Hyperbolic Visual Embedding Learning for Zero-Shot Recognition

Shaoteng Liu[1,2] Jingjing Chen[1†] Liangming Pan[3] Chong-Wah Ngo[4] Tat-Seng Chua[3] Yu-Gang Jiang[1]
[1]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
[2] Xi'an Jiaotong University [3] National University of Singapore [4]City University of Hong Kong

Abstract

This paper proposes a Hyperbolic Visual Embedding Learning Network for zero-shot recognition. The network learns image embeddings in hyperbolic space, which is capable of preserving the hierarchical structure of semantic classes in low dimensions. Comparing with existing zero-shot learning approaches, the network is more robust because the embedding feature in hyperbolic space better represents class hierarchy and thereby avoid misleading resulted from unrelated siblings. Our network outperforms exiting baselines under hierarchical evaluation with an extremely challenging setting, i.e., learning only from 1,000 categories to recognize 20,841 unseen categories. While under flat evaluation, it has competitive performance as state-of-the-art methods but with five times lower embedding dimensions. Our code is publicly available *.

1. Introduction

Real-world image recognition applications are usually faced with thousands of object classes. Collecting sufficient training data for each class is time-consuming and sometimes infeasible. Therefore, Zero-Shot Learning (ZSL) [34, 35, 9], which aims to recognize the novel categories which are unseen during the training phase, has become an important research problem that needs to study.

Nevertheless, zero-shot learning is generally regarded as a difficult problem. As reported in [17], for generalized large-scale zero-shot image recognition, the best performance attained on the ImageNet dataset (with 2,2841 class) is less than 10% in terms of top-5 accuracy, which is far away from real-world applications. On the one hand, the class defined in the ImageNet is organized according to the WordNet hierarchy, including both general and fine-grained objects. For example, there are hundreds of species of dogs; distinguishing them with sufficient labeled training data is difficult, not mentioning the situation when the training sample is unavailable. Therefore, learning a model aiming to predict correct specific classes is not practical under large-scale zero-shot settings. To make ZSL more applicable in real-world applications, we notice that when the ZSL system cannot make a specific prediction targeted to the leaf class, users tend to tolerate a relative general but correct predictions rather than a specific but wrong prediction. For example, Figure 1 shows a part of the class hierarchy in the ImageNet. Given an unseen image of "red squirrel", users may prefer to have the prediction "tree squirrel" rather than "Kangaroo". Besides, making a slightly general but correct prediction makes it convenient to design the user interface for improving the results. Therefore, in this paper, we argue that a robust ZSL system should have the ability to output a correct but less fine-grained label (*e.g.*, the direct parent of the ground-truth label).

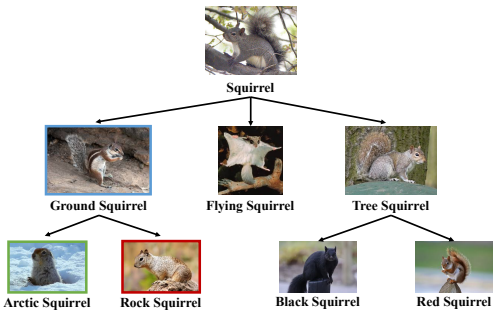

Figure 1. Given an unseen image from "Arctic Squirrel" (bounded in green), the proposed hyperbolic visual embedding learning networks will tend to predict to its immediate parent class "ground squirrel"(bounded in blue) while existing methods will predict it to its wrong sibling class "rock squirrel".

Existing works, however, are not designed to optimize the robustness of the zero-shot recognition systems. Generally, most of the existing works are implicit models that directly learn a mapping from visual space to semantic space [9] [29]. The semantic space are usually represented by semantic vectors, such as word vectors obtained from GloVe [28] or Word2Vec [24] models. As hierarchical relations among categories are not encoded in the semantic space, these models are unlikely to fulfill the robust recognition. More recent works propose to introduce class hierarchy by modeling the hierarchical relation with graph neural net-

†Corresponding author. Email: chenjingjing@fudan.edu.cn
*https://github.com/ShaoTengLiu/Hyperbolic_ZSL

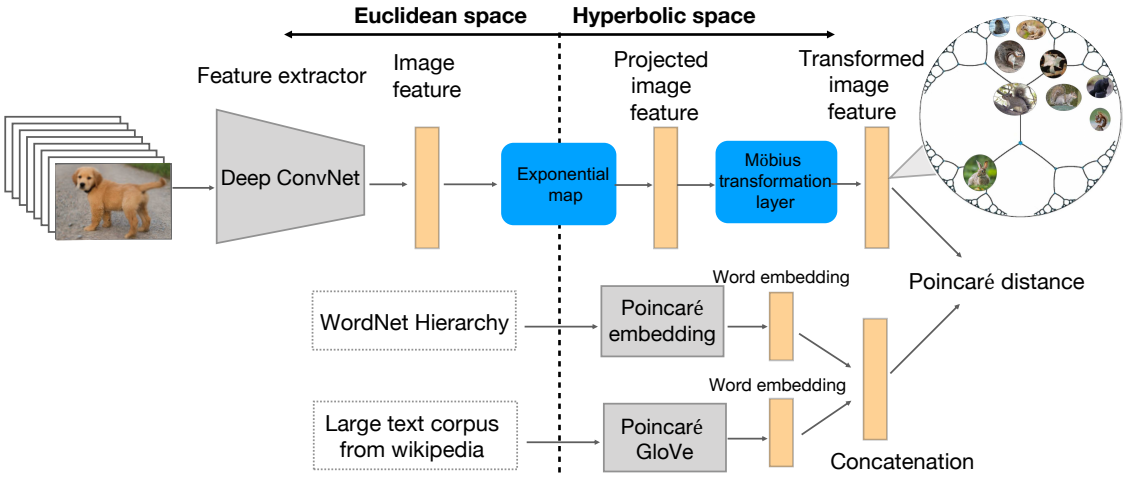

Figure 2. The general model framework. For a given image, our model first extracts its visual features using deep ConvNet. The extracted features are then projected to the hyperbolic space via the exponential map and the Möbius transformation to align with the class embeddings, which is learned by combining two kinds of embeddings in the hyperbolic space (Poincaré embeddings and Poincaré Glove).

