concepts using some natural language tokens, e.g., the concept *kgc:company* in Figure 1. We thus propose a text-aware ontology embedding technique, which can learn the structural and textual representations for each concept simultaneously.

Developing generative models such as GANs for ZSL has been a popular strategy recently, and proven to be more effective and easier to generalize compared with traditional mapping-based ZSL methods (more comparisons are introduced in Section 3.2). However, most of existing generative ZSL methods are merely built upon one type of priors such as textual or attribute descriptions. While in our paper, we propose an ontology-based GAN to incorporate richer priors within ontological schemas to generate more discriminative sample features for ZSL.

We demonstrate the effectiveness of our framework on ZSL tasks from two different domains including a task of image classification (IMGC) from vision and a knowledge graph completion (KGC) task. In IMGC, we build an ontological schema for image classes. While in KGC, we generalize the unseen concepts to unseen relations and build an ontological schema for KG relations. The examples of ontological schemas in two domains are shown in Figure 1.

Our main contributions are summarized as below:

- To the best of our knowledge, this is among the first ones to explore expressive class semantics from ontological schemas in Zero-shot Learning.
- We propose OntoZSL, a novel ontology-guided ZSL framework that not only adopts a text-aware ontology embedding method to incorporate prior knowledge from ontological schemas, but also employs an ontology-based generative model to synthesize training samples for unseen classes.
- In comparison with the state-of-the-art baselines, our framework achieves promising scores on image classification task for standard AwA [46] and constructed ImNet-A and ImNet-O datasets, as well as on knowledge graph completion task for datasets from NELL and Wikidata[1].

2 PRELIMINARIES

We first begin by formally introducing the zero-shot learning in two tasks: image classification (IMGC) and knowledge graph completion (KGC), and their corresponding ontological schemas used.

2.1 Zero-shot Image Classification

Zero-shot learning in image classification is to recognize the new classes whose images are not seen during training. Let $\mathcal{D}_{tr} = \{(x,y)|x \in \mathcal{X}_s, y \in \mathcal{Y}_s\}$ be the training set, where x is the CNN features of a training image, y denotes its class label in $\mathcal{Y}_s$ consisting of seen classes. While the testing set is denoted as $\mathcal{D}_{te} = \{(x,y)|x \in \mathcal{X}_u, y \in \mathcal{Y}_u\}$, where $\mathcal{Y}_u$, the set of unseen classes, has no overlap with $\mathcal{Y}_s$. Suppose that we have class representations $O \in \mathbb{R}^{n\times(|\mathcal{Y}_s|+|\mathcal{Y}_u|)}$ learned from semantic descriptions for $|\mathcal{Y}_s|$ seen classes and $|\mathcal{Y}_u|$ unseen classes, the task of zero-shot IMGC is to learn a classifier for each unseen class given $\{\mathcal{D}_{tr}, O\}$ for training. These representations can be provided as binary/numerical attribute vectors, word embeddings/RNN features or class embeddings learned from our ontological schema. We study two settings at the testing stage: one is standard ZSL which classifies the testing samples in $\mathcal{X}_u$ with candidates from $\mathcal{Y}_u$, while the other is generalized ZSL (GZSL) which extends the testing set to $\mathcal{X}_s \cup \mathcal{X}_u$, with candidates from both seen and unseen classes i.e., $\mathcal{Y}_s \cup \mathcal{Y}_u$.

2.2 Zero-shot Knowledge Graph Completion

Different from the clustered instances in IMGC, a KG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ is composed of a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$ and a set of triple facts $\mathcal{T} = \{(h,r,t)|h,t \in \mathcal{E}; r \in \mathcal{R}\}$. The task of knowledge graph completion (KGC) is proposed to improve Knowledge Graphs by completing the triple facts in KGs when one of h, r, t is missing. Typical KGC methods utilize KG embedding models such as TransE [3] to embed entities and relations in continuous vector spaces (e.g., the embeddings of h, r, t are represented as x_h, x_r, x_t respectively) and conduct vector computations to complete the missing triples, which are trained by existing triples and assume all testing entities and relations are available at training time. Therefore, the zero-shot KGC task is defined as predicting for the newly-added entities or relations which have no associated triples in the training data.

In this study, we focus on those newly-added KG relations (i.e., unseen relations). Specifically, we separate two disjoint relation sets: the seen relation set $\mathcal{R}_s$ and the unseen relation set $\mathcal{R}_u$. The triple set $\mathcal{T}_s = \{(h, r_s, t)|h,t \in \mathcal{E}; r_s \in \mathcal{R}_s\}$ is then collected for training, and $\mathcal{T}_u = \{(h, r_u, t)|h,t \in \mathcal{E}; r_u \in \mathcal{R}_u\}$ is collected to evaluate the prediction of the triples of unseen relations. It is noted that we consider a closed set of entities, i.e., each entity that appears in the testing set already exists in the training set, because making both entity set and relation set open makes the problem much more challenging, and the current work now only considers one of them (see references introduced in Section 3.3). Similar to IMGC, there are also semantic representations of relations in $R_s \cup R_u$, which are learned from textual descriptions or ontological schemas.

With the zero-shot setting, the KGC problem in our study is formulated as predicting the tail entity t given the head entity h and the relation r in a triple. More specifically, for each query tuple (h, r), we assume there is one ground-truth tail entity t such that the triple (h, r, t) is true[2]. The target of KGC model is to assign the highest ranking score to t against the rest of all the candidate entities which are denoted as $C_{(h,r)}$. Therefore, during zero-shot testing, we will predict the triple facts of r_u by ranking t with the candidate tail entities $t' \in C_{(h,r_u)}$. Accordingly, we do not set generalized ZSL (GZSL) testing in this case, considering that the candidate space only involves entities and the prediction with unseen relations is independent of the prediction with seen relations, while the latter is a traditional KGC task which is out of the scope of this paper.

2.3 Ontological Schema

Ontological schemas are used as the semantic prior knowledge for the above ZSL tasks in our paper. The ontology, denoted as $O = \{\mathcal{C}^O, \mathcal{P}^O, \mathcal{T}^O\}$, is a multi-relational graph formed with $\mathcal{C}^O$, a set of concept nodes, $\mathcal{P}^O$, a set of property edges, and $\mathcal{T}^O = \{(c_i, p, c_j)|c_i, c_j \in \mathcal{C}^O, p \in \mathcal{P}^O\}$, a set of RDF triples. The concept nodes here refer to the domain-specific concepts. For example, in IMGC, they are image classes, image attributes, etc. While in KGC, they are KG relations, and their domain (i.e., head entity types) and range (i.e., tail entity types) constraints, etc. As for property edge, it refers to a link between two concepts. The properties in our ontology are a combination of domain-specific properties (e.g., *imgc:hasDecoration*) and RDF/RDFS[3] built-in properties (e.g., *rdfs:subClassOf, rdfs:subPropertyOf*). For example, the triple (*imgc:Zebra, imgc:hasDecoration, imgc:Stripe*) in Figure 1 denotes that animal class "Zebra" is decorated with "Stripe", while the triple

[1] Our code and datasets are available at https://github.com/genggengcss/OntoZSL.

[2] Generally in KGs, there may be more than one correct tail entity for a query tuple. Here, we follow previous KGC work [45] to apply a *filter* setting during testing where other correct tail entities are filtered before ranking and only the current test one left.

