Generalized Zero-Shot Recognition based on Visually Semantic Embedding

Pengkai Zhu   Hanxiao Wang   Venkatesh Saligrama
Electrical and Computer Engineering Department, Boston University

Abstract

We propose a novel Generalized Zero-Shot learning (GZSL) method that is agnostic to both unseen images and unseen semantic vectors during training. Prior works in this context propose to map high-dimensional visual features to the semantic domain, which we believe contributes to the semantic gap. To bridge the gap, we propose a novel low-dimensional embedding of visual instances that is "visually semantic." Analogous to semantic data that quantifies the existence of an attribute in the presented instance, components of our visual embedding quantifies existence of a prototypical part-type in the presented instance. In parallel, as a thought experiment, we quantify the impact of noisy semantic data by utilizing a novel visual oracle to visually supervise a learner. These factors, namely semantic noise, visual-semantic gap and label noise lead us to propose a new graphical model for inference with pairwise interactions between label, semantic data, and inputs. We tabulate results on a number of benchmark datasets demonstrating significant improvement in accuracy over state-of-art under both semantic and visual supervision.

1. Introduction

Zero-shot learning (ZSL) is emerging as an important tool for large-scale classification [20], where one must account for challenges posed by non-uniform and sparse annotated classes [5], the prohibitive expense in labeling large fractions of data [4], and the need to account for appearance of novel objects for in-the-wild scenarios.

ZSL proposes to learn a model for classifying images for "unseen" classes for which no training data is available by leveraging semantic features, which are shared by both seen and unseen classes. A few recent works [7, 24] point out that, unseen image class recognition, while important, overlooks real-world scenarios, where both seen and unseen instances appear. Consequently, generalized zero-shot learning (GZSL) methods capable of recognizing both seen and unseen instances at test time are required.

We propose to train a GZSL method, that takes labeled seen class images and associated semantic side information as input, while being agnostic to both unseen images and unseen associated semantic vectors.

Challenges. We list challenges in this context:

Visual → Semantic Gap. Visual feature representations such as the final-layer outputs of deep neural networks are high-dimensional and not semantically meaningful. This limits the learner in identifying robust associations between visual patterns and semantic data.

Semantic→Visual Gap. A fundamental drawback of semantic data is that they are often not visually meaningful and it is difficult for a learner to identify and suppress non-visual semantic components during training.Additionally, semantic information provided for some classes (ex. sofa-chair), are nearly identical. This is challenging in a GZSL setting particularly when one of such classes is among the unseen.

Novelty. At a conceptual level visual representation and supervision fundamentally impacts accuracy. We re-examine these concepts and propose novel methods in Sec. 3 to identify and bridge the visual-semantic gap.

Visually Semantic Embedding. By a visually semantic embedding, we mean a mapping of visual instances to a representation that mirrors how semantic data is presented for an instance. In Sec. 3.1 we propose to train a model that learns a finite list of parts based on a multi-attention model and expresses the input as a finite probabilistic mixture of part-types, which we then output as our representation. Our intuition is informed by semantic data where for each instance, an annotator could score existence of attributes from a common vocabulary. Analogously, our embedding scores existence of proto-typical part types in a presented instance.

3-Node Graphical Model. A key aspect of our setup, which is presented in Sec. 3.1, is a graphical model[13] that has semantic (S), input (X) and label (Y) variables in a 3-node clique. This is based on the key insight that the labels are not fully explained by either the input or the semantic instance and thus we require a model that accounts for 3-way connection $(S \leftrightarrow X, X \leftrightarrow Y, Y \leftrightarrow S)$. This is a significant departure from existing works [15], where the semantic signal is given paramount importance and the structure is a chain $X \leftrightarrow S \leftrightarrow Y$.

Following convention, we conduct experiments with semantic supervision on benchmark datasets in Sec. 4. While we demonstrate significant improvement over state-of-the-art, we are driven to understand and quantify how semantic

noise can explain GZSL performance loss.

Visually Semantic Supervision. As a thought experiment we propose to train a novel visual oracle for GZSL supervision. Our intuition is that a visual oracle can reduce semantic noise and provide more definitive feedback about the presence/absence of prototypes. To ensure fair comparison between semantic and visual feedback, we first learn a common vocabulary of protypical parts and part-types unbeknownst to the learner. Like a semantic signal, our visual oracle for each input instance provides the learner only a list of similarity scores, with no other additional description of what the components in the list mean.