work [34] [17]. With graph propagation, the classifier weights of well-trained seen classes are propagated to the unseen ones. Compared to the implicit methods, these models which explicitly utilize the hierarchy structure (explicit models) are more robust and effective. However, as the hierarchical relationship is reflected in the classifier learned in the Euclidean space, it cannot guarantee that the class is closer to its immediate parent node compared to its unrelated siblings. This becomes even harder when the number of unseen classes increases.

In this paper, we find that hyperbolic space is well-suited to address the aforementioned problems, leading to a more robust ZSL model. Hyperbolic space is a kind of manifold space studied in the Riemannian Geometry, in which basic mathematical operations (*e.g.*, distance measurement) are defined differently from the Euclidean space. It has been shown that Hyperbolic space is particularly suitable for modeling hierarchical data [26, 13]. For example, we can represent a tree with a branching factor of b with a two-dimensional embedding in hyperbolic space such that its structure is reflected in the embeddings. This property enables us to encode the hierarchical structure of classes in low dimensional space, resulting in a light-weight ZSL model. More importantly, in hyperbolic space, a class is closer to its immediate ancestors while distant from its siblings, which perfectly meet our requirements on the robustness. With the image embeddings learned in the hyperbolic space, our model is naturally endowed with robustness.

Therefore, we propose a novel ZSL framework (shown in Figure 2) that learns the hierarchy-aware image embedding features in the hyperbolic space. In our framework, image labels are projected into the hyperbolic space with Poincaré hierarchy embedding model [26] and Poincaré Glove [33]. Poincaré hierarchy embedding model [26] learns the label embeddings that preserve the hierarchical information, while Poincaré Glove [33] captures the semantic information. Meanwhile, image features extracted from DCNN in the Euclidean space are firstly projected into hyperbolic space with the exponential map and then aligned with the corresponding hyperbolic label embeddings through learning a Möbius version transformer network. The objective of the Möbius version transformer network is to minimize the Poincaré distance from image embeddings to their label embeddings in the hyperbolic space. During testing, the label of an unseen image can be obtained by searching the label embeddings that have the minimum Poincaré distance with its image embeddings. The contributions of this work are summarized as follows:

- We propose the Hyperbolic Visual Embedding Learning Network that learns hierarchical-aware image embeddings in hyperbolic space for ZSL. As far as we know, this is the first attempt to introduce Non-Euclidean Space for zero-shot learning problem.
- We conduct both empirical and analytic studies to demonstrate that introducing hyperbolic space into ZSL problem results in a model that produces more robust predictions.

2. Related Work

As image recognition systems have achieved near-human accuracy when training samples are ample [15], recent research focus has shifted to the problem of zero-shot image recognition [16, 38, 22, 8, 14, 12, 37], a challenging but more practical setting where the recognition is performed on categories that were unseen during the training.

Early works on zero-shot learning mostly rely on semantic attributes including both user-defined attributes [18, 19] and data-driven attributes [10, 23], which are automatically

discovered from visual data. These attributes are then used as the intermediate representations for knowledge transfer across classes, supporting zero-shot recognition of unseen classes. Recent works on zero-shot learning are mostly based on deep learning technologies and basically can be grouped into two major paradigms. The first paradigm is based on semantic embeddings (implicit knowledge) which directly learn a mapping from visual space to semantic space [4, 5, 11, 9, 36, 29, 31], represented by semantic vectors such as word vectors. For example, Socher *et al.*, [32] proposed to learn a linear mapping to align the image embeddings and the label embeddings learned from two different neural networks. Motivated by this work, Frome *et al.*, [9] proposed the *DeViSE* model to train this mapping using a ConvNet and a transformation layer, which showed that this paradigm can be exploited to make predictions about tens of thousands of unseen image labels. Instead of training a ConvNet to match the image features and the category embeddings, Norouzi et al. [27] proposed to map image features into the semantic embedding space via convex combination, which requires no additional training.

Instead of representing image categories as semantic embeddings, the second paradigm directly models the relations between categories for zero-shot recognition. For example, Salakhutdinov *et al.*, [30] used WordNet hierarchy structure to share knowledge among different classifiers so that the knowledge of seen categories can be propagated to unseen categories. Knowledge graph are found to be effective to perform this knowledge propagation. Deng *et al.*, [6] applied knowledge graph to organize the attributes relations between objects and use it to propagate knowledge of seen categories to unseen categories.

Recently, it has been demonstrated in [34] [17] that combining both implicit knowledge and explicit knowledge enables the model to achieve better recognition performance. As illustrated in [34], by leveraging Graph Convolutional Network (GCN) to combine semantic embedding and class hierarchy, it achieves the state-of-the-art results for zero-shot recognition on the ImageNet dataset (*e.g.* "2-hops","3-hops" and "All"), which almost doubles the performance of the model with only semantic embedding. Similar to [34] and [17], our work also utilizes both semantic embedding and class hierarchy, taking advantages of both implicit knowledge and explicit knowledge for zero-shot recognition. However, different to [34] and [17], our work models the implicit knowledge and explicit knowledge in the hyperbolic space. The explicit knowledge — the WordNet hierarchy is encoded with hierarchical-aware Poincaré embeddings, which better captures the class hierarchy with fewer dimensions. Meanwhile, the semantic embeddings are learned in hyperbolic space with Poincaré GloVe. As hyperbolic space is more suitable for modeling the hierarchical data, which endows our model with better robustness than existing models working in the Euclidean Space.

