VGSE: Visually-Grounded Semantic Embeddings for Zero-Shot Learning

Wenjia Xu1,7,8 Yongqin Xian2 Jiuniu Wang5,7,8 Bernt Schiele3 Zeynep Akata3,4,6
1 Beijing University of Posts and Telecommunications 2 ETH Zurich
3 Max Planck Institute for Informatics 4 University of Tübingen
5 City University of Hong Kong 6 Max Planck Institute for Intelligent Systems
7 University of Chinese Academy of Sciences 8 Aerospace Information Research Institute, CAS

Abstract

Human-annotated attributes serve as powerful semantic embeddings in zero-shot learning. However, their annotation process is labor-intensive and needs expert supervision. Current unsupervised semantic embeddings, i.e., word embeddings, enable knowledge transfer between classes. However, word embeddings do not always reflect visual similarities and result in inferior zero-shot performance. We propose to discover semantic embeddings containing discriminative visual properties for zero-shot learning, without requiring any human annotation. Our model visually divides a set of images from seen classes into clusters of local image regions according to their visual similarity, and further imposes their class discrimination and semantic relatedness. To associate these clusters with previously unseen classes, we use external knowledge, e.g., word embeddings and propose a novel class relation discovery module. Through quantitative and qualitative evaluation, we demonstrate that our model discovers semantic embeddings that model the visual properties of both seen and unseen classes. Furthermore, we demonstrate on three benchmarks that our visually-grounded semantic embeddings further improve performance over word embeddings across various ZSL models by a large margin. Code is available at https://github.com/wenjiaXu/VGSE

1. Introduction

Semantic embeddings aggregated for every class live in a vector space that associates different classes even when visual examples of these classes are not available. Therefore, they facilitate the knowledge transfer in zero-shot learning (ZSL) [1, 28, 42, 59] and are used as side-information in other computer vision tasks like fashion trend forecast [4, 23, 64], face recognition and manipulation [11, 27, 29], and domain adaptation [10, 24].

Human annotated attributes [19, 36, 55], characteristic properties of objects annotated by human experts, are widely used as semantic embeddings [61, 62]. However, obtaining attributes is often a labor-intensive two-step process. First, domain experts carefully design an attribute vocabulary, e.g., color, shape, etc., and then human annotators indicate the presence or absence of an attribute in an image or a class (as shown in Figure 1). The labeling effort devoted to human-annotated attributes hinders its applicability of performing zero-shot learning for more datasets in realistic settings [30].

Previous works tackle this problem by using word embeddings for class names [31, 38], or semantic embeddings from online encyclopedia articles [3, 39, 67]. Though they model the semantic relation between classes without using human annotation, some of these relations may not be visually detectable by machines, resulting in a poor performance in zero-shot learning. Similarly, discriminative visual cues may not all be represented in those semantic embeddings.

To this end, we propose the Visually-Grounded Semantic Embedding (VGSE) Network to discover semantic embeddings with minimal human supervision (we only use category labels for seen class images). Our network explicitly explores visual clusters that relate image regions from different categories, which is useful for knowledge transfer between classes under zero-shot learning settings (see our
We propose to discover visual properties through patch-level AWA2, CUB, and SUN), our learned VGSE semantic embeddings [3, 25, 33, 44, 50, 65]. Semantic embeddings collected automatically, where no additional human annotation is required except for the class labels of seen class images. Despite their importance, semantic embeddings are relatively under-explored in zero-shot learning. Human-annotated attributes [19, 36, 55, 59], i.e., the properties of objects such as color and shape, are the most commonly used semantic embeddings in zero-shot learning. Though the attributes can be discriminative for each class, their annotation process is labor-intensive and require expert knowledge [50, 55, 65]. We propose to discover visual properties through patch-level clustering over image datasets, and predict semantic embeddings automatically, where no additional human annotation is required except for the class labels of seen class images.

Semantic Embeddings with Minimal Supervision is drawing attention in image classification [6, 9, 26, 40, 45], transfer learning [10, 37, 54] and low-shot learning problems [3, 25, 33, 44, 50, 65]. Semantic embeddings collected from text corpora are alternatives to manual annotations, which include word embeddings learned from large corpora [31, 38, 49, 63], semantic relations such as knowledge graphs [9, 26, 56], and semantic similarities [12, 57], etc. More recently, [3, 39, 41, 67] collect attribute-class associations from online encyclopedia articles that describe each category. The semantic similarity can be encoded by a taxonomical hierarchy or by incorporating co-occurrence statistics of words within the document. However, this may not reflect visual similarity, e.g., sheep is semantically close to dog since they often co-occur in online articles, while visually sheep is closer to a deer. We focus on discovering visually-grounded semantic embeddings in the image space, and further incorporate the semantic relations between classes into our semantic embedding for better zero-shot knowledge transfer.