Visual supervision naturally leads us to propose visual evaluation, which involves evaluating predicted visually semantic outputs against the ground-truth. In Sec. 4, we show that GZSL performance improves not only for our method but also for a well-known baseline method [10] when we substitute visual in place of semantic supervision.

2. Related Work

Zero-shot learning as a topic has evolved rapidly over the last decade and documenting the extensive literature here is not possible. As summarized in [24] many existing methods can be grouped into attribute methods as exemplified in [15] that leverage attributes as an intermediate feature space to link different classes, embedding methods [10] that directly map visual domain to semantic space, and hybrid methods [28], that map semantic and visual domain into a shared feature space. Recent work [7, 24] introduces GZSL problem and developed calibration and evaluation protocols showing significant drop in accuracy between ZSL and GZSL. Many recent works are beginning to focus attention on the GZSL setup.

In this context, we propose an embedding based method and describe closely related setups and concepts that have appeared in the literature. We first categorize existing work based on problem setup and different types of side information utilized during training. There are a number of recent works that propose approaches for both ZSL and GZSL cases [17, 3, 30, 14, 25, 11, 8, 23, 16]. Among these, there are works that leverage some form of unseen class information during training [30, 14, 25, 11] to synthesize unseen examples by means of GAN or VAE training. Other works employ knowledge graphs [16, 23] incorporating both seen and unseen classes during training to infer classifiers for unseen classes. Still others are transductive, namely, at test-time [17] they leverage a batch of test examples to further refine their model. While these proposed approaches that leverage unseen class information are interesting, we take the view that for applications involving recognition in-the-wild scenarios, novel classes may only appear at test-time, and it is important to consider such situations. Ultimately, as a subject of future work, it would be interesting to incorporate cases where some unseen class information is known during training, while leaving open the possibility of existence of novel classes at test-time.

Like us, there are works [8, 3] focusing on ZSL and GZSL, while being agnostic to any unseen class information during training. In [3], authors propose an encoder-decoder network with the goal of mirroring learnt semantic relations between different classes in the visual domain. While the goal is similar, our approach is significantly different. We propose to mirror information provided by semantic attributes visually by means of a low-dimensional statistical embedding. Our embedding scores existence of prototypical part types, where the prototypical part types are learnt from training data.

In [8], the authors propose an approach that extends methods of [12]. Their idea is to penalize approximation error in reconstructing visual domain features from the semantic domain, in addition to penalizing classification loss. Their claim is that by doing so they can overcome semantic information loss suffered in methods that are based on visual to semantic embedding and prevent situations where attributes possibly corresponding to unseen examples maybe lost during training. In contrast, our claim is precisely that many semantic attributes are not visual and conventional visual features are not represented in the presented semantic vectors. Consequently, we propose approaches that on the one hand produces semantically closer visual representations through low-dimensional graphical models, and on the other hand suppress semantic components that are visually unrepresentative by means of discriminative loss functions. In general we do not require high-dimensional estimation, which these methods require.

In this context our approach bears some similarities to [18] and [30]. In particular, [18] propose zoom-net as a means to filter-out redundant visual features such as deleting background and focus attention on important locations of an object. [30] further extend this insight and propose visual part detector (VPDE-Net) and utilize high-dimensional part feature vectors as an input for semantic transfer. [30]'s proposal is to incorporate the resulting reduced representation as a means to synthesize unseen examples leveraging knowledge of unseen class attributes. Different from these works we develop methods to learn a statistical representation of mixture proportions of latent parts. Apart from being low-dimensional the mixture proportions intuitively capture underlying similarity of a part-type to other such part-types found in other classes. The focus of semantic mapping is then to transfer knowledge between mixture proportion of part types and semantic similarity.