3. Preliminary

Hyperbolic space is an important concept in hyperbolic geometry, which is considered as a special case in the Riemannian geometry. Before presenting our proposed model, this section introduces the basic information of Riemannian geometry and hyperbolic space.

3.1. Basics of Riemannian Geometry

Manifolds are the generalization of curved surfaces that are studied in differential geometry. Riemannian geometry is one of the branches of differential geometry that studies with a Riemannian metric. Each point on the manifold can be assigned a curvature. When the curvature is a negative constant, the geometry becomes hyperbolic geometry. For a point x in a manifold $\mathcal{M}$, one can define the *tangent space* $T_x\mathcal{M}$ of $\mathcal{M}$ at x as a vector space that contains all possible directions in which one can tangentially pass through. An inner product can be defined on $T_x\mathcal{M}$. A Riemannian metric g on $\mathcal{M}$ is a collection of inner products $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$, $x \in \mathcal{M}$. A Riemannian manifold $(\mathcal{M}, g)$ is a manifold $\mathcal{M}$ together with a Riemannian metric g. Based on these concepts, we introduce the following definitions:

• **Geodesic** is the shortest curve between two points, analogous to a straight line in the Euclidean space.

• **Parallel Transport** is a way of transporting tangent vectors along smooth curves, such as geodesic, in a manifold, which can be formulated as $P_{x\to y} : T_x\mathcal{M} \to T_y\mathcal{M}$.

• **Exponential Map** is a map from a subset of a tangent space $T_x\mathcal{M}$ of a Riemannian manifold $\mathcal{M}$ at point x to $\mathcal{M}$ itself, which provides a way to project a vector in Euclidean space to hyperbolic space. For any tangent vector $v \in T_{\mathbf{0}}\mathcal{M} \setminus \{\mathbf{0}\}$, the exponential map $\exp_{\mathbf{0}} : T_{\mathbf{0}}\mathcal{M} \to \mathcal{M}$ is formally defined as follows:

$$\exp_{\mathbf{0}}(v) = \tanh(\|v\|)\frac{v}{\|v\|}, \tag{1}$$

where we choose $\mathbf{0}$ as the reference point. We use exponential map to project image features learned in the Euclidean space to the hyperbolic space, which will be elaborated later. The reverse process of exponential map is the *logarithmic map* $\log_{\mathbf{0}}(y) : \mathcal{M} \to T_{\mathbf{0}}\mathcal{M}$. Obviously, $\log_{\mathbf{0}}(\exp_{\mathbf{0}}(v)) = v$.

3.2. Poincaré Ball.

There are 5 common models for hyperbolic space. Among them, the Poincaré ball model and the Lorentz model are most commonly used in machine learning. Similar to [26], we choose Poincaré Ball as the embedding model because its distance function is differentiable and

it has a relatively simple constraint on the representations. Poincaré ball is a model in which the points are inside a unit ball. It can be defined as a manifold $(\mathbb{D}^n, g^{\mathbb{D}})$, where $\mathbb{D}^n = \{x \in \mathbb{R}^n : \|x\| < 1\}$ is the n-dimensional hyperbolic space within the Poincaré ball. The Riemannian metric of a Poincaré ball is given as:

$$g_x^{\mathbb{D}} = \lambda_x^2 g^E, \text{ where } \lambda_x = \frac{1}{1 - \|x\|^2} \tag{2}$$

where $g^E = I_n$ is the Euclidean metric tensor and x is a point on Poincaré ball. The Poincaré distance between two points (x, y) can be induced with the Riemannian metric as follows:

$$d_{\mathbb{D}}(x, y) = \cosh^{-1}(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|)(1 - \|y\|)}) \tag{3}$$

Because the Poincaré ball is conformal to the Euclidean space, the definition of angle is the same for these two spaces. Formally, the angle between two vectors (u, v) in the Poincaré ball is defined as:

$$\cos(\angle(u, v)) = \frac{g_x^{\mathbb{D}}(u, v)}{\sqrt{g_x^{\mathbb{D}}(u, u)}\sqrt{g_x^{\mathbb{D}}(v, v)}} \tag{4}$$

4. Approach

The proposed framework aims to learn the embeddings of both images and labels in the hyperbolic space such that the hierarchical information, as well as semantic information, can be well preserved with a few dimensions for zero-shot recognition. As shown in Figure 2, the proposed framework consists of two modules: (1) **Hyperbolic Label Embedding Learning**, which embeds image labels $\mathcal{C}$ into a Hyperbolic space, denoted as $\mathcal{H}$, encoding both hierarchical information (via Poincaré Embedding) and semantic information (via Poincaré Glove) among classes; (2) **Poincaré Image Feature Embedding learning**, which learns the image embeddings in hyperbolic space $\mathcal{H}$ that are nearest to the corresponding poincaré label embeddings.

4.1. Hyperbolic Label Embedding Learning

For text labels, two hyperbolic embedding models are investigated for embedding learning: Poincaré hierarchy embedding model [26] and Poincaré Glove [33]. The former one learns the hyperbolic word embeddings with WordNet hierarchy [25] while the latter one embeds labels with a hyperbolic version of the GloVe [28]. The final label embeddings are obtained by combining the embeddings learned from both models.