[3] https://www.w3.org/TR/rdf-schema/

(*kgc:radiostation_in_city, rdfs:subPropertyOf, kgc:has_office_in_city*) denotes that KG relation "radiostation in city" is a subrelation of "has office in city". In addition to RDF triples with structural relationships between concepts, each concept in the ontological schema also contains a paragraph of textual descriptions. These descriptions are lexically meaningful information of concepts, which can also be represented by triples using properties e.g., *rdfs:comment.*

In our study, we use a semantic embedding technique to encode all the concept nodes in the ontological schema as vectors, through which the class labels in IMGC and the relation labels in KGC are embedded. They are then used as the additional input of GAN models to generate more discriminative samples for unseen image classes or unseen KG relations.

3 RELATED WORK

3.1 Prior Knowledge for ZSL

The first we discuss is the prior knowledge previously used in the ZSL literature. Some works employ attribute descriptions as the priors [1, 10, 24, 25, 27]. In these works, each class is annotated with a series of attributes which describe its characteristics, and the semantic relationships between classes are thus represented by those shared attributes. However, attributes focus on describing "local" characteristics and the semantically identical attributes may perform inconsistently across different classes. For example, in image classification, the animal classes "Horse" and "Pig" share the same attribute "hasTail", but the visual appearance of their tails differs greatly. The model trained with "Horse" may not generalize well on the prediction of "Pig" (i.e., domain shift problem mentioned earlier). Some works prefer to utilize textual descriptions or distributed word embeddings of classes pre-trained on textual data to model the class semantics [11, 31, 34, 51]. Textual data can be easily obtained from linguistic sources such as Wikipedia articles, however, they are noisy and often lead to poor performance.

There are also some works utilizing label ontologies for inter-class relationships, such as label taxonomy [22, 44], Hierarchy and Exclusion (HEX) label graph [8], and label ontology in OWL (Web Ontology Language) [5]. However, these ontologies also have their limitations. The label taxonomy lacks discriminative semantics for those sibling classes which may look quite different (e.g., "Horse" and "Zebra" in Figure 1), while the HEX label graph still focuses on modeling the relationships between any two labels via attributes – one class is regarded as a subclass of the attributes annotated for it, and as exclusive with those that are irrelevant with it. Different from these works, our proposed ontological schema contains more complete semantics, in which the existing priors are well fused and benefit each other. For example, the class-level priors such as label taxonomy provide global constraint for attribute descriptions while class-specific attributes provide more detailed and discriminative priors for classes especially for sibling classes.

Comparison with OWL-based label ontology [5]. Although it expresses the same complete class semantics as we do, the OWL-based semantic representation is difficult to apply due to its complicated definition. While our ontological schema is mainly in the form of multi-relational graphs composed of RDF triples, which is easier to model and embed using many successful triple embedding algorithms. On the other hand, the construction of the ontologies used in [5] heavily relies on the manual work, while our ontological schemas are built upon existing resources or are directly available.

3.2 Zero-shot Learning Strategy

Given the prior knowledge, existing approaches differ significantly in how the features are transferred from seen classes to unseen classes. One branch is based on mapping. Some methods [11, 23, 24, 31] learn an instance-class mapping with seen samples in training. In testing, the features of an input are projected into the vector space of the labels, and the nearest neighbor (a class label) in that space is computed as the output label. Some other methods [4, 22, 44, 51] learn a reverse mapping – labels are mapped to the space of input instances. However, all of these mappings are trained by seen samples, and thus have a strong bias towards seen classes during prediction, especially in generalized ZSL where the output space includes both seen and unseen classes.

Recently, by taking advantages of generative models such as GANs, several methods [13, 21, 25, 34, 47, 53, 54] have been proposed to directly synthesize samples (or features) for unseen classes from their prior knowledge, which convert the ZSL problem to a standard supervised learning problem with the aforementioned bias issue avoided. Although these generative models are trained using the samples of seen classes, the generators can generalize well on unseen classes according to the semantic relationships between them. In this study, we also introduce and evaluate GANs in our framework. As far as we know, our work is among the first to incorporate the ontological schema with GAN for feature generation.

3.3 Zero-shot Knowledge Graph Completion

Reviewing the literature of ZSL, we find that most of works especially those mentioned above are developed in the computer vision community for image classification. There are also several ZSL studies for knowledge graph completion. Some of them devote to deal with the unseen entities by exploiting the auxiliary connections with seen entities [17, 37, 42, 52], introducing their textual descriptions [36], or learning entity-independent relational semantics which summarize the substructure underlying KG so that naturally generalizing to unseen entities [39]. While few works such as [34] focus on the unseen relations. In our work, we also concentrate on these newly-added relations. Different from [34] which generates unseen relation embeddings solely from their textual descriptions, our OntoZSL generates from the ontological schema which describes richer correlations between KG relations, such as domain and range constraints. Besides, OntoZSL is well-suited for zero-shot KGC considering that many KGs inherently have ontologies which highly summarize the entities and relations in KGs.

4 METHODOLOGY

In this section, we will introduce our proposed general ZSL framework **OntoZSL**, which builds upon an ontology embedding technique and a generative adversarial network (GAN) and can be applied to two different zero-shot learning tasks: image classification (IMGC) and knowledge graph completion (KGC). Figure 2 presents its overall architecture, including four core parts:

Ontology Encoder. We learn semantically meaningful class representations or relation representations from the ontological

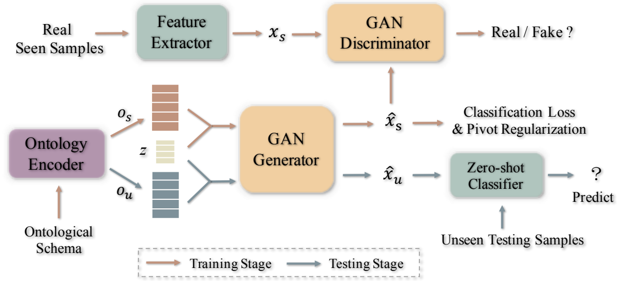


Figure 2: An overview of OntoZSL in the standard ZSL setting. o_s and o_u represent the semantic embeddings of seen and unseen concepts (i.e., image classes in IMGC task and KG relations in KGC task) learned from the ontological schema, respectively. During training, the GAN model generates fake samples $\hat{x}_s$ for seen concepts and distinguishes them with the real samples x_s learned from a feature extractor. With a trained generator, the samples of unseen classes ($\hat{x}_u$) can be generated to learn classifiers for prediction.

schema using an ontology embedding technique, which considers the structural relationships between concepts as well as their correlations implied in textual descriptions.

Feature Extractor. We utilize a feature extractor to extract the real data (instance) representations, which will be taken as the guidance of adversarial training. Regarding the different data forms in two tasks, we adopt different strategies to obtain real data distributions for different tasks.

Generation Model. A typical scheme of GAN is adopted for data generation. It consists of (i) a generator to synthesize instance features from random noises, (ii) a discriminator to distinguish the generated features from real ones, and (iii) some additional losses to ensure the inter-class discrimination of generated features. Notably, we generate instance features instead of raw instances for both higher accuracy and computation efficiency [47].

Zero-shot Classifier. With the well-trained generator, the samples of unseen classes (relations) can be synthesized with their semantic representations, from which the unseen classifiers can be learned to predict the testing samples of unseen classes (relations).

Among these parts, the ontology encoder and the generation model are general across different tasks, while the feature extractor and the zero-shot classifier is task-specific. Next, we will first introduce these parts w.r.t. the IMGC task, and then introduce the difference of the task-specific parts in addressing the KGC task.

4.1 Ontology Encoder

In this subsection, we provide a text-aware semantic embedding technique for encoding the graph structure as well as textual information in the ontological schema.