2. Related Work

Zero-shot Learning aims to classify images from novel classes that do not appear during training. Existing ZSL methods usually assume that both the seen and unseen classes share a common semantic space, thus the key insight of performing ZSL is to transfer knowledge from seen classes to unseen classes. To assign the image to a semantic class embedding, many classical approaches learn a compatibility function to associate visual and semantic space [1, 20, 48, 58]. Recent works mainly focus on synthesizing image features or classifier weights with a generative model [43, 60, 61], or training enhanced image features extractors with visual attention [66, 68] or local prototypes [62].

Semantic embeddings are crucial in relating different categories with shared characteristics, i.e., the semantic space. Despite their importance, semantic embeddings are relatively unexplored in zero-shot learning. Human-annotated attributes [19, 36, 55, 59], i.e., the properties of objects such as color and shape, are the most commonly used semantic embeddings in zero-shot learning. Though the attributes can be discriminative for each class, their annotation process is labor-intensive and require expert knowledge [50, 55, 65]. We propose to discover visual properties through patch-level clustering over image datasets, and predict semantic embeddings automatically, where no additional human annotation is required except for the class labels of seen class images.

3. Visually-Grounded Semantic Embedding

We are interested in the (generalized) zero-shot learning task where the training and test classes are disjoint sets. The training set \( \{(x_n, y_n) | x_n \in X^s, y_n \in Y^s\}_{n=1}^{N_s} \) consists of images \( x_n \) and their labels \( y_n \) from the seen classes \( Y^s \). In
the ZSL setting, test images are classified into unseen classes $Y^u$, and in the GZSL setting, both $Y^s$ and $Y^u$ with the help of a semantic embedding space, e.g., human annotated attributes. Since human-annotated attributes are costly to obtain, while prior unsupervised semantic embeddings are incomplete to describe the rich visual world, we propose to automatically discover a set of $D_v$ visual clusters as the semantic embedding, denoted by $\Phi^{VGE} \in \mathbb{R}^{(|Y^s|+|Y^u|) \times D_v}$. The semantic embeddings for seen classes $\{\phi^{VGE}(y) | y \in Y^s\}$, describing diverse visual properties of each category, are learned on seen classes images $X^s$. The semantic embeddings for unseen classes $\{\phi^{VGE}(y) | y \in Y^u\}$ is predicted with the help of unsupervised word embeddings, e.g., w2v embeddings for class names $\Phi^w \in \mathbb{R}^{(|Y^u|+|Y^s|) \times D_w}$.

Our Visually-Grounded Semantic Embedding (VGSE) Network (see Figure 2) consists of two main modules. (1) The Patch Clustering (PC) module takes the training dataset as input, and clusters the image patches into $D_v$ visual clusters. Given one input image $x_n$, PC can predict the cluster probability $a_n \in \mathbb{R}^{D_v}$, indicating how likely the image would contain the visual property appearing in each cluster. (2) Since unseen class images cannot be observed during training, we propose the Class Relation (CR) module to infer the semantic embeddings of unseen classes. Finally, the learned semantic embedding $\Phi^{VGE}$ can be used to perform downstream tasks, e.g., Zero-Shot Learning.

### 3.1. Patch Clustering (PC) Module

**Patch image generation.** Patch-level embeddings allow us to explore the visual properties that appear in local image regions [17, 55], e.g., the shape and texture of animal body parts or the objects in scenes. To obtain image patches that cover the entire semantic image region (e.g., an animal head), we segment an image into regularly shaped regions via an unsupervised compact watershed segmentation algorithm [32]. As shown in Figure 2, for each image $x_n$, we find the smallest bounding box that fully covers each segment and crop $x$ into $N_t$ patches $\{x_{nt}\}_{nt=1}^{N_t}$ that cover different parts of the image. The number of patches $N_t$ is empirically set to be around 9, as we observed in initial experiments that larger patches may include too many attributes, while smaller patches will be too tiny to contain any visual attribute. In this way, we reconstruct our training set consisting of image patches $\{(x_{nt}, y_n) | x_{nt} \in X^{sp}, y_n \in Y^s\}_{nt=1}^{N_t}$, where $|X^{sp}| = N_sN_t$, and $N_s$ is the train set size.