3. Proposed Approach

3.1. A Probabilistic Perspective of GZSL

Let us motivate our approach from a probabilistic modeling perspective. This will in turn provide a basis for our
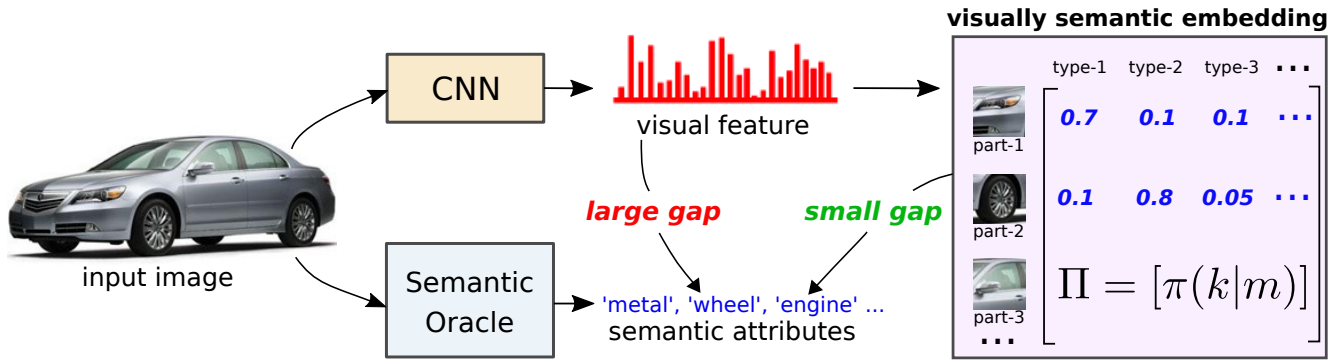
visually semantic embedding

Figure 1. Many existing works attempt to transfer high-dimensional visual features into semantic domain leading to significant visual-semantic gap. To bridge the gap we propose a new latent visual embedding that is *visually semantic*. As illustrated our new representation is low dimensional and its components are likelihoods of a part-type relative to proto-typical part-types found across all instances. We posit that our embedding mirrors how semantic components score similarity of an attribute found in an instance in the visual domain.
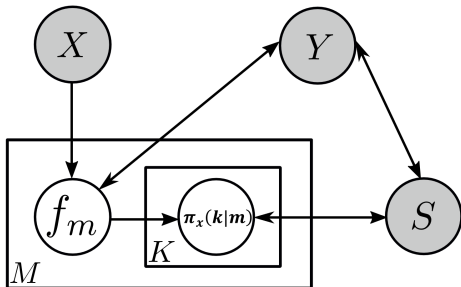

Figure 2. Graphical Model of label, semantic signal, and input as a cycle. Input is transformed into a finite collection of feature vector parts indexed by items in a part-list. Feature parts are mapped into structured probability space, with component $\pi_x(k|m)$ denoting the probability of k-th type in item m in the part-list. $\Pi_k(x) = [\pi_x(k|m)]$ denoting visual embedding for part k.

discriminative learning method. $[N]$ denotes integers from 1 to N. Overloading notation we denote $[\eta_{k,m}]$ to mean the matrix as k, m range over their values. Following convention we denote random variables with upper case letters and a realization by lower case letters. Let $x \in \mathcal{X}$ be inputs taking values in an arbitrary feature space and $\mathcal{Y}$ the space of objects or classes. The set $\mathcal{Y}$ is partitioned into $o \in \mathcal{O}$ and $u \in \mathcal{U}$ denoting the collection of observed class labels and unobserved labels respectively. Associated with each observed and unobserved class labels are semantic signals, $s_o, s_u \in \mathcal{S}$, taking values in a general space respectively. We denote by p the joint density or marginal densities wherever appropriate.

Given training data $(x_1, y_1, s_{y_1}), \ldots, (x_n, y_n, s_{y_n}) \subset \mathcal{X} \times \mathcal{O} \times \mathcal{S}$, the task of GZSL is to accurately predict a label with input drawn from $x \overset{d}{\sim} p_X(\cdot)$. If we had knowledge of joint probability density, the optimal predictor is the MAP estimate, $y_{\text{MAP}}(x) = \arg\max_y \max_s \log(p(y, s \mid x))$. Existing work [15] posits instead a chain $X - S - Y$, namely, conditioned on semantic signals, the input and class labels are independent or that the chain $X - Y - S$ [27]

is true. Nevertheless, we take the view that, *since semantic information is not fully visual, visual features are not fully semantic, and labels are not fully captured by either visual or semantic signals*, the three random variables form a cycle in a graphical model. By means of potential functions for graphical models we decompose $\log(p(y, s|x) \propto \phi_{YX}(y, x) + \phi_{XS}(x, s)) + \phi_{SY}(s, y)$.