Default Embedding. Considering the structural RDF triples in ontological schema, there are many triple embedding techniques that can be applied [43]. Given a triple (c_i, p, c_j), the aim of triple embedding is to design a scoring function $f(c_i, p, c_j)$ as the optimization objective. A higher score indicates a more plausible triple. In this paper, we adopt a mature and widely-used triple embedding method TransE [3] which assumes the property in each triple as a translation between two concepts. Its score function is defined as follows:

$$f_{TransE}(c_i, p, c_j) = -||\boldsymbol{c}_i + \boldsymbol{p} - \boldsymbol{c}_j|| \tag{1}$$

where $\boldsymbol{c}_i, \boldsymbol{p}, \boldsymbol{c}_j$ denote the embeddings of c_i, p, c_j, respectively.

To learn the embeddings of all concepts in the ontological schema O, a hinge loss is minimized for all triples in ontology:

$$\mathcal{J}_O = \frac{1}{|\mathcal{T}^O|} \sum_{\substack{(c_i,p,c_j)\in\mathcal{T}^O \\ \wedge(c'_i,p,c'_j)\notin\mathcal{T}^O}} [\gamma_o + f(c'_i, p, c'_j) - f(c_i, p, c_j)]_+ \tag{2}$$

where γ_o is a margin parameter which controls the score difference between positive and negative triples, the negative triples are generated by replacing either head or tail concepts in positive triples with other concepts and not exist in the ontology. Notably, there are other triple embedding techniques can potentially be used for encoding our ontological schema. Since exploring different techniques is not the focus of this paper, we leave them as future work.

Text-Aware Embedding. However, the textual descriptions of concepts in ontological schema describe the knowledge of concepts from another modal. Such semantics require special modeling than regular structural triples. Therefore, we propose the text-aware semantic embedding model by projecting the structural representations and the textual representations into a common space and learning them simultaneously using the same objective score function, as shown in Figure 3.

Specifically, given a triple (c_i, p, c_j), we first project its structural embeddings $\boldsymbol{c}_i, \boldsymbol{p}, \boldsymbol{c}_j$ learned above and the textual representation of concepts $\boldsymbol{d}_i, \boldsymbol{d}_j$ into a common space using fully connected (FC) layers, e.g., $\boldsymbol{c}_i$ into $\boldsymbol{c}_i^s$ and $\boldsymbol{d}_i$ into $\boldsymbol{c}_i^t$ (cf. Figure 3). In this space, the structure-based score is still defined as proposed by TransE:

$$f^s = -||\boldsymbol{c}_i^s + \boldsymbol{p}^s - \boldsymbol{c}_j^s|| \tag{3}$$

while the text-based score is defined as:

$$f^t = -||\boldsymbol{c}_i^t + \boldsymbol{p}^s - \boldsymbol{c}_j^t|| \tag{4}$$

which also constrains the textual representations under the translational assumption.

To make these two types of representations compatible and complementary with each other, we follow the proposals of [30, 48, 49] to define the crossed and additive score function:

$$\begin{aligned} f^{st} &= -||\boldsymbol{c}_i^s + \boldsymbol{p}^s - \boldsymbol{c}_j^t|| \\ f^{ts} &= -||\boldsymbol{c}_i^t + \boldsymbol{p}^s - \boldsymbol{c}_j^s|| \\ f^{add} &= -||(\boldsymbol{c}_i^s + \boldsymbol{c}_i^t) + \boldsymbol{p}^s - (\boldsymbol{c}_j^s + \boldsymbol{c}_j^t)|| \end{aligned} \tag{5}$$

All of these score functions ensure that the two kinds of concept representations are learned in the same vector space. The overall score function are defined as:

$$f^T(c_i, p, c_j) = f^s + f^t + f^{st} + f^{ts} + f^{add} \tag{6}$$

Therefore, the final training loss changes to:

$$\mathcal{J}_O^{Text} = \frac{1}{|\mathcal{T}^*|} \sum_{\substack{(c_i,p,c_j)\in\mathcal{T}^* \\ \wedge(c'_i,p,c'_j)\notin\mathcal{T}^*}} [\gamma_o + f^T(c'_i, p, c'_j) - f^T(c_i, p, c_j)]_+ \tag{7}$$

where $\mathcal{T}^*$ refers to the triple set with regular structural properties. The triples with *rdfs:comment* property that connects a concept

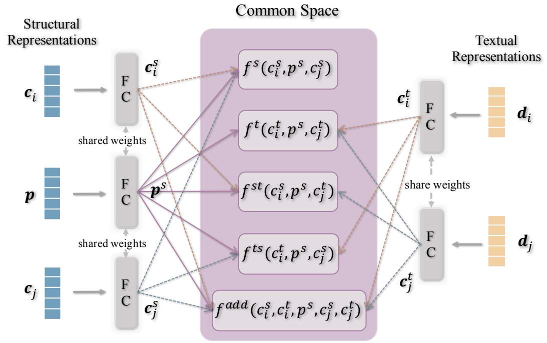


Figure 3: Overview of the network architecture for text-aware semantic embedding.

with its textual description are removed here considering that these text nodes are encoded from another view.

After training, for each concept i in ontological schema, we can learn two types of concept embeddings, i.e., structure based $\boldsymbol{c}_i^s$ and text based $\boldsymbol{c}_i^t$. To fuse the semantic features from these two types, we concatenate them to form the final concept embedding:

$$o_i = [\boldsymbol{c}_i^s; \boldsymbol{c}_i^t] \tag{8}$$

As for the initial textual representations, we use word embeddings pre-trained on textual corpora to represent the words in the text. Also, to suppress the text noises, we employ TF-IDF features [35] to evaluate the importance of each word. Besides, the FC layers used for projection also have a promotion for noise suppression.

4.2 Feature Extractor

Following most of the previous works [46], we employ ResNet101 [19] to extract the real features of images in zero-shot image classification. ResNet is a well-performed CNN architecture pre-trained on ImageNet [9] with 1K classes, in which no unseen classes of our evaluation datasets are involved.

4.3 Feature Generation with GAN

With class embeddings learned from ontology encoder, we train a generator G, which takes a class embedding o_i and a random noise vector z sampled from Normal distribution $\mathcal{N}(0, 1)$ as input, and generates the CNN features $\hat{x}$ of class i. The loss of G is defined as:

$$\mathcal{L}_G = -\mathbb{E}[D(\hat{x})] + \lambda_1 \mathcal{L}_{cls}(\hat{x}) + \lambda_2 \mathcal{L}_P \tag{9}$$

where $\hat{x} = G(z, o_i)$. The first term of loss function is the Wasserstein loss [2] which can effectively eliminate the mode collapse problem during generation. While the second term is a supervised classification loss for classifying the synthesized features, and the third item is a pivot regularization proposed by [53], which regularizes the mean of generated features of each class to be the mean of real feature distribution. Both of the latter two loss terms encourage the generated features to have more inter-class discrimination. λ_1 and λ_2 are the corresponding weight coefficients.

The discriminator D then takes the synthesized features $\hat{x}$ and the real features x extracted from a training image of class i as input, the loss can be formulated as:

$$\begin{aligned} \mathcal{L}_D = & \ \mathbb{E}[D(x, o_i)] - \mathbb{E}[D(\hat{x})] \\ & -\beta \mathbb{E}[(|| \nabla_{\tilde{x}} D(\tilde{x}) ||_p - 1)^2] \end{aligned} \tag{10}$$

where the first two terms approximate the Wasserstein distance of the distribution of real features and synthesized features, and the last term is the gradient penalty to enforce the gradient of D to have unit norm (i.e., Lipschitz constraint proposed by [15]), in which $\tilde{x} = \varepsilon x + (1 - \varepsilon)\hat{x}$ with $\varepsilon \sim U(0, 1)$, and β is the weight coefficient.