**Patch clustering.** Our patch clustering module is a differentiable middle layer, that simultaneously learns image patch representations and clustering. As shown in Figure 2 (left), we start from a deep neural network that extracts patch feature $\theta(x_{nt}) \in \mathbb{R}^{D_f}$, where we use a ResNet [22] pretrained on ImageNet [14] as in other ZSL models [59, 61]. Afterwards, a clustering layer $H : \mathbb{R}^{D_f} \rightarrow \mathbb{R}^{D_v}$ converts the feature representation into cluster scores:

$$a_{nt} = H \circ \theta(x_{nt}) ,$$

where $a_{nt}^k$ (the $k$-th element of $a_{nt}$) indicates the probability of assigning image patch $x_{nt}$ to cluster $k$, e.g., the patch clusters of spotty fur, fluffy head in Figure 2.

A pretext task can be adopted to obtain semantically meaningful representations [21, 34, 53] in an unsupervised manner. Our pretext task [53] enforces the image patch $x_{nt}$ and its neighbors being predicted to the same clusters. We

---

**Figure 2.** Our visually-grounded semantic embedding network consists of two modules. The Patch Clustering (PC) module learns clusters from patch images, and predicts semantic embeddings for seen classes with their images. The Class Relation (CR) module predicts the unseen class embeddings $\phi^{VGE}(y_n)$ using unseen and seen class relations learned from external knowledge, e.g., word2vec. For instance, the embedding for unseen class *sheep* is predicted using the semantic embeddings of the seen classes, e.g., *antelope, cow, deer*, and so on.
retrieve nearest patch neighbors of $x_{nt}$ as $X_{nh}$ by the $L_2$ distance of patch features $||\theta(x_{nt}) - \theta(x_i)||_2$, where $x_i \in X_{sp}$ and $x_i \neq x_{nt}$. The clustering loss is defined as:

$$L_{clus} = - \sum_{x_{nt} \in X_{sp}} \sum_{x_i \in X_{nb}} \log(a_{nt}^T a_i),$$

(2)

where $a_i = H \circ \theta(x_i)$. $L_{clus}$ imposes consistent cluster assignment for $x_{nt}$ and its neighbors. To avoid all images being assigned to the same cluster, we follow [53] to add an entropy penalty as follows:

$$L_{pel} = \sum_{k=1}^{D_v} \tilde{a}_{nt}^k \log \tilde{a}_{nt}^k, \quad \tilde{a}_{nt}^k = \frac{1}{N_s N_t} \sum_{x_i \in X_{sp}} a_{nt}^k,$$

(3)

ensuring that images are spread uniformly over all clusters.

**Class discrimination.** To impose class discrimination information into the learnt clusters, we propose to apply an cluster-to-class layer $Q : \mathbb{R}^{D_v} \rightarrow \mathbb{R}^{Y_s}$ to map the cluster prediction of each image to the class probability, i.e., $p(y|x_{nt}) = \text{softmax}(Q \circ \theta(x_{nt}))$. We train this module with the following cross-entropy loss,

$$L_{cls} = - \log \frac{\exp (p(y|x_{nt}))}{\sum_{\bar{y} \in Y_s} \exp (p(\bar{y}|x_{nt}))}.$$  

(4)

**Semantic relatedness.** We further encourage the learned visual clusters to be transferable between classes, to benefit the downstream zero-shot learning tasks. We learn clusters shared between semantically related classes, e.g., horse share more semantic information with deer than with dolphin. We implement this by mapping the learned cluster probability to the semantic space constructed by w2v embeddings $\phi^w$. The cluster-to-semantic layer $S : \mathbb{R}^{D_v} \rightarrow \mathbb{R}^{D_w}$ is trained by regressing the w2v embedding for each class,

$$L_{sem} = ||S \circ a_{nt} - \phi^w(y_{nt})||_2,$$

(5)

where $y_{nt}$ denotes the ground truth class, and $\phi^w(y_{nt}) \in \mathbb{R}^{D_w}$ represents the w2v embedding for the class $y_{nt}$.

The overall objective for training the model is as follows:

$$L = L_{clus} + \lambda L_{pel} + \beta L_{cls} + \gamma L_{sem}.$$  

(6)

**Predict seen semantic embeddings.** After we learned the visual clusters, given one input image patch $x_{nt}$, the model extracts the feature $\theta(x_{nt})$ followed by predicting the cluster probability $a_{nt} = H \circ \theta(x_{nt}) \in \mathbb{R}^{D_v}$, where each dimension indicates the likelihood that the image patch $x_{nt}$ being assigned to a certain cluster learned by this module.

The image embedding $a_{nt} \in \mathbb{R}^{D_v}$ for $x_{nt}$ is calculated by averaging the patch embedding in that image:

$$a_{nt} = \frac{1}{N_t} \sum_{i=1}^{N_t} a_{nt}^i.$$  

(7)

Similarly, we calculate the semantic embedding for $y_{nt}$ by averaging the embeddings of all images belonging to $y_{nt}$:

$$\phi^\text{VGESE}(y_{nt}) = \frac{1}{|I_i|} \sum_{j \in I_i} a_j,$$

(8)

where $I_i$ is the indexes of all images belonging to class $y_{nt}$, and $a_j$ denotes the image embedding of the $j$-th image.