Poincaré Embedding. Following [26], we embed the WordNet Noun hierarchy into the Poincaré ball. WordNet Noun hierarchy includes 82,115 synsets (nodes) and 743,241 hypernymy relations (edges). We learn the embeddings for each synset such that the distances of synset pairs are preserved in the Poincaré ball. To this end, we employ a training objective to ensure that the distance between nodes with hypernymy relationship is minimized, while the distance is maximized for nodes without a relationship. Formally, let $\mathcal{E} = \{(u, v)\}$ be the set of observed hypernymy relations between two classes $u, v \in \mathcal{C}$, we minimize the following objective in the hyperbolic space.

$$\mathcal{L} = -\sum_{(u,v)\in\mathcal{E}} \log \frac{e^{-d_{\mathbb{D}}(u,v)}}{\sum_{v'\in\mathcal{N}(u)} e^{-d_{\mathbb{D}}(u,v')}} \tag{5}$$

where $\mathcal{N}(u) = \{v \mid (u, v) \notin \mathcal{E}\} \cup \{u\}$ is the set of negative examples for u (including u). Same to [26], we randomly sample 10 negative examples per positive example during the training. As we train in the hyperbolic space, the distance metric is replaced with the Poincaré distance defined in Equation 3. After training, we obtain the embedding for each WordNet synset that can be mapped to an image class.

Poincaré GloVe. Besides the hierarchy structure, semantic relations between image classes also play a vital role in ZSL. Semantic information for an image class is often obtained by learning the word embedding of the class label. GloVe [28] is a commonly-used method to learn word embeddings in the Euclidean space based on word co-occurrences in a large text corpus. Semantic relations between words are then reflected by their distances in the embedding space. To capture the semantic relations among image classes, we train a GloVe model in Poincaré ball following the method of [33]. In case when an image class has multiple senses, we only select the first sense in the WordNet synset to learn semantic embedding. The major challenge of training the Poincaré GloVe is that there is no clear definition for inner-product in hyperbolic space. Following [33], we replace the inner-product in the original GloVe loss function with the Poincaré distance defined in Equation 3, resulting in a hyperbolic version of GloVe loss function J as follows.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(-\frac{1}{2}d_{\mathbb{D}}^2(w_i, \tilde{w_j}) + b_i + \tilde{b_j} - \log X_{ij})^2, \tag{6}$$

where V is the size of the text corpus, X_{ij} is the number of times that words i and j occur in the same window context, w_i is an embedding of word i, $\tilde{w_j}$ is an embedding of a context word j, and $d_{\mathbb{D}}$ is the Poincaré distance in Equation 3. For the training corpus, we use the English Wikipedia dump with 1.4 billion tokens provided by [21] and [20].

Feature Fusion. We concatenate Poincaré embedding p_c and Poincaré GloVe embedding q_c to form the final class embedding t_c in the Poincaré ball. The embedding contains both the structural and semantic information of the class.

However, although the norms of p_c and q_c are both smaller than 1 (the radius of Poincaré ball), the norm of the concatenated vector t_c may be greater than one, which may move it outside of the Poincaré ball. To address this problem, we use the exponential map to project the vector back to the ball. This gives the final representation of t_c as follows.

$$t_c = \exp_{\mathbf{0}}([p_c; q_c]) \tag{7}$$

where $\exp_{\mathbf{0}}$ is the exponential map defined in Equation 1.

4.2. Hyperbolic Image Embedding Learning

ResNet [15] is employed to extract feature $v_{\mathcal{I}}$ from unseen image $\mathcal{I}$. $v_{\mathcal{I}}$ is a 2,048-dimensional vector in the Euclidean space. The hyperbolic visual feature transform network is proposed to project $v_{\mathcal{I}}$ to the hyperbolic space and align with its class label. The transformation network consists of an *exponential map* for projecting image features into the hyperbolic space, and a *Möbius transformation network* for aligning images to labels.

Exponential Map. Using the exponential map in Equation 1, we first project the image features $v_{\mathcal{I}}$ into the Poincaré Ball. By choosing the Euclidean space as the tangent space of hyperbolic space, Exponential map can project the image features in Euclidean space into the hyperbolic space, as shown below:

$$\tilde{v}_{\mathcal{I}} = \exp_{\mathbf{0}}(v_{\mathcal{I}}) \tag{8}$$

where $\tilde{v}_{\mathcal{I}}$ is the projected image feature of image $\mathcal{I}$.

Möbius Transformation. We then train a Möbius transformer to align the projected image features $\tilde{v}_{\mathcal{I}}$ with the corresponding label embeddings $t_{c_{\mathcal{I}}}$. Our Möbius transformer is essentially a two-layer feed forward neural network implemented in hyperbolic space. For an arbitrary function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ in the Euclidean space, the Möbius version of f is a function that maps from $\mathbb{D}^n$ to $\mathbb{D}^m$ in the hyperbolic space:

$$f^{\otimes}(x) = \exp_{\mathbf{0}}(f(\log_{\mathbf{0}}(x))) \tag{9}$$

where $\exp_{\mathbf{0}}\colon T_{0_m}\mathbb{D}^m \to \mathbb{D}^m$ and $\log_{\mathbf{0}}\colon \mathbb{D}^n \to T_{0_n}\mathbb{D}^n$. When $M\colon \mathbb{R}^n \to \mathbb{R}^m$ is a linear map and $Mx \neq 0$, substituting M into Equation 9 obtains the Möbius matrix-vector multiplication $M^{\otimes}(x)$ as follows.

$$M^{\otimes}(x) = \tanh\left(\frac{\|Mx\|}{\|x\|}\tanh^{-1}(\|x\|)\right)\frac{Mx}{\|Mx\|} \tag{10}$$

Based on Equation 10, we can implement a feed forward layer in hyperbolic space, and our Möbius Transformation is a two-layer network constructed by stacking two feed forward layers. After the Möbius transformation, the projected image features $\tilde{v}_{\mathcal{I}}$ are transformed to image embedding features (denoted as $h_{\mathcal{I}}$), which have the same dimension with the label embeddings in hyperbolic space. With this, a class label that is nearest to the image feature is assigned.