The GAN is optimized by a minimax game, which minimizes the loss of G but maximizes the loss of D. We also note that the generator and discriminator are both incorporated with the class embeddings during training. This is a typical method of conditional GANs [29] that introduces external information to guide the training of GANs, which is consistent with the generative ZSL – synthesizing instance features based on the prior knowledge of classes.

4.4 Zero-shot Classifiers

Once the GAN is trained to be able to generate sample features for seen classes, it can also synthesize features for unseen classes with random noises and their corresponding class embeddings. Consequently, with synthesized unseen data $\hat{\mathcal{X}}_u$, we can learn a typical softmax classifier for each unseen class and classify its testing samples. The classifier is optimized by:

$$\min_{\theta} -\frac{1}{|\mathcal{X}|} \sum_{(x,y) \in (\mathcal{X}, \mathcal{Y})} logP(y|x; \theta) \tag{11}$$

where $\mathcal{X}$ represents the features for training, $\mathcal{Y}$ is the label set to be predicted on, θ is the training parameter and $P(y|x;\theta) = \frac{exp(\theta_y^T x)}{\sum_i^{|\mathcal{Y}|} exp(\theta_i^T x)}$. Regarding the different prediction setting in IMGC, $\mathcal{X} = \hat{\mathcal{X}}_u$ when it is standard ZSL and $\mathcal{X} = \mathcal{X}_s \cup \hat{\mathcal{X}}_u$ when it is GZSL, while the label set $\mathcal{Y}$ corresponds to $\mathcal{Y}_u$ and $\mathcal{Y}_s \cup \mathcal{Y}_u$ respectively.

4.5 Adapting to Knowledge Graph Completion

Similar to zero-shot image classification, we can also generate features for unseen relations in knowledge graph with their semantic representations learned from ontological schema using the above generation model. However, considering the different data instances in KGs, we adopt different strategies to extract sample features and design different zero-shot classifiers.

Feature Extractor. Unlike traditional KG embedding methods which learn entity and relation embeddings based on some assumptions (or constraints), we hope to learn cluster-structure feature distribution for seen and unseen relation facts so that preserving the higher intra-class similarity and relatively lower inter-class similarity as most ZSL works did. Therefore, we follow [34] to learn and train the real relation embeddings in bags. To be more specific, suppose that there are bags $\{B_r | r \in \mathcal{R}_s\}$ in the training set, a bag B_r is named by a seen relation r, in which all the triples involving relation r are contained. The real embeddings x_r of relation r are thus represented by the embeddings of entity pairs in bag B_r, whose training are thus supervised by some reference triples in this bag.

Concretely, for an entity pair (h, t) in bag B_r, we first embed each entity using a simple fully connected (FC) layer and generate the

Table 1: Statistics of the zero-shot image classification datasets.

Datasets	Granularity	# Classes			# Images					Ontological Schema		
						for Training		for Testing				
		Total	Seen	Unseen	**Total**	Seen	Unseen	Seen	Unseen	# RDF Triples	# Concepts	# Properties
AwA	coarse	50	40	10	37,322	23,527	0	5,882	7,913	1,256	180	12
ImNet-A	fine	80	28	52	77,323	36,400	0	1,400	39,523	563	227	19
ImNet-O	fine	35	10	25	39,361	12, 907	0	500	25,954	222	115	8

entity pair embedding u_{ep} as:

$$\begin{aligned} u_{ep} &= \sigma([f_1(x_h); f_1(x_t)]) \\ f_1(x_h) &= W_1(x_h) + b_1 \\ f_1(x_t) &= W_1(x_t) + b_1 \end{aligned} \tag{12}$$

where $[\cdot;\cdot]$ represents the vector concatenation operation, σ is the *tanh* activation function. We also consider the one-hop structure of each entity. For the tail entity t, its structural embedding u_t is represented as:

$$\begin{aligned} u_t &= \sigma\Big(\frac{1}{|\mathcal{N}_t|} \sum_{(r^n,t^n)\in\mathcal{N}_t} f_2(x_{r^n}, x_{t^n})\Big), \\ f_2(x_{r^n}, x_{t^n}) &= W_2([x_{r^n}; x_{t^n}]) + b_2 \end{aligned} \tag{13}$$

where $\mathcal{N}_t = \{(r^n, t^n) | (t, r^n, t^n) \in \mathcal{T}_s\}$ denotes the one-hop neighbors of entity t, and f_2 is a FC layer which encodes the neighborhood information. In consideration of the scalability, the number of neighbors (i.e., $|\mathcal{N}_t|$) is set with an upper limit e.g., 50. The structural embedding of the head entity h, denoted as u_h is calculated in the same way as the tail entity. The final entity pair embedding (i.e., the relation embedding x_r) is then formulated as:

$$x_r = x_{(h,t)} = [u_{ep}; u_h; u_t] \tag{14}$$

We train the real relation embeddings with some reference triples. Specifically, for each relation r, the triples in bag B_r are randomly split into two parts: one is taken as the reference set $\{h^*, r, t^*\}$, and the other is taken as the positive set $\{h^+, r, t^+\}$. We also generate a set of negative triples $\{h^+, r, t^-\}$ by replacing the tail entity of each triple in the positive set with other entities. With m reference triples, we take the mean of reference relation embeddings, i.e., $x^c_{(h^*,t^*)} = \frac{1}{m}\sum_{i=1}^m x^i_{(h^*,t^*)}$, where $x^i_{(h^*,t^*)}$ is computed by equations (12), (13) and (14), and calculate its cosine similarity with the relation embedding of each positive triple (i.e., $x_{(h^+,t^+)}$) as a positive score denoted as $score^+$, and calculate its cosine similarity with that of each negative triple (i.e., $x_{(h^+,t^-)}$) as a negative score denoted as $score^-$. A hinge loss is then adopted to optimize the training:

$$\mathcal{J}_{FE} = \frac{1}{|B^*_r|} \sum_{\substack{(h^+,r,t^+)\in B^*_r \\ \wedge (h^+,r,t^-)\notin B^*_r}} [\gamma_f + score^+ - score^-]_+ \tag{15}$$

where B^*_r means the training triples of relation r except reference triples, and γ_f denotes the margin parameter. Instead of random initialization, we use pre-trained KG embedding to initialize the entities and relations in bags and neighborhoods.

During feature generation, when generating the fake relation embedding $\hat{x}_r = G(z, o_r)$ for relation r, we also take the above hinge loss as the classification loss to preserve the inter-class discrimination. Specifically, a positive score is calculated between $\hat{x}_r$ and the cluster center of real relation embeddings, i.e., $x^c_r = \frac{1}{|N_r|}\sum_{i=1}^{N_r} x^i_r$, where N_r denotes the number of training triples of relation r. A negative score is computed between x^c_r and the negative relation embedding calculated by negative triples with replaced tail entities.

Zero-shot Classifiers. With the well-trained generator, we can generate plausible relation embedding $\hat{x}_{r_u} = G(z, o_{r_u})$ for unseen relation r_u with its semantic representations o_{r_u}. For a query tuple (h, r_u), the similarity ranking value $v_{(h,r_u,t')}$ of candidate tail t' is calculated by the cosine similarity between $\hat{x}_{r_u}$ and $x_{(h,t')}$. The candidate with the highest value is the predicted tail entity of tuple (h, r_u). For better generalization, we generate multiple relation embeddings for each relation and average the ranking value:

$$v_{(h,r_u,t')} = \frac{1}{|N_g|} \sum_{i=1}^{N_g} cosine(\hat{x}^i_{r_u}, x_{(h,t')}) \tag{16}$$

where N_g denotes the number of generated relation embeddings for relation r_u.