**3.2. Class Relation (CR) Module**

While seen semantic embeddings can be estimated from training images using Eq. 8, how to compute the unseen semantic embeddings is not straightforward since their training images are not available. As semantically related categories share common properties, e.g., sheep and cow both live on grasslands, we propose to learn a Class Relation Module to formulate the similarity between seen classes $Y_s$ and unseen classes $Y_u$. In general, any external knowledge, e.g., word2vec [31, 38] or human-annotated attributes, can be utilized to formulate the relationship between two classes. Here we use word2vec learned from a large online corpus to minimize the human annotation effort. Below, we present two solutions to learn the class relations: (1) directly averaging the semantic embeddings from the neighbor seen classes in the word2vec spaces, (2) optimizing a similarity matrix between unseen and seen classes.

**Weighted Average (WAvg).** For unseen class $y_m$, we first retrieve several nearest class neighbours in seen classes by the similarity measured with $L_2$ distance over w2v embedding space, and we denote the neighbor classes set as $Y_{nb}^m$. The semantic embedding vector for $y_m$ is calculated as the weighted combination [5] of seen semantic embeddings:

$$\phi^\text{VGESE}(y_m) = \frac{1}{|Y_{nb}^m|} \sum_{\bar{y} \in Y_{nb}^m} \text{sim}(y_m, \bar{y}) \cdot \phi^\text{VGESE}(\bar{y}),$$  

(9)

$$\text{sim}(y_m, \bar{y}) = \exp(-\eta \|\phi^w(y_m) - \phi^w(\bar{y})\|_2),$$  

(10)

where $\exp$ stands for the exponential function and $\eta$ is a hyperparameter to adjust the similarity weight. We denote our semantic embeddings learned with weighted average strategy as VGSE-WAvg.

**Similarity Matrix Optimization (SMO).** Given the w2v embeddings $\phi^w(Y_s) \in \mathbb{R}^{Y_s \times D_w}$ of seen classes and embedding $\phi^w(y_m)$ for unseen class $y_m$, we learn a similarity mapping $r \in \mathbb{R}^{Y_s}$, where $r_i$ denotes the similarity between the unseen class $y_m$ and the $i$-th seen class. The similarity mapping is learned via the following optimization problem:

$$\min_r \|\phi^w(y_m) - r^T \phi^w(Y_s)\|_2$$

s.t. $0 < r < 1$ and $\sum_{i=1}^{Y_s} r_i = 1.$  

(11)
Here $\alpha$ is the lower bound which can be either 0 or $-1$, indicating whether we only learn positive class relations or we learn negative relations as well. We base this mapping on the assumption that semantic embeddings follow linear analogy, e.g., $\phi^w(\text{king}) - \phi^w(\text{man}) + \phi^w(\text{woman}) \approx \phi^w(\text{queen})$, which holds for w2v embeddings and our semantic embeddings $\phi^{\text{VGSE}}$. After the mapping is learned, we can predict the semantic embeddings for the unseen class $y_m$ as:

$$
\phi^{\text{VGSE}}(y_m) = r^T \phi^{\text{VGSE}}(Y_s),
$$

where the value of each discovered semantic embedding for unseen class $y_m$ is the weighted sum of all seen class semantic embeddings. We denote our semantic embeddings learned with similarity matrix optimization (SMO) as $\text{VGSE-SMO}$.

4. Experiments

After introducing the datasets and experimental settings, we demonstrate that our $\text{VGSE}$ outperforms unsupervised word embeddings over three benchmark settings and this phenomenon generalizes to five SOTA ZSL models (§4.1). With extensive ablation studies, we showcase clustering with images patches is effective for learning the semantic embeddings, and demonstrate the effectiveness of the PC module and CR module (§4.2). In the end, we present visual clusters as qualitative results (§4.3, §4.4).

Dataset. We validate our model on three ZSL benchmark datasets. AWA2 [59] is a coarse-grained dataset for animal categorization, containing 30,475 images from 50 classes, where 40 classes are seen and 10 are unseen classes. CUB [55] is a fine-grained dataset for bird classification, containing 11,788 images and 200 classes, where 150 classes are seen and 50 are unseen classes. SUN [36] is also a fine-grained dataset for scene classification, with 1,340 images coming from 717 scene classes, where 645 classes are seen and 72 are unseen classes.