Model Training. Rank loss is employed as the loss function for model training, with the objective of minimizing the distance between the image embeddings $h_{\mathcal{I}}$ learned from Möbius transformer and its label embeddings $t_{c_{\mathcal{I}}}$. As the model is trained in hyperbolic space, we use the Poincaré distance defined in Equation 3 as the distance metric. The input of the loss function is a triplet $\langle h_{\mathcal{I}}, t_{c_{\mathcal{I}}}, t^-_{c_{\mathcal{I}}}\rangle$, in which $t^-_{c_{\mathcal{I}}}$ denote a random sample with negative label. Let the margin as $\delta \in (0, +\infty)$, the loss function is defined as

$$L = \max[0, \delta + d_{\mathbb{D}}(t_{c_{\mathcal{I}}}, h_{\mathcal{I}}) - d_{\mathbb{D}}(t_{c_{\mathcal{I}}}, t^-_{c_{\mathcal{I}}})]. \tag{11}$$

For optimization, we adopt different optimization tools in hyperbolic space. The training of Poincaré Glove is optimized by RADAGRAD [1]. For Poincaré embedding, RADAGRAD is not suitable as it requires the hyperbolic space to be a product of Riemannian manifolds. Therefore, we train the Poincaré embedding using full Riemannian stochastic gradient descent (RSGD) [2, 13] and set the learning rate to 0.01.

5. Experiment

5.1. Datasets

The experiments are conducted on ImageNet [7], which is a popular benchmark for ZSL [27, 9, 34, 17]. The benchmark includes 1,000 known classes from the ImageNet 2012 1K dataset. The unseen classes are distributed into three datasets of "2-hops", "3-hops" and "All" concepts from the 1K known concepts based on WordNet hierarchy. For example, the "2-hops" unseen concepts are within 2 hops of known concepts, totaling to 1,589 classes. The "3-hops" dataset has 7,860 classes, while the "All" dataset includes all the 20,841 classes in ImageNet. Note that there is no overlap between the seen and unseen classes in the three datasets. The difficulty of a dataset is proportional to the number of unseen classes.

We evaluate our model on these three datasets for both the Zero-Shot Learning (ZSL) setting and the Generalized Zero-Shot Learning (GZSL) setting. In ZSL, we only evaluate the model performance on the unseen classes, *i.e.*, the model recognizes which unseen class that a testing sample belongs to. However, in GZSL, we use all classes (the union of seen and unseen classes) as the candidate set to evaluate the model. We name the GZSL setting of the above 3-datasets as: "2-hops + 1K", "3-hops + 1K", and "All + 1K". GZSL is a more challenging setting. We adopt the same train/test split settings as [9, 34].

5.2. Baselines

We compare our model against several state-of-the-art ZSL baselines on both ZSL and GZSL settings. The base-

lines are listed as follows. **DeViSE** [9]: It is a typical implicit knowledge transfer method that linearly maps visual features to the semantic word-embedding space by learning a transformation using the hinge ranking loss. **DeViSE***: We enhance the original DeViSE by concatenating hierarchical embeddings of texts with semantic embeddings during joint space learning. The hierarchical embeddings are with the same dimension with concatenating the same dimensional hierarchical embeddings with Poincaré embeddings for a fair comparison, and are learned by using the loss function in [26] which encourages semantically similar objects to be close in the embedding space according to their Euclidean distance.**ConSE** [27]: ConSE changes the feature transformation of DeViSE to a convex combination of the semantic embeddings from the T-closest seen classes, weighted by the probabilities that the image belongs to the seen classes. **SYNC** [3]: SYNC is another implicit knowledge transfer method that aligns the semantic space with the visual model by adding a set of phantom object classes, based on which new embedding is derived as a convex combination of these phantom classes. **GCNZ** [34]: GCNZ leverages both implicit knowledge and explicit knowledge by using Word2vec embeddings to represent class labels and leveraging GCN to model the class relations for unseen class prediction. The Word2vec embeddings are used as the inputs of GCN, and classifier weights of seen classes are transferred to unseen classes during graph propagation. **DGP** [17]: As an improved version of GCNZ, DGP proposes a dense graph propagation module to alleviate the dilution of knowledge from distant nodes.

5.3. Experimental Settings

For Poincaré embedding, we use the 100-dimensional vector trained on the transitive closure of the WordNet Noun hierarchy by [26]. For Poincaré Glove, we train a 100-dimensional vector for each image class based on the English Wikipedia dump (containing 1.4 billion tokens) provided by [21] and [20]. At last, we obtain a combined class embedding of 200 dimension. The Möbius feature transformation is trained for 2000 epochs with a learning rate of 0.01 using RSGD [2, 13], with the input as a $2,048$-dimensional projected image feature vector and output as a 200-dimensional vector. The margin δ in Equation 11 is set as 1 through cross-validation. The model is implemented by PyTorch, training on four GTX 1080Ti GPUs.

5.4. Hierarchical Evaluation

The standard evaluation metric for ZSL is the *Top-k Hit Ratio* (Hit@k), which measures the percentage of hitting the ground-truth labels among the top-k positions of prediction. However, this metric does not reflect the robustness of a ZSL model. Therefore, we propose a hierarchical evaluation metric that expands the GT label with its immediate parent class. The ability to predict the immediate parent class accurately reflects the robustness of a ZSL model. To illustrate this, Figure 1 shows a part of the class hierarchy in ImageNet. Given an image of "red squirrel", when the model cannot make the correct prediction, a more robust ZSL model should be able to output the second-best prediction, *i.e.*, classifying the image as its parent class "tree squirrel", which is more acceptable than assigning a wrong label from another branch of the hierarchy, such as "Kangaroo". A ZSL system equipped with this ability is more applicable in real-world applications, as a simple UI can be designed to help users select the correct leaf class.