5 EXPERIMENTS

In the experiments, we evaluate OntoZSL by the two different tasks of zero-shot image classification and zero-shot knowledge graph completion. We also compare the ontological schema against other prior knowledge for zero-shot learning, and finally analyze the impact of different semantics of the ontological schema.

5.1 Image Classification

Datasets. We evaluate the zero-shot image classification task with a standard benchmark named Animals with Attributes (AwA) and two benchmarks ImNet-A and ImNet-O which are contributed by ourselves. AwA [46] is a widely used coarse-grained benchmark for animal classification which contains 50 animal classes with 37, 322 images, while ImNet-A and ImNet-O are two fine-grained datasets we extract from ImageNet [9]. ImNet-A is for the classification of animals, while ImNet-O is for the classification of more general objects. Details of the construction of the latter two benchmarks can be found in Appendix A. The classes of all the three benchmarks are split into seen and unseen classes, following the *seen-unseen* strategy proposed in [46].

Table 1 provides detailed statistics of these datasets. Compared with AwA, ImNet-A and ImNet-O are more challenging as they have fewer seen classes. Both of these datasets have inherent label taxonomies which are helpful for building the ontological schemas. Specifically, each class corresponds to a node in WordNet [28] – a lexical database of semantic relations between words, and thus these classes are underpinned by the same taxonomy as WordNet.

The Ontological Schema for IMGC mainly focuses on the class hierarchy, class attributes and literal descriptions. To build

Table 2: The *acc*uracy (%) of image classification in the standard and generalized ZSL settings. The best results are marked in bold. "–" means the case where the method cannot be applied.

	Standard ZSL			Generalized ZSL								
Methods	**AwA**	**ImNet-A**	**ImNet-O**	**AwA**			**ImNet-A**			**ImNet-O**		
	acc	acc	acc	acc_s	acc_u	H	acc_s	acc_u	H	acc_s	acc_u	H
DeViSE	37.46	14.30	14.32	81.06	3.29	6.32	60.21	0.64	1.27	68.00	3.68	6.98
CONSE	22.99	20.28	12.41	51.64	3.28	6.17	86.40	0.00	0.00	62.00	0.00	0.00
SAE	42.28	18.98	14.84	71.03	9.79	17.21	84.43	0.17	0.34	**92.60**	0.16	0.32
SYNC	39.14	20.52	18.58	**88.21**	8.43	15.39	**88.72**	0.00	0.00	62.53	0.00	0.00
GCNZ	–	32.47	30.05	–	–	–	47.79	15.15	23.01	44.60	14.48	21.87
DGP	58.99	34.88	31.23	86.19	16.59	27.82	50.14	17.87	26.35	47.40	19.00	27.13
GAZSL	56.29	21.20	19.40	87.64	15.40	26.19	86.56	1.28	2.52	86.80	6.16	11.50
LisGAN	58.89	21.90	20.20	60.03	44.30	50.98	35.50	15.55	21.62	35.00	13.87	19.87
LsrGAN	56.34	19.69	20.20	85.98	35.73	50.48	36.29	13.49	19.67	37.80	14.27	20.72
OntoZSL	**63.31**	**39.00**	**34.24**	64.90	**49.35**	**56.06**	37.86	**27.94**	**32.15**	43.40	**21.50**	**28.76**

such an ontological schema, we first adopt the taxonomy of WordNet to define the class hierarchy, where the class concepts are connected via the property *rdfs:subClassOf*, as Figure 1 shows. Then, we define the domain-specific properties such as *imgc:hasDecoration*, *imgc:coloredIn* to associate the class concepts with attribute concepts so that describing the visual characteristics of classes in ontology. The attributes of classes are usually hand-labeled, in our paper, we reuse existing attribute annotations for AwA [24] and manually annotate attributes for classes in ImNet-A and ImNet-O since they have no open attributes. During annotating, we also transfer some annotations from other datasets (e.g., AwA) to reduce the annotation cost (more details are in Appendix A). As for the literal descriptions of concepts, we adopt the words of class names, which are widely-used text-based class semantics in the literature. The statistics of constructed ontological schemas are shown in Table 1.

Baselines and Metrics. We compare our framework with classic ZSL methods published in the past few years and the state-of-the-art ones reported very recently. Specifically, **DeViSE** [11], **CONSE** [31], **SAE** [23] and **SYNC** [4] are mapping-based which map the image features into the label space represented by class embeddings or vice versa; while **GAZSL** [53], **LisGAN** [25] and **LsrGAN** [40] are generative methods which generate visual features conditioned on the class embeddings. We evaluate these methods with their available class embeddings. For AwA, the binary attribute vectors are adopted, while for ImNet-A and ImNet-O, as the attributes of ImageNet classes are not available, we use the word embeddings of class names provided by [4]. We also make a comparison with **GCNZ** [44] and **DGP** [22] which leverage the word embeddings of class labels and label taxonomy to predict on AwA and ImageNet.

We evaluate these methods by accuracy. Considering the imbalanced samples across classes, we follow the current literature [46] to report the class-averaged (macro) accuracy. Specifically, we first calculate the per-class accuracy – the ratio of correct predictions over all the testing samples of this class, and then average the accuracies of all targeted classes as the final metric. Regarding the two testing settings in zero-shot image classification, the metrics are computed on all unseen classes in the standard setting, while in the GZSL setting, the class-averaged accuracy is calculated on seen and unseen classes separately, denoted as acc_s and acc_u respectively, and then a harmonic mean $H = (2 \times acc_s \times acc_u)/(acc_s + acc_u)$ is computed as the overall metric.

Implementation. We employ ResNet101 [19] to extract 2,048-dimensional visual features of images. As for the word embeddings used for initializing the textual representation of ontology concepts, we use released 300-dimensional word vectors, which are pre-trained on Wikipedia corpora using GloVe [32] model.

The results are reported based on the following settings. For ontology encoder, we set $\gamma_o = 12$ as default, and learn the class embedding of dimension 200 (i.e., 100-dimensional structure-based representation and 100-dimensional text-based representation). We set other parameters in ontology encoder as recommended by [38] and [30]. Regarding GAN, the generator and discriminator both consist of two fully connected layers. The generator has 4,096 hidden units and outputs image features with 2,048 dimensions, while the discriminator also has 4,096 hidden units and outputs a 2-dimensional vector to indicate whether the input feature is real or not. The dimension of noise vector z is set to 100. The learning rate is set to 0.0001. The weight λ_1 for classification loss is set to 0.01, λ_2 for pivot regularization is set to 5, and the weight β for gradient penalty is set to 10.

Results. We report the prediction results under the standard ZSL setting and the generalized ZSL setting in Table 2. Giving a first look at the standard ZSL, we find that our method achieves the best accuracy on all three datasets. In comparison with the mapping-based baselines, e.g., DeVise, CONSE and the generative baselines, e.g., GAZSL, LsrGAN, our method outperforms the traditional class attribute annotations used for AwA as well as class word embeddings for ImNet-A and ImNet-O. Most importantly, it also has a better performance than the previously proposed label ontologies (i.e., GCNZ and DGP), which simply consider the hierarchical relationships of classes. These observations demonstrate the superiority of our OntoZSL compared with the state of the art.