Implementation details. Specifically, in the patch clustering (PC) module we learn scene-semantic embeddings with train set (seen classes) proposed by [59], the unseen-class embeddings are predicted in the class relation (CR) module without seeing unseen images. We adopt ResNet50 [22] pretrained on ImageNet1K [14] as the backbone. The cluster number $D_r$ is set as 150 for three datasets. For the Weighted Average module in Eq. 9, we set $\eta$ as 5 for all datasets, and use 5 neighbors for all datasets. For the similarity matrix optimization in Eq. 11, we set $\alpha$ as -1 for AWA2 and CUB, and as 0 for SUN. More details are in the supplementary.

Semantic embeddings for ZSL. To be fair, we compare our $\text{VGSE}$ semantic embeddings with other alternatives using the same image features and ZSL models. All the image features are extracted from ResNet101 [22] pretrained on ImageNet [14]. We follow the data split provided by [59]. The semantic embeddings are L2 normalized following [59]. All ablation studies use the $\text{SJE}$ [2, 62] as the ZSL model as it is simple to train. Besides, we verify the generalization ability of our semantic embeddings over five state-of-the-art ZSL models with their official code. The non-generative models include $\text{SJE}$ [2], APN [62], GEM–ZSL [28], learning a compatibility function between image and semantic embeddings. The generative approaches consist of $\text{CADA–VAE}$ [43] and $f$–$\text{VAEGAN–D2}$ [61], learning a generative model that synthesizes image features of unseen classes from their semantic embeddings. Note that for all ZSL models, we use the same hyperparameters as proposed in their original papers for all semantic embeddings with no hyperparameter tuning.

### 4.1. Comparing with the State-of-the-Art

We first compare our semantic embeddings $\text{VGSE-SMO}$ with the unsupervised word embeddings w2v [31] on three benchmark datasets and five ZSL models. We further compare ours with other state-of-the-art methods that learn semantic embeddings with less human annotation.

**VGSE surpasses w2v by a large margin.** The results shown in Table 1 demonstrate that our $\text{VGSE-SMO}$ semantic embeddings significantly outperform word embedding w2v on all datasets and all ZSL models. Considering the non-generative ZSL models, $\text{VGSE-SMO}$ outperforms w2v on all three datasets by a large margin. In particular, on AWA2 dataset, when coupled with $\text{GEM-ZSL}$, our $\text{VGSE-SMO}$ boosts the ZSL performance of w2v from 50.2% to 58.0%. On the fine-grained datasets CUB and SUN, $\text{VGSE-SMO}$ achieves even higher accuracy boosts. For example, when coupled with the APN model, $\text{VGSE-SMO}$ increases the ZSL accuracy of CUB from 22.7% to 28.9%, and the accuracy of SUN from 23.6% to 38.1%. These results demonstrate that our approach not only works well on generic object categories, but also has great potential to benefit the challenging fine-grained classification task. $\text{VGSE}$ improves the GZSL performance of both seen and unseen classes, yielding a much better harmonic mean (e.g., when trained with $\text{SJE}$, $\text{VGSE-SMO}$ improves over the harmonic mean of w2v by 8.0% on AWA2, 10.3% on CUB, and 7.6% on SUN). These results indicate that our $\text{VGSE}$ facilitates the model to learn a better compatibility function between image and semantic embeddings, for both seen and unseen classes.

Our $\text{VGSE}$ semantic embeddings show great potential on generative models as well. In particular, $\text{VGSE}$ coupled with $f$–$\text{VAEGAN–D2}$ surpasses all other methods by a wide margin on SUN and CUB datasets, i.e., we obtain 35.0% vs 32.2% (w2v) on CUB, and 41.1% vs 39.6% (w2v) on SUN. As our embeddings are more machine detectable than w2v, introducing visual properties to the conditional GAN will allow them to generate more discriminative image features.

**VGSE outperforms SOTA weakly supervised ZSL semantic embeddings.** We compare $\text{VGSE}$ with other works that learn ZSL semantic embeddings with less human annotation.
Table 1. Comparing our VGSE-SMO, with w2v semantic embedding over state-of-the-art ZSL models. In ZSL, we measure Top-1 accuracy (T1) on unseen classes, in GZSL on seen/unseen (s/u) classes and their harmonic mean (H). Feature Generating Methods, i.e., f-VAEGAN-d2, and CADA-VAE generating synthetic training samples, and SJE, APN, GEM-ZSL using only real image features.