Table 1. Top-k accuracy for different models on the ImageNet dataset for hierarchical evaluation. The candidates become the categories in "hops" test set and the parent of them. The baseline models are re-implemented by us. For all models, the image features are extracted with ResNet-101.

Data Set	Model	Hierarchical precision@k(%)				
		1	2	5	10	20
2-hops & Their Parents	DeViSE	3.2	5.3	9.5	15.6	21.2
	DeViSE*	4.5	7.0	9.9	15.6	22.0
	ConSE	4.2	6.8	12.3	18.5	25.1
	GCNZ	9.2	15.6	27.5	36.8	44.5
	Ours	**16.6**	**24.3**	**43.8**	**58.6**	**70.3**
3-hops & Their Parents	DeViSE	1.3	2.1	3.3	4.9	7.3
	DeViSE*	1.7	2.6	4.4	6.6	9.3
	ConSE	1.9	2.6	4.4	7.2	9.7
	GCNZ	2.7	4.6	8.2	12.5	15.1
	Ours	**7.9**	**12.5**	**21.4**	**28.7**	**37.5**
All	DeViSE	0.9	1.5	2.9	4.4	6.5
	DeViSE*	1.0	1.6	2.9	4.4	6.5
	ConSE	1.5	2.4	4.2	6.5	9.7
	GCNZ	2.2	3.8	7.2	10.5	13.9
	Ours	**5.1**	**6.9**	**12.9**	**16.5**	**19.3**

Under the hierarchical evaluation, given a test image $\mathcal{I}$ whose class label is c (*e.g.*, red squirrel), we set both c and the immediate parent class of c (*e.g.*, tree squirrel) as the ground-truth and evaluate the score of Hit@k. Table 1 lists the performance comparison results. Compared to DeViSE, DeViSE* achieves better performances, showing the advantages of introducing hierarchical embedding in ZSL under the hierarchical evaluation. Besides, it is obvious that our model significantly outperforms all baselines, even triples the performances of DeViSE* and doubles the performance of another strong baseline - GCNZ. The results demonstrate that hyperbolic space can better capture the class hierarchy, resulting in a robust ZSL model that tends to assign a general class (*e.g.*, squirrel) to an unseen image even if it cannot locate its exact specific class (in this case, red squirrel). The superior performance of our method is due to the nature of the class distribution in the hyperbolic space; a class tends to be close to its parent while distant from its sibling in the hyperbolic space. We believe this properly achieved by

DeViSE: teddy, orangutan, valley, langur, cliff
GCNZ: phalanger, **red squirrel**, kangaroo, lemur, tree wallaby
Ours: **red squirrel**, **tree squirrel***, squirrel, kangaroo, phalanger

DeViSE: rugby ball, soccer ball, golf ball, basketball, cricket
GCNZ: **volleyball**, basketball, golf ball, punching bag, rugby ball
Ours: **volleyball**, **ball***, basketball, rugby ball, soccer ball

DeViSE: bullet train, freight car, school bus, police van, minibus
GCNZ: mail train, express, **passenger train**, cargo ship, shuttle bus
Ours: **passenger train**, **railroad train***, bus, school bus, trolleybus

Figure 3. Qualitative result comparison. The true category is highlighted in bold. The direct parent category of the true category is highlighted in bold and with a "*". We list the top-5 predictions.

hyperbolic embedding is well-suited for a real-world ZSL system. We further show some prediction examples in Figure 3. For an image of "red squirrel", although both GCNZ and our model rank the true label into top-2, our model successfully ranks the direct parent "tree squirrel" into the top-2. For an image of "volleyball", our model successfully ranks the direct parent "ball" into top-2.

5.5. Performance Comparison

We then conduct the performance comparison on the standard ZSL and GZSL settings, respectively. The results are summarized in Table 2 and Table 3, based on which we have four major observations.

First, the methods that consider both implicit knowledge and explicit knowledge basically outperform the method with implicit knowledge only. GCNZ, DGP and our method outperform DeViSE, ConSE, and SYNC in terms of all evaluation metrics with a large margin on both ZSL and GZSL settings. In the ZSL setting, when using ResNet-50 for image feature extraction, our model outperforms DeViSE, ConSE, and SYNC by 98.5%, 74.3% and 38.5% in terms of Hit@1, respectively. It is worthwhile to note that compared to DeViSE, DeViSE* that concatenates both hierarchical embeddings and semantic embeddings for joint embedding space learning preforms even worse, which suggests that directly concatenating different types of word embeddings in Euclidean space for may not be the correct way to make use of different types of knowledge.

Second, our method shows more stable results compared with other implicit methods. For example, when using ResNet-50 for image feature extraction, the performance drop of Hit@1 for DeViSE, ConSE, and SYNC are 74.6%, 71.4%, and 75.9%, when changing the test set from "2-hops" to "3-hops". Our method only suffers from a 51.1% drop in Hit@1. In Section 5.4, we have shown that in the hyperbolic space, a class is close to its parent while distant from its sibling. This makes it hard for our model to misclassify an image to its sibling. However, this error is easy to happen in the Euclidean space, as classes sharing a common ancestor tend to be clustered together.

Third, compared with state-of-the-arts explicit knowledge transfer methods based on GCN, including GCNZ and DGP, our model has lower performance on "2-hops" test set. However, on "3-hops" and "All", our model achieves comparable results with them, with better Hit@1, Hit@2, and Hit@5 but slightly worse Hit@10 and Hit@20. As

Table 2. Top-k accuracy of different methods on ZSL setting.