While in the GZSL setting, we have similar observations as the standard one. Our method performs better than the baselines and obtains significant outperformance on the metrics of acc_u and H. This shows our method has a better generalization. Furthermore, we notice that among all the methods which utilize the same prior knowledge (i.e., word embeddings of classes or class attribute vectors), the performance of those mapping-based ones dramatically drops in comparison with the standard ZSL setting. CONSE and SYNC even drop to 0.00 on ImNet-A and ImNet-O. This verifies our points that these methods have a bias towards seen classes during prediction, i.e., their models tend to predict the unseen testing samples on seen classes. In contrast, those generative methods which generate training samples for unseen classes have no such bias towards unseen classes. We also find that although our framework does not achieve the best results on the prediction of seen testing samples (acc_s), it still accomplishes competitive performance as the state-of-the-arts. This motivates us to explore algorithms to predict unseen testing samples correctly as well as maintain reasonable accuracy on seen classes.

5.2 Knowledge Graph Completion

Datasets. We evaluate the zero-shot knowledge graph completion task on two benchmarks proposed by [34], i.e., NELL-ZS and Wikidata-ZS extracted from NELL[4] and Wikidata[5] respectively, two knowledge graphs are also known for constructing few-shot KGC datasets [7]. The dataset statistics are listed in Table 3.

Ontological Schema for KGC. Different from the personally defined ontological schemas for IMGC task, many KGs inherently have ontologies which abstractly summarize the entities and relations in knowledge graphs. Therefore, we access public ontologies of KGs and make a reorganization to construct the ontological schemas we need. Specifically, for NELL, we process the original ontology file[6] and filter out four kinds of properties to describe the high-level knowledge about NELL relations, i.e., the *kgc:domain* and *kgc:range* properties which constrain the types of the head entity and the tail entity of a specific relation, respectively, the *kgc:generalizations* property which describes the hierarchical structure of relations and entity types, and the *kgc:description* property which introduces the literal descriptions of relations and entity types. While for Wikidata, we utilize Wikidata toolkit packaged in Python[7] to access the knowledge of Wikidata relations, in which the *kgc:P2302* is used to describe the domain and range constraints of relations, and *rdfs:subPropertyOf* and *rdfs:subClassOf* are used to describe the hierarchical structure of relation and entity types. Apart from the textual descriptions of relation and entity types, we also leverage the properties *kgc:P31*, *kgc:P1629* and *kgc:P1855* as the additional knowledge. The statistics of the processed ontological schemas are shown in Table3. It is noted that we can also take the original ontologies of these two KGs, but some ontology simplification techniques such as [41] may be needed to forget the irrelevant concepts or properties for prediction tasks contained in the ontologies. We will consider to develop them in the future.

Baselines and Metrics. We mainly compare our proposed OntoZSL with the ZSGAN proposed in [34], which generates embeddings for unseen KG relations from their textual descriptions and entity type descriptions. In ZSGAN and our OntoZSL, the feature extractor can be flexibly incorporated with different pre-trained KG embeddings. In view of generalization, we adopt two representative KG embedding models TransE [3] and DistMult [50] in the feature extractor in our experiments. We also compare the original TransE and DistMult in the zero-shot learning setting, i.e., ZS-TransE and ZS-DistMult, where the randomly initialized relation embeddings are replaced by their textual embeddings which are trained together with entity embeddings by the original score functions.

Table 3: Statistics of the zero-shot knowledge graph completion datasets. # Ent. and # Triples denote the number of entities and triples in KGs. # Rel. (Tr/V/Te) denotes the number of KG relations for training/validation/testing. # Onto. (Trip./Con./Pro.) denotes the number of the RDF triples/concepts/properties in the ontological schemas.

Datasets	# Ent. & Triples	# Rel. Tr/V/Te	# Onto. Trip./Con./Pro.
NELL-ZS	65,567 / 188,392	139/10/32	3,055/1,186/4
Wikidata-ZS	605,812 / 724,967	469/20/48	10,399/3,491/8

As mentioned in Section 2.2, the KGC is to complete (rank) the tail entity given the head entity and relation in a triple. Therefore, we adopt two metrics commonly used in the KGC literature [43]: mean reciprocal ranking (*MRR*) and $Hit@k$ to evaluate the prediction results of all testing triples. *MRR* represents the average of the reciprocal predicted ranks of all correct entities; while $Hit@k$ denotes the percentage of testing samples whose correct entities are ranked in the top-k positions. The k is often set to $1, 5, 10$.

Implementation. We pre-train the feature extractor to extract 200-dimensional relation embedding for NELL-ZS and extract 100-dimensional relation embedding for Wikidata-ZS. In pre-training, the margin parameter γ_f is set to 10, we also follow [34] to split 30 triples as references, and set the learning rate to 0.0005.

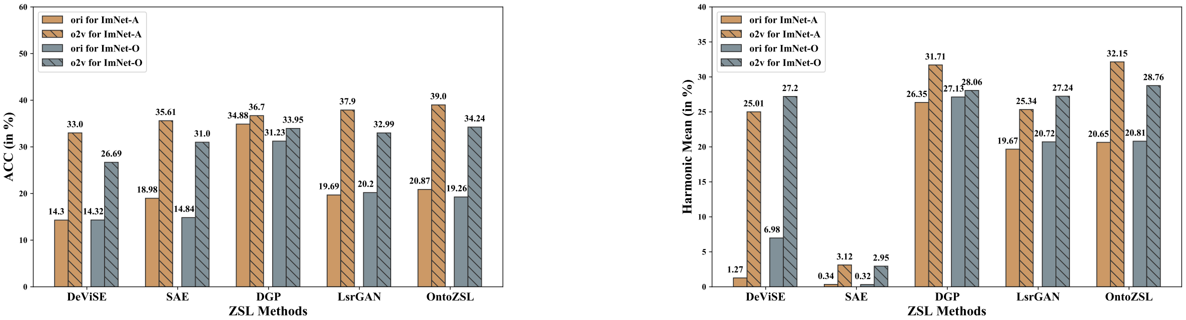
We adopt the same ontology encoder configurations and parameters as IMGC to learn 600-dimensional class embeddings for KGC task. Especially, the TF-IDF features [35] are used to evaluate the importance of words in textual descriptions. Also, we adopt the same GAN architecture as IMGC. But the difference is, for NELL-ZS, the generator has 400 hidden units and outputs 200-dimensional relation embeddings, while the discriminator has 200 hidden units and outputs a 2-dimensional vector to indicate whether the input embedding is real or not. While for Wikidata-ZS, the hidden units of the generator is 200 and that of the discriminator is 100. The dimension of noise vector z is set to 15 for both datasets, and the number of generated relation embeddings N_g is 20. The weight λ_1 for classification loss is set to 1, λ_2 for pivot regularization is set to 3. Other parameters for training GAN are identical to those in IMGC task.

Results. Considering that the dataset splits proposed in [34] are rather new in this domain and the authors do not provide any explanations for the splits, for a fairer comparison, we conduct experiments with originally proposed train/validation splits as well as with random splits for 3-fold cross-validation. Notably, the testing relations are fixed, only the training and validation set are redistributed (i.e., 139 training relations and 10 validation relations for NELL-ZS, 469 training and 20 validation relations for Wikidata-ZS). We evaluate our method and ZSGAN on these 4 splits for both datasets and report the average results in Table 4. The results of ZS-TransE and ZS-DistMult from original paper, referred as ZS-TransE (Paper) and ZS-DistMult (Paper), are included in the comparison.

[4] http://rtw.ml.cmu.edu/rtw/
[5] https://www.wikidata.org/
[6] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1115.ontology.csv.gz
[7] https://pypi.org/project/Wikidata/

Table 4: Results (%) of zero-shot knowledge graph completion with unseen relations. The underlined results are the best in the whole column, while the bold results are the best in the pre-training group.