<table>
<thead>
<tr>
<th>ZSL Model</th>
<th>Semantic Embeddings</th>
<th>Generalized Zero-Shot Learning</th>
<th>Zero-Shot Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AWA2</td>
<td>CUB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T1</td>
<td>T1</td>
</tr>
<tr>
<td>CADA-VAE</td>
<td>w2v [31]</td>
<td>49.0</td>
<td>22.5</td>
</tr>
<tr>
<td>f-VAEGAN-D2 [61]</td>
<td>VGSE-SMO (Ours)</td>
<td>52.7</td>
<td>24.8</td>
</tr>
<tr>
<td>SJE [2]</td>
<td>w2v [31]</td>
<td>58.4</td>
<td>32.7</td>
</tr>
<tr>
<td>GEM-ZSL</td>
<td>w2v [31]</td>
<td>61.3</td>
<td>35.0</td>
</tr>
<tr>
<td>APN [62]</td>
<td>w2v [31]</td>
<td>53.7</td>
<td>14.4</td>
</tr>
<tr>
<td></td>
<td>VGSE-SMO (Ours)</td>
<td>57.4</td>
<td>22.8</td>
</tr>
</tbody>
</table>

Table 2. Comparing with state-of-the-art methods for learning semantic embeddings with less human annotation (T: online textual articles, H: human annotation) using same image features and ZSL model (f-VAEGAN-d2 [61]).

CAAP [5] learns the unseen semantic embeddings with the help of w2v and the human annotated attributes for seen classes. Auto-Dis [3] collects attributes from online encyclopaedia articles that describe each category, and learn attribute-class association with the supervision of visual data and category label. GAZSL [67] and ZSLNS [39] learn semantic embeddings from wikipedia articles.

The results shown in Table 2 demonstrate that our VGSE embedding, using only w2v as external knowledge, surpasses all other method that uses textual articles on three datasets. In particular, our VGSE-SMO achieves an accuracy of 61.3% on AWA2, improving the closest semantic embedding w2v by 2.9%. On SUN, we also outperform the closest semantic embedding w2v by 1.5%.

4.2. Ablation study

We provide ablation studies for our PC and CR modules.

Is PC module effective? We first ask if learning semantic embeddings through clustering is effective in terms of ZSL accuracy, when compared to other alternatives. We compare our semantic embeddings against the following baselines:

- **ResNet features** are extracted by feeding image patch
- **w2v**
- **VGSE-SMO** (Ours)
- **Auto-Dis** [3] which uses textual articles and pretrained ResNet are listed below for comparison.

<table>
<thead>
<tr>
<th>Semantic Embeddings</th>
<th>Zero-shot learning</th>
<th>Zero-shot learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AWA2</td>
<td>CUB</td>
</tr>
<tr>
<td>w2v [31]</td>
<td>58.4</td>
<td>32.7</td>
</tr>
<tr>
<td>ZSLNS [39]</td>
<td>T</td>
<td>57.4</td>
</tr>
<tr>
<td>GAZSL [67]</td>
<td>T</td>
<td>-</td>
</tr>
<tr>
<td>Auto-Dis [3]</td>
<td>T</td>
<td>52.0</td>
</tr>
<tr>
<td>CAAP [5]</td>
<td>T and H</td>
<td>55.3</td>
</tr>
<tr>
<td>VGSE-SMO (Ours)</td>
<td>w2v</td>
<td>61.3 ± 0.3</td>
</tr>
</tbody>
</table>

Table 3. Ablation study over the PC module reporting ZSL T1 on AWA2, CUB, and SUN (mean accuracy and std over 5 runs). The baseline is the PC module with the cluster loss $L_{clus}$ and $L_{pel}$. Our full model VGSE-SMO is trained with two additional losses $L_{clus}$, $L_{sem}$. Two kinds of semantic embeddings learned from k-means clustering and pretrained ResNet are listed below for comparison.

$x_{nt}$ to a pretrained ResNet50. We follow Eq. 7 and Eq. 8 to predict semantic embeddings for seen classes. **K-means clustering** is an alternative for our clustering model. We cluster the patch images features $\theta(x_{nt})$ learned from our PC module into $D_v$ visual clusters. The patch embedding $a_{nt}^k$ is defined as the cosine similarity between the patch feature $\theta(x_{nt})$ and the cluster center. In both cases the unseen semantic embeddings are predicted with our SMO module.

We ablate our losses and compare our VGSE-SMO with the two alternatives, then report ZSL results on three benchmark datasets in Table 3. First, the k-means-SMO achieves on par results with our baseline model trained with only the cluster losses $L_{clus}$ and $L_{pel}$ [53], the reason we adopt [53] instead of k-means is that we can easily train the network with our proposed losses in an end-to-end manner. Second, the addition of the classification loss $L_{clus}$ leads to notable improvement over the baseline model trained with $L_{clus}$ and $L_{pel}$, and the semantic relatedness loss $L_{sem}$ further improve the performance of our semantic embeddings, e.g., in total, we gain
5.8%, 9.4% and 9.5% improvement on AWA2, CUB, and SUN, respectively. The result demonstrates that imposing class discrimination and semantic relatedness leads to better performance in the ZSL setting. Third, our VGSE-SMO embeddings improve over the ResNet-SMO embeddings by 71.1%, 10.7% and 10.7% on AWA2, CUB, and SUN, respectively. We conjecture that the visual clusters learned in our model is shared among different classes and lead to better generalization ability when the training and testing sets are disjoint (see qualitative results in Figure 1 and Section 4.3).