Latent Graphical Model. Fig. 2 presents a detailed framework of our model. The input $X = x$ is mapped into a finite set of feature vector parts, $f_m(x) \in \mathbb{R}^C$ indexed by discrete part-list, $m \in [M]$. Feature parts are derived from a multi-attention model with different items $m, m' \in [M], m \neq m'$ focusing on different regions in the image. We then model each feature part vector as a C-dimensional Gaussian Mixture Model, $f_m(x) \approx \sum_i \pi_x(k|m)\mathcal{N}(\theta_{k,m}, \gamma^2 I)$, with isotropic components. Note that the parameters, $\theta_{k,m} \in \mathbb{R}^C$, are part and type dependent but shared among all instances. We refer to $\theta_{k,m}$'s as *prototypical part-types*. We collect the parameters into a matrix $\Theta_m = [\theta_{k,m}] \in \mathbb{R}^{C \times K}$.

Each mixture component, $\pi_x(k|m)$ represents the probability of type k conditioned on part m. In this way, the input x is embedded into a collection of mixture components $\Pi_m(x) = [\pi_x(k|m)]$ and $\Pi(x) = [\Pi_m(x)]$. We decompose the likelihood as:

$$\log(p(y, s \mid x)) \propto \phi_{SX}(s, \Pi(x)) + \phi_{XY}([f_m(x)], y) \quad (1)$$
$$+ \phi_{YS}(y, s) - \sum_{m=1}^{M} L_{mix}(\Theta_m, \Pi_m(x), f_m(x))$$
$$- L_{prt}([f_m(x)])$$

where, the last two terms respectively model mixture likelihood and enforce diversity of multi-attention of parts. The goal of training algorithm is to estimate the potential functions, ϕ, the feature part backbone, $f_m(\cdot)$, and the probability maps, $\Pi_m(\cdot)$ by leveraging training data.

Latent Visual Embedding Intuition. Note that, by design

(see Fig. 2 and Eq. 1), the semantic random vector interacts with the input only through the mixture component $\Pi(x)$. This is in contrast to existing works where the interaction is high-dimensional. Our intuition is that, just like a component of a semantic vector quantifies existence of an attribute, in an analogous fashion, $\pi_x(k|m)$ quantifies existence of part-type k in part m. While prototypical parts such as $\theta_{k,m}$ are high-dimensional, the corresponding mixture component $\pi_x(k|m)$ is a scalar number. In this way we propose to reduce the semantic-visual gap by removing irrelevant visual features that are not transferable.

Visual Oracle Supervision (VOS). We consider visual oracles capable of providing feedback for learner predicted visual embedding. We denote $\Pi_{vo}(x)$ as oracle feedback. As to how to build a visual oracle will be discussed later. We can consider structured and class-averaged VOS. In the structured version, for each instance, x, VOS reveals the probabilistic embedding $\Pi_{vo}(x)$; and in the class-averaged case, only reveals $\bar{\Pi}_{vo} = \mathbb{E}_{X|Y}[\Pi_{vo}(x) \mid y]$. The main difference in Eq. 1 is that we substitute oracle parameters for the semantic signal, i.e., s with $\Pi_{vo}(x)$, $\bar{\Pi}_{vo}$ etc.

Justification of Visual Oracle Supervision. In constructing the visual oracle, our goal is driven by the need to quantify semantic noise., To do so we need an oracle that provides no more information than a "noiseless" semantic one. This is not that hard since our visual oracle presents the learner with mixture values $\Pi_{vo}(x)$ with out identifying what these numbers mean or which classes, parts or locations they refer to. All that a learner knows is that what the oracle is communicating information that has definitive visual meaning.

Visual and Semantic Test-Time Evaluation. At test-time, following convention, for the semantic setting, we assume that the codebook consisting of seen and unseen semantic attribute vectors, $\{s_y | y \in \mathcal{Y}\} = \{s_o | o \in \mathcal{O}\} \cup \{s_u | u \in \mathcal{U}\}$ are revealed to the learner. For a test image, x, the learner must identify the hidden label. To do this, the learner computes $\Pi(x)$, and estimates the label by maximizing the visual-semantic potential $\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} \phi_{SX}(s_y, \Pi(x))$. In the visual evaluation setting, the class-level $\{\bar{\Pi}_{vo}^y | y \in \mathcal{Y}\}$ is revealed during test time and the learner make prediction by maximizing the visual-semantic potential in the Π space $\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} \phi_{SX}(\bar{\Pi}_{vo}^y, \Pi(x))$.