Pre-trained KG Embedding	Methods	NELL-ZS				Wikidata-ZS			
		MRR	*Hit*@10	*Hit*@5	*Hit*@1	*MRR*	*Hit*@10	*Hit*@5	*Hit*@1
TransE	ZS-TransE (Paper)	9.7	20.3	14.7	4.3	5.3	11.9	8.1	1.8
	ZSGAN	23.4	37.3	30.4	16.0	17.7	25.8	20.7	13.1
	OntoZSL	**25.0**	**39.9**	**32.7**	**17.2**	**18.4**	**26.5**	**21.5**	**13.8**
DistMult	ZS-DistMult (Paper)	23.5	32.6	28.4	18.5	18.9	23.6	21.0	16.1
	ZSGAN	24.9	37.6	30.6	18.3	20.7	28.4	23.5	16.4
	OntoZSL	**25.6**	**38.5**	**31.8**	**18.8**	**21.1**	**28.9**	**23.8**	**16.7**



(a) Standard ZSL

(b) Generalized ZSL

Figure 4: Performance of using different priors w.r.t. different ZSL methods on ImNet-A and ImNet-O. "ori" denotes the original class embedding by word2vec or class hierarchy; "o2v" denotes the ontology-based class embedding. ACC (resp. Harmonic Mean) is reported for the standard (resp. generalized) ZSL setting.

In Table 4, we categorize the results into two groups, based on the different pre-trained KG embeddings. In each group, our OntoZSL achieves consistent improvements over baselines on both datasets. It indicates that the prior knowledge of KG relations that exists in the ontological schema is superior to that in the textual descriptions. It is also observed that a higher improvement is achieved when the score function used for the ontology encoder is consistent with that used for pre-training KG embeddings. For example, compared with ZSGAN on NELL-ZS, the performance is improved by 2.6% on *Hit*@10 with TransE-based pre-trained KG embedding (see the second and third row of Table 4), while is only improved by 0.9% on *Hit*@10 with DistMult-based KG embedding (see the fifth and sixth row of Table 4).

5.3 Impact of Ontological schema

To further validate the effectiveness of our ontology-based class semantics, we compare the capabilities of different prior knowledge by applying different class embeddings to multiple ZSL methods including some representative baselines as well as ours. Taking the experiments on image classification datasets ImNet-A and ImNet-O as examples, the originally used word embeddings of classes and the ontology-based class embeddings are applied to the baselines including DeViSE, SAE, DGP and LsrGAN and our method, respectively. For DeViSE, SAE and LsrGAN, the original class embeddings can be directly replaced with the ontology-based class embeddings we learned, while for DGP which involves word embeddings of classes and class hierarchy, we add attribute nodes produced in our ontological schema into the hierarchical graph to predict the unseen classifiers.

As reported in Figure 4, the ontology-based class embedding achieves higher performance for all the methods. For those methods that use class word embeddings as priors, the ontology-based class embeddings have a more than 12% increment on two datasets under the standard ZSL setting, and a more than 2.5% increment under the GZSL setting. In particular, the harmonic mean of DeViSE increases from 1.27% to 25.01% on ImNet-A and from 6.98% to 27.20% on ImNet-O. On the other hand, our method expectedly has worse performance after using word embeddings of classes as priors. Our ontology-based class semantics also improve the performance of DGP when we add attributes to its original class semantics. For example, on ImNet-A, its performance is improved by 1.82% in the standard ZSL setting and by 5.36% in the GZSL setting. To sum up, our ontology-based class embedding which includes richer class semantics actually performs better than those traditional priors and is beneficial to kinds of ZSL methods.

5.4 Impact of Ontology Components

In this subsection, we evaluate the contribution of different components of the ontological schemas by analyzing the performance drop when one of them is removed. Specifically, with crippled ontological schema, we retrain the ontology encoder to generate class embedding, and then take it as the input of generation model to synthesize unseen features. We conduct experiments on both two tasks.

Table 5: Results of OntoZSL on ImNet-A when textual descriptions ("-text"), class hierarchy ("-hie") or class attributes ("-att") are removed from the ontological schema.

	Standard ZSL	Generalized ZSL		
	acc	acc_s	acc_u	H
all	39.00	37.86	27.94	32.15
- text	37.63	35.07	28.50	31.45
- hie	35.50	39.29	24.35	30.07
- att	33.88	38.07	23.71	29.22

Table 6: Results of OntoZSL on NELL-ZS when textual descriptions ("-text"), relation and entity type hierarchy ("-hierarchy") or relation constrains ("-domain&range") are removed from the ontological schema.

	MRR	$Hit@10$	$Hit@5$	$Hit@1$
all	0.250	39.9	32.7	17.2
- text	0.247	39.7	32.5	16.8
- hierarchy	0.221	35.8	29.5	14.7
- domain & range	0.243	38.0	31.6	16.7

For IMGC, we respectively consider removing the literal descriptions, class hierarchy and class attributes in the ontological schema, while for KGC, we consider removing relation constraints (i.e., domain and range constraints), relation and entity type hierarchy, and literal descriptions. The results on ImNet-A and NELL-ZS are shown in Table 5 and Table 6, respectively. The results on NELL-ZS are based on the KG pre-training setting of TransE.

From Table 5, we can see that the performance of zero-shot image classification is significantly declined when the class attributes are removed under both standard and generalized ZSL settings. This may be due to the following facts. First, the attributes describe quite discriminative visual characteristics of classes. Second, the classes in ImNet-A are fine-grained. They contain some sibling classes whose differences by the taxonomy are not discriminative. One example is *Horse* and *Zebra* in Figure 1. It is hard to distinguish the testing images of such classes when there is a lack of attribute knowledge. As for the KGC task, as shown in Table 6, we find that the hierarchy of relation and entity type has a great influence on the performance. It is probably because around 58% of the relations in the NELL-ZS dataset are hierarchically related, while only nearly 30% of them have identical domain and range constraints.

We also find that the performance on both two tasks are slightly influenced when the literal descriptions are removed, indicating the class semantics that exist in text are weak or redundant compared with other semantics. However, when all of these semantics combined, the best results are achieved. This means these different components of the ontological schema all have a positive contribution to the ZSL model and are complementary to each other.

6 CONCLUSION AND OUTLOOK

In this paper, we propose to use an ontological schema to model the prior knowledge for ZSL. It not only effectively fuses the existing priors such as class hierarchy, class attributes and textual descriptions for image classes, but also additionally introduces more comprehensive prior information such as relation value constraints for KG relations. Accordingly, we develop a new ZSL framework OntoZSL, in which a well-suited semantic embedding technique is used for ontology encoder and a Generative Adversarial Network is adopted for feature generation. It achieves higher performance than the state-of-the-art baselines on various datasets across different tasks. The proposed ontological schemas are shown to be more effective than the traditional prior knowledge.

In this work, we mainly focus on the newly-added KG relations for the KGC task. In the future, we would extend our OntoZSL to learn embeddings for those newly-added entities. Furthermore, we also plan to further extend OntoZSL to explore other tasks such as those related to natural language processing.

ACKNOWLEDGMENTS

This work is funded by 2018YFB1402800/NSFCU19B2027/NSFC91846204. Jiaoyan Chen is mainly supported by the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project 237889) and Samsung Research UK.

REFERENCES

[1] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. 2013. Label-Embedding for Attribute-Based Classification. In *CVPR*. 819–826.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized Classifiers for Zero-Shot Learning. In *CVPR*. 5327–5336.

[5] Jiaoyan Chen, Freddy Lecue, Yuxia Geng, Jeff Z Pan, and Huajun Chen. 2020. Ontology-guided Semantic Composition for Zero-Shot Learning. In *KR2020*.