Test Set	Model	ConvNets	Flat Hit@k(%)				
			1	2	5	10	20
2-hops	DeViSE (us)	ResNet-50	6.7	11.2	19.4	28.1	38.3
	DeViSE* (us)	ResNet-50	6.1	10.6	18.8	27.4	37.2
	ConSE [3]	Inception-v1	8.3	12.9	21.8	30.9	41.7
	ConSE [3]	ResNet-50	7.63	-	-	-	-
	SYNC [3]	Inception-v1	10.5	16.7	28.6	40.1	52.0
	SYNC[3]	ResNet-50	9.6	-	-	-	-
	GCNZ [34]	ResNet-50	19.8	33.3	53.2	65.4	74.6
	DGP [17]	ResNet-50	**26.2**	**40.4**	**60.2**	**71.9**	**81.0**
	Ours	ResNet-50	13.3	20.8	39.2	52.7	62.4
	Ours	ResNet-101	14.2	22.1	40.7	53.7	63.2
3-hops	DeViSE (us)	ResNet-50	2.1	3.5	6.3	9.5	14.1
	DeViSE* (us)	ResNet-50	1.9	3.3	6.0	9.1	13.6
	ConSE [3]	Inception-v1	2.6	4.1	7.3	11.1	16.4
	ConSE [3]	ResNet-50	2.18	-	-	-	-
	SYNC [3]	Inception-v1	2.9	4.9	9.2	14.2	20.9
	SYNC [3]	ResNet-50	2.31	-	-	-	-
	GCNZ [34]	ResNet-50	4.1	7.5	14.2	20.2	27.7
	DGP [17]	ResNet-50	6.0	10.4	18.9	**27.2**	**36.9**
	Ours	ResNet-50	6.5	10.6	18.8	25.8	35.2
	Ours	ResNet-101	**7.3**	**11.3**	**19.6**	26.3	35.7
All	DeViSE (us)	ResNet-50	1.0	1.8	3.0	4.6	7.1
	DeViSE* (us)	ResNet-50	0.9	1.7	2.9	4.3	6.8
	ConSE [3]	Inception-v1	1.3	2.1	3.8	5.8	8.7
	ConSE [3]	ResNet-50	0.95	-	-	-	-
	SYNC [3]	Inception-v1	1.4	2.4	4.5	7.1	10.9
	SYNC [3]	ResNet-50	0.98	-	-	-	-
	GCNZ [34]	ResNet-50	1.8	3.3	6.3	9.1	12.7
	DGP [17]	ResNet-50	2.8	4.9	9.1	**13.5**	**19.3**
	Ours	ResNet-50	3.7	5.9	10.3	13.0	16.4
	Ours	ResNet-101	**4.2**	**6.3**	**10.8**	13.3	16.6

Table 3. Top-k accuracy of different models on GZSL setting.

Test Set	Model	ConvNets	Flat Hit@k(%)				
			1	2	5	10	20
2-hops (+1K)	DeViSE (us)	ResNet-50	1.1	3.1	8.4	15	23.8
	DeViSE*(us)	ResNet-50	1.0	2.9	8.2	14.7	23.4
	ConSE [34]	ResNet-50	0.1	11.2	24.3	29.1	32.7
	GCNZ [34]	ResNet-50	9.7	20.4	42.6	57.0	68.2
	DGP [17]	ResNet-50	**11.9**	**27.0**	**50.8**	**65.1**	**75.9**
	Ours	ResNet-50	6.4	11.9	27.2	35.3	45.2
	Ours	ResNet-101	6.8	12.2	27.4	35.4	45.2
3-hops (+1K)	DeViSE (us)	ResNet-50	0.6	1.6	3.8	6.5	10.5
	DeViSE* (us)	ResNet-50	0.5	1.5	3.6	6.3	10.2
	ConSE [34]	ResNet-50	0.2	3.2	7.3	10.0	12.2
	GCNZ [34]	ResNet-50	2.2	5.1	11.9	18.0	25.6
	DGP [17]	ResNet-50	3.2	7.1	**16.1**	**24.6**	**34.6**
	Ours	ResNet-50	**3.6**	**8.7**	15.3	20.5	29.1
	Ours	ResNet-101	3.7	8.8	15.3	20.5	29.1
All (+1K)	DeViSE (us)	ResNet-50	0.3	0.9	2.2	3.6	5.8
	DeViSE* (us)	ResNet-50	0.3	0.8	2.0	3.4	5.5
	ConSE [34]	ResNet-50	0.1	1.5	3.5	4.9	6.2
	GCNZ [34]	ResNet-50	1.0	2.3	5.3	8.1	11.7
	DGP [17]	ResNet-50	1.5	3.4	7.8	12.3	**18.2**
	Ours	ResNet-50	**2.2**	**4.6**	**9.2**	**12.7**	15.5
	Ours	ResNet-101	2.3	4.6	9.2	12.7	15.5

Table 4. Effect of different hyperbolic label embeddings. Image features are extracted with ResNet-101. The testing is done on unseen categories.

Test Set (# Categories)	Model	Flat Hit@k(%)				
		1	2	5	10	20
2-hops (1,589)	PH Only	12.1	18.5	34.4	46.5	53.6
	PG Only	10.6	15.4	32.6	43.1	51.7
	PH + PG	**14.2**	**22.1**	**40.7**	**53.7**	**63.2**
3-hops (7,860)	PH Only	6.2	9.7	16.1	23.5	31.5
	PG Only	3.7	7.3	13.1	18.8	26.3
	PH + PG	**7.3**	**11.3**	**19.6**	**26.3**	**35.7**
All (20,841)	PE Only	3.6	4.6	8.8	10.0	13.7
	PG Only	2.4	3.3	6.5	8.6	10.7
	PE + PG	**4.2**	**6.3**	**10.8**	**13.3**	**16.6**

classifier weights are directly shared among nearby classes, knowledge transfer methods that base on graph propagation tend to perform better when the seen classes and unseen classes are similar. However, their performance is less robust when the unseen classes are dominant in number and also distant from the seen classes (the cases of "3-hops" and "All"). For example, the Hit@1 of DGP drops significantly from 26.2 to 6.0 when changing the dataset from "2-hops" to "3-hops". The main reason could be the dilution of knowledge in long-distance graph propagation. On the contrary, our model performs more stable when it comes to "3-hops" and "All" as our model is based on feature mapping rather than weight propagation within an explicit graph.