**How many clusters are needed?** To measure the influence of the cluster number \(D_v\) on our semantic embeddings, we train the PC module with various \(D_v\) (results shown in Figure 3a). When the unseen semantic embeddings are predicted under an oracle setting (predicted from the unseen class images), various dimension \(D_v\) does not influence the classification accuracy on unseen classes (the orange curve). While under the ZSL setting where unseen semantic embeddings are predicted from class relations (VGSE-SMO), the cluster numbers influence the ZSL performance. Before the cluster number increases up to a breaking point (\(D_v = 200\)), the ability of the semantic embeddings is also improved (from 58.4% to 62.5%), since the learned clusters contain visually similar patches from different classes, which can model the visual relation between classes. However, increasing the number of clusters leads to small pure clusters (patches coming from only a single category), resulting in poor generalization between seen and unseen classes.

**SMO vs WAvg.** We compare our two class relation functions VGSE-WAvg and VGSE-SMO in Table 3 (Row 7 and 6). The results demonstrate that VGSE-WAvg works on par with VGSE-SMO on SUN and CUB datasets, with < 0.5% performance gap. While on AWA2 dataset, VGSE-SMO yields better ZSL performance (with 62.4%) than VGSE-WAvg (with 57.7%). The results indicate that predicting the unseen semantic embeddings with the weighted average of a few seen classes semantic embeddings (VGSE-WAvg) is working well for fine-grained datasets since the visual discrepancy between classes is small. However, for coarse-grained dataset AWA2, the class relation function considering all the seen classes embeddings (VGSE-SMO) works better.

**Ablation over patches.** We further study if using patches for clustering is better than using the whole image, and how many patches do we need from one image. The experiment results in Figure 3b demonstrate that with the patch number increase from 1 (single image clustering) to 9, the ZSL performance increases as well, since the image patches used for semantic embedding learning contain semantic object parts and thus result in better knowledge transfer between seen and unseen classes. However, for a large \(N_t\), the patches might be too tiny to contain consistent semantic, thus resulting in performance dropping, e.g., the ZSL accuracy on AWA2 drops from 62.4% \((N_t = 9)\) to 58.7% \((N_t = 128)\). We also compare the patches generated by watershed segmentation proposal with using 3x3 grid patches \((N_t = 9)\), and we found that using watershed as the region proposal results in accuracy boost (8.2% on AWA2) compared to the regular grid patch, since the former patches tend to cover more complete object parts rather than random cropped regions.

**Can we do better with human annotated attributes?** Table 4 shows the performance of our model when different external knowledge is used to predict the unseen class embeddings in the CR module. Nearly all of our conclusions from former section carry over, e.g., VGSE-SMO class embeddings outperform the other class embeddings by a large margin. For instance, we improve the ZSL accuracy over glove by 7.7% (AWA2) and 5.9% (CUB). Furthermore, VGSE-SMO (Attribute) also outperform Attribute on both AWA2 and CUB dataset, i.e., we achieve 66.7% (ZSL) on AWA2, compared to human attributes with 62.8%. The results demonstrate that our

---

**Table 4. Evaluating the external knowledge, i.e., word embeddings w2v [31], glove [38], fasttext [7], and the human annotated attributes, for our VGSE-SMO embeddings, e.g., VGSE-SMO (glove) indicates that CR module is trained with glove embedding. T1: top-1 accuracy in ZSL, H: harmonic mean in GZSL trained with SJE [2] on AWA2, and CUB. (std over 5 runs).**

<table>
<thead>
<tr>
<th>Semantic Embeddings</th>
<th>AWA2</th>
<th>CUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>w2v [31]</td>
<td>53.7 ± 0.2</td>
<td>88.8 ± 0.1</td>
</tr>
<tr>
<td>VGSE-SMO (w2v)</td>
<td>62.4 ± 0.1</td>
<td>56.8 ± 0.1</td>
</tr>
<tr>
<td>glove [38]</td>
<td>38.8 ± 0.2</td>
<td>38.7 ± 0.3</td>
</tr>
<tr>
<td>VGSE-SMO (glove)</td>
<td>46.5 ± 0.1</td>
<td>46.0 ± 0.1</td>
</tr>
<tr>
<td>fasttext [7]</td>
<td>47.7 ± 0.1</td>
<td>44.6 ± 0.3</td>
</tr>
<tr>
<td>VGSE-SMO (fasttext)</td>
<td>51.9 ± 0.2</td>
<td>53.2 ± 0.1</td>
</tr>
<tr>
<td>Attribute</td>
<td>62.8 ± 0.1</td>
<td>62.6 ± 0.3</td>
</tr>
<tr>
<td>VGSE-SMO (Attribute)</td>
<td>66.7 ± 0.1</td>
<td>64.9 ± 0.1</td>
</tr>
</tbody>
</table>