[6] Muhao Chen, Yingtao Tian, Xuelu Chen, Zijun Xue, and Carlo Zaniolo. 2018. On2vec: Embedding-based relation prediction for ontology population. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 315–323.

[7] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. In *EMNLP/IJCNLP (1)*. 4216–4225.

[8] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. 2014. Large-Scale Object Classification Using Label Relation Graphs. In *ECCV (1)*, Vol. 8689. 48–64.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[10] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. 2009. Describing objects by their attributes. In *CVPR*. IEEE Computer Society, 1778–1785.

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*. 2121–2129.

[12] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, and Shaogang Gong. 2015. Transductive Multi-View Zero-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 11 (2015), 2332–2345.

[13] Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2020. Generative Adversarial Zero-shot Learning via Knowledge Graphs. *CoRR* (2020).

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[15] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *NIPS*. 5767–5777.

[16] Víctor Gutiérrez-Basulto and Steven Schockaert. 2018. From knowledge graph embedding to ontology embedding? An analysis of the compatibility between vector space representations and rules. *arXiv preprint arXiv:1805.10461* (2018).

[17] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge Transfer for Out-of-Knowledge-Base Entities : A Graph Neural Network Approach. In *IJCAI*. ijcai.org, 1802–1808.

[18] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. In *KDD*. 1709–1719.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[20] Ian Horrocks. 2008. Ontologies and the semantic web. *Commun. ACM* 51, 12 (2008), 58–67.

[21] He Huang, Changhu Wang, Philip S. Yu, and Chang-Dong Wang. 2019. Generative Dual Adversarial Network for Generalized Zero-Shot Learning. In *CVPR*. 801–810.

[22] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. 2019. Rethinking Knowledge Graph Propagation for Zero-Shot Learning. In *CVPR*. Computer Vision Foundation / IEEE, 11487–11496.

[23] Elyor Kodirov, Tao Xiang, and Shaogang Gong. 2017. Semantic Autoencoder for Zero-Shot Learning. In *CVPR*. IEEE Computer Society, 4447–4456.

[24] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence* 36, 3 (2013), 453–465.

[25] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. 2019. Leveraging the Invariant Side of Generative Zero-Shot Learning. In *CVPR*. 7402–7411.

[26] Juan Li, Ruoxu Wang, Ningyu Zhang, Wen Zhang, Fan Yang, and Huajun Chen. 2020. Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification. In *COLING*. 2967–2978.

[27] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2020. Attribute Propagation Network for Graph Zero-Shot Learning.. In *AAAI*. 4868–4875.

[28] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[29] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[30] Hatem Mousselly-Sergieh, Teresa Botschen, Iryna Gurevych, and Stefan Roth. 2018. A multimodal translation-based approach for knowledge graph representation learning. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. 225–234.

[31] Mohammad Norouzi, Tomás Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-Shot Learning by Convex Combination of Semantic Embeddings. (2014).

[32] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[33] Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. 2016. Less is More: Zero-Shot Learning from Online Textual Documents with Noise Suppression. In *CVPR*. IEEE Computer Society, 2249–2257.

[34] Pengda Qin, Xin Wang, Wenhu Chen, Chunyun Zhang, Weiran Xu, and William Yang Wang. 2020. Generative Adversarial Zero-Shot Relational Learning for Knowledge Graphs. In *AAAI*. AAAI Press, 8673–8680.

[35] Gerard Salton and Chris Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manag.* 24, 5 (1988), 513–523.

[36] Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. 2019. An open-world extension to knowledge graph completion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3044–3051.

[37] Baoxu Shi and Tim Weninger. 2018. Open-World Knowledge Graph Completion. In *AAAI*. AAAI Press, 1957–1964.

[38] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR (Poster)*. OpenReview.net.

[39] Komal K Teru, Etienne Denis, and William L Hamilton. 2019. Inductive Relation Prediction by Subgraph Reasoning. *arXiv* (2019), arXiv–1911.

[40] Maunil R Vyas, Hemanth Venkateswara, and Sethuraman Panchanathan. 2020. Leveraging seen and unseen semantic relationships for generative zero-shot learning. In *European Conference on Computer Vision*. Springer, 70–86.

[41] Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, and Grigoris Antoniou. [n.d.]. Eliminating Concepts and Roles from Ontologies in Expressive Descriptive Logics. *Computational Intelligence* 30(2) ([n. d.]), 205–232.

[42] Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. 2019. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7152–7159.

[43] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[44] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs. In *CVPR*. IEEE Computer Society, 6857–6866.

[45] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[46] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 9 (2019), 2251–2265.

[47] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature Generating Networks for Zero-Shot Learning. In *CVPR*. IEEE Computer Society, 5542–5551.

[48] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *AAAI*.

[49] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied Knowledge Representation Learning. In *IJCAI*. 3140–3146.

[50] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[51] Li Zhang, Tao Xiang, and Shaogang Gong. 2017. Learning a Deep Embedding Model for Zero-Shot Learning. In *CVPR*. IEEE Computer Society, 3010–3019.

[52] Ming Zhao, Weijia Jia, and Yusheng Huang. 2020. Attention-Based Aggregation Graph Networks for Knowledge Graph Information Transfer. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 542–554.

[53] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. 2018. A Generative Adversarial Approach for Zero-Shot Learning From Noisy Texts. In *CVPR*. IEEE Computer Society, 1004–1013.

[54] Yizhe Zhu, Jianwen Xie, Bingchen Liu, and Ahmed Elgammal. 2019. Learning Feature-to-Feature Translator by Alternating Back-Propagation for Generative Zero-Shot Learning. In *ICCV*. 9843–9853.

A APPENDIX: DATASETS CONSTRUCTION AND ATTRIBUTE ANNOTATION

In this appendix, we provide more details of ImNet-A and ImNet-O (from ImageNet) and their manually annotating attributes.

A.1 Extracting Classes

ImageNet [9] is a widely used image classification dataset consisting of labeled images of 21 thousand classes. We conditionally extract different class families from its fine-grained parts. Classes in a family have the same type, in which 1) the unseen classes are 1-hop or 2-hops away from the seen classes according to the WordNet texonomy (enabling the transferability from seen classes to unseen classes); 2) the total number of seen and unseen classes is more than 3 (making the classification is fine-grained); and 3) each class has a Wikipedia entry (ensuring valid attribute descriptions from Wikipedia for human annotation).

With these conditions, we extract a domain-specific subset ImNet-A, which consists of 11 animal families, such as *Bees*, *Ants*, and *Foxes*, and a general subset ImNet-O, which consists of 5 general object families, such as *Snack Food* and *Fungi*.

A.2 Preparing Attribute List

Before annotating attributes, we first prepare the candidate attribute list. Inspired by the attribute annotations of AwA, which describe the color, shape, texture and important parts of objects, we reuse some attributes from it as well as extract textual phrases which characterize the appearances of classes from Wikipedia entries. Fox example, one sentence of Wikipedia that describes the class *Spoonbill* is: "Spoonbills are most distinct from the ibises in the shape of their bill, which is long and flat and wider at the end.", from which we can conclude the attribute *long flat and wider bill*.

A.3 Class-Specific Attribute Annotation

We invite volunteers to manually annotate attributes for these classes. Specifically, for each class, annotators are asked to assign $3 \sim 6$ attributes from the attribute list with 25 images given as the reference. Each class is reviewed by $3 \sim 4$ volunteers, and we take the consensus as the final annotations. Finally, we annotate a total of 85 attributes for ImNet-A classes and 40 attributes for ImNet-O classes. We associate these attributes with their corresponding classes to construct the ontological schema.