Lastly, the performance drops for all the methods in GZSL as seen classes are mixed in the candidates. Similar results can be observed in this setting. Our model outperforms DeViSE and ConSE on all test sets, while achieving comparable performance with GCNZ and DGP on the "3-hops+1K" and "All+1K" test sets.

5.6. Ablation Study

We further perform ablation studies to show the effectiveness of combining structural information and semantic information. As shown in Table 4, we evaluate three versions of our model: the model with only Poincaré Hierarchy Embedding (PH Only), the model with only Poincaré GloVe (PG Only), and the model with both (PH + PG). It shows that the performances of PH+PG are consistently better than the model using only Poincaré Embedding or only Poincaré GloVe across all test sets. This demonstrates the complementary role of hierarchy information and semantic information in ZSL. We use a specific example to explain how these two embeddings complement each other. Take the images of Red Squirrel (Figure 1) for example, the top-5 predicted labels for Poincaré Embedding are: "Red Squirrel", "Tree Squirrel", "Squirrel", "Tree Wallaby", and "Kangaroo"; while the top-5 predicted labels for Poincaré GloVe are: "Red Squirrel", "Kangaroo", "Tree Wallaby", "Tree Squirrel", and "Lemur". We find that Poincaré Embedding tends to predict the general labels, such as "Tree Squirrel" and "Squirrel", as it mainly captures the class hierarchy. Conversely, Poincaré Glove tends to predict similar and specific labels (*e.g.*, "Kangaroo"), since it models the semantic similarity between different classes.

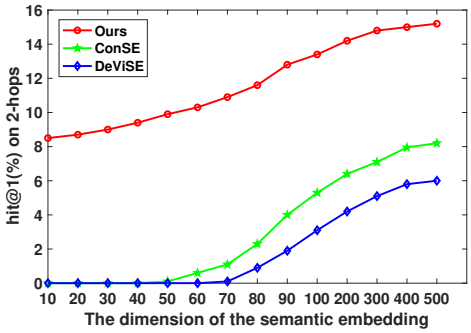

Figure 4. Performance comparison under different dimensions of semantic embeddings. The testing is performed on "2-hops" set.

5.7. Dimension Analysis

As the volume of hyperbolic space increases exponentially with the radius, the embedding dimension needed to represent the feature embeddings can be much lower than that in Euclidean space. To demonstrate this, we investigate the performance of our model regarding different semantic embedding dimensions. In Figure 4, we compare the performance of our model with DeViSE and ConSE. The Hit@1 on the "2-hops" dataset is reported for different embedding dimensions. As shown in the Figure, when the dimension of the semantic embedding decreases to 10, our model still achieves a satisfactory Hit@1 of 8.3. On the contrary, the performance DeViSE and ConSE, which learns the embeddings in Euclidean space, decreases to 0 as both models cannot converge in training with 10-dimensional semantic embeddings. The results clearly show the advantages of learning embeddings in hyperbolic space.

6. Conclusion

In this paper, we proposed the Hyperbolic Visual Embedding Learning Networks. As far as we know, this is the first attempt to introduce Non-Euclidean Space for ZSL problem. Furthermore, we conducted both empirical and analytic studies to demonstrate that introducing hyperbolic space into ZSL problem results in a more robust model. Under hierarchical evaluation, our framework outperforms existing baseline methods by a large margin.

7. Acknowledgement

This research is part of NExT++ research, which is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

References

[1] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *ICLR*, 2019.

[2] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[3] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[4] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*, 2017.

[5] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *CVPR*, 2018.

[6] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] Zhengming Ding and Hongfu Liu. Marginalized latent semantic encoder for zero-shot learning. In *CVPR*, 2019.

[9] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[10] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Attribute learning for understanding unstructured social activity. In *ECCV*, 2012.

[11] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016.

[12] Yanwei Fu, Xiaomei Wang, Hanze Dong, Yu-Gang Jiang, Meng Wang, Xiangyang Xue, and Leonid Sigal. Vocabulary-informed zero-shot and open-set learning. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[13] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML*, 2018.

[14] Tristan Hascoet, Yasuo Ariki, and Tetsuya Takiguchi. On zero-shot recognition of generic objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9553–9561, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] He Huang, Changhu Wang, Philip S. Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *CVPR*, 2019.

[17] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR*, 2019.

[18] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[19] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[20] Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, 2014.

[21] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[22] Jin Li, Xuguang Lan, Yang Liu, Le Wang, and Nanning Zheng. Compressing unknown images with product quantizer for efficient zero-shot classification. In *CVPR*, 2019.

[23] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR*, 2011.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[25] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[26] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, 2017.

[27] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.

[28] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[29] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[30] Ruslan Salakhutdinov, Antonio Torralba, and Joshua B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.

[31] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[32] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[33] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *ICLR*, 2019.

[34] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018.

[35] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *CVPR*, 2017.

[36] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015.

[37] Bo Zhao, Xinwei Sun, Yanwei Fu, Yuan Yao, and Yizhou Wang. Msplit lbi: Realizing feature selection and dense estimation simultaneously in few-shot and zero-shot learning. *arXiv preprint arXiv:1806.04360*, 2018.

[38] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Generalized zero-shot recognition based on visually semantic embedding. In *CVPR*, 2019.