---

**Figure 3.** (a) Influence of the cluster number \(D_v = 50, \ldots, 3000\). In the oracle setting, we feed unseen classes images to the PC module to predict unseen semantic embeddings. (b) Influence of the patch number \(N_t\) we used per image with the watershed segmentation for obtaining our VGSE-SMO class embeddings. \(N_t = 1\) uses the whole image (no patches). “3x3 grid” crops the image into 9 square patches. Both plots report ZSL accuracy with SJE model trained on AWA2 dataset (mean and std over 5 runs).
VGSE-SMO embeddings coupled with visually-grounded information can not only outperform the unsupervised word embeddings, but also improve over human attributes in transferring knowledge under the zero-shot setting.

4.3. Qualitative Results

In Figure 4, we show the 2D visualization of image patches in the AWA2, where 10,000 image patches are presented by projecting their embeddings $a_{nl}$ onto two dimensions with t-SNE [52]. To picture their distribution on the embedding space, we sample several visual clusters (dots marked in the same color) and the image patches from the cluster center of both seen and unseen categories. Note that the unseen patches are not used to predict the unseen semantic embeddings, but only used for visualization here.

We observe that samples in the same cluster tend to gather together, indicating that the embeddings provide discriminative information. Besides, images patches in one cluster do convey consistent visual properties, though coming from disjoint categories. For instance, the white fur appears on rabbit, polar bear, and fox are clustered into one group, and the striped fur from tiger, zebra, and bobcat gather together because of their similar texture. We further observe that nearly all clusters consist images from more than one categories. For instance, the horns from seen classes ox, deer, rhinoceros, and unseen class sheep, that with slightly different shape but same semantic, are clustered together. Similar phenomenon can be observed on the spotted fur and animals in ocean clusters. It indicates that the clusters we learned contain semantic properties shared across seen classes, and can be transferred to unseen classes. Another interesting observation is that our VGSE clusters discover visual properties that my be neglected by human-annotated attributes, e.g., the cage appear for hamsters and rat, and the black and white fur not only appear on gaint panda but also on sheeps.

4.4. Human Evaluation

To evaluate if our VGSE conveys consistent visual and semantic properties, we randomly pick 50 clusters, each equipped with 30 images from the cluster center, and ask 5 postgraduate students without prior knowledge of ZSL to examine the clusters and answer the following three questions. Q1: Do images in this cluster contain consistent visual property? Q2: Do images in this cluster convey consistent semantic information? Q3: Please name the semantics you observed from the clusters, if your answer to Q2 is true. We do the same user study to 50 randomly picked clusters from the k-means clustering model. The results reveal that in 88.5% and 87.0% cases, users think our clusters convey consistent visual and semantic information. While for k-means clusters, the results are 71.5% and 71.0%, respectively. The user evaluation results agree with the quantitative results in Table 3, which demonstrates that the class embeddings containing consistent visual and semantic information can significantly benefit the ZSL performance. Interestingly, by viewing VGSE clusters, users can easily discover semantics and even fine-grained attributes not depicted by human-annotated attributes, i.e., the fangs and horns in figure 1. Note that the whole process, i.e., naming 50 attributes for 40 classes, took less than 1 hour for each user.

5. Conclusion

We develop a Visually-Grounded Semantic Embedding Network (VGSE) to learn distinguishing semantic embeddings for zero-shot learning with minimal human supervision. By clustering image patches with respect to their visual similarity, our network explores various semantic clusters shared between classes. Experiments on three benchmark datasets demonstrate that our semantic embeddings predicted from the class-relation module are generalizable to unseen classes, i.e., achieving significant improvement compared with word embeddings when trained with five models in both ZSL and GZSL settings. We further show that the visually augmented semantic embedding outperforms other semantic embeddings learned with minimal human supervision. The qualitative results verify that we discover visually consistent clusters that generalize from seen to unseen classes and can unearth the fine-grained properties not depicted by humans.

Acknowledgements

This work has been partially funded by the ERC 853489 - DEXIM and by the DFG – EXC number 2064/1 – Project number 390727645.
References


[30] 2015. 1