3.2. Model and Loss Parameterization

Part Feature model $f_m(\cdot)$: Inspired by [29], we use a multi-attention convolutional neural network (MA-CNN) to map input images into a finite set of feature vector parts, $f_m(x)$. Specifically, $f_m(x) = [f_{m,c}]$ contains a feature extractor E and a channel grouping model G, where $E(x) \in \mathbb{R}^{W \times H \times C}$ is a global feature map, and $G(E(x)) \in \mathbb{R}^{M \times C}$ is a channel grouping weight matrix. We then calculate an

attention map $A_m(x) \in \mathbb{R}^{W \times H}$ for the m-th part:

$$A_m(x) = \text{sigmoid}\Big(\sum_c G_{m,c}(x) \times E_c(x)\Big) \quad (2)$$

The part feature $f_m(x) \in \mathbb{R}^C$ is then calculated as:

$$f_{m,c}(x) = \sum_{w,h} [A_m(x) \odot E_c(x)]_{(w,h)}, \quad \forall c \in [C] \quad (3)$$

where $\odot$ is the element-wise multiplication. We parameterized $E(\cdot)$ by the ResNet-34 backbone (to $conv5\_x$), and $G(\cdot)$ by a fully-connected layer.

Mixture model $\Pi(\cdot)$: Note that our Gaussian mixture model implies:

$$\mathbb{E}_Z[f_m(x)|\Pi_m, \Theta_m] = \Theta_m \Pi_m(x)$$

This sets up a matrix factorization problem, with positivity constraints on the components of $\Pi_m(x)$. Observe that, at test-time, since the matrices Θ_m are known, the solution to $\Pi(x)$ reduces to solving a linear system of equations with positivity constraints. Alternatively, we can employ a Bayesian perspective (which is what we do) and compute:

$$\pi_x(k|m) \propto \widehat{\pi(k|m)} \mathcal{N}(f_m(x); \theta_{k,m}, \gamma^2 I) \quad (4)$$

where, $\widehat{\pi(k|m)}$ is the prior for prototype k in part m estimated during training.

Part Feature Learning Loss L_{prt}: To encourage a part-based representation $f_m(x)$ to be learned, we follow [29]. Since $f_m(x)$ can be decomposed into $A_m(x) \odot E(x)$, we want to force the learned attention maps A_m to be both compact within the same part, and divergent among different parts. We define $L_{prt}([f_m(x)])$ to be:

$$L_{prt}([f_m(x)]) = \sum_m (L_{dis}(A_m(x)) + \lambda L_{div}(A_m(x))) \quad (5)$$

where the compact loss $L_{dis}(A_m)$ and divergent loss $L_{div}(A_m)$ are defined as (x is dropped for simplicity):

$$L_{dis}(A_m) = \sum_{w,h} A_m^{w,h}[\|w - w^*\|^2 + \|h - h^*\|^2] \quad (6)$$

$$L_{div}(A_m) = \sum_{w,h} A_m^{w,h}[max_{n,n\neq m} A_n^{w,h} - \zeta] \quad (7)$$

where $A_m^{w,h}$ is the amplitude of A_m at coordinate (w, h), and (w^*, h^*) is the coordinate of the peak value of A_m, ζ is a small margin to ensure the training robustness.

Mixture Model Learning Loss L_{mix}: We pose this as a standard max-likelihood estimation problem, and learn $\theta_{k,m}, \gamma$ parameters using the EM algorithm to fit the feature vectors $f_{m,i} \triangleq f_m(x_i)$, with x_i being training examples. We can write the negative log-likelihood for i-th sample:

$$L_{mix}(\Theta_m, \Pi_m, f_{m,i}) = -\log(\sum_k \bar{\pi}(k|m) \mathcal{N}(f_{m,i}; \Theta_m, \gamma^2 I)) \quad (8)$$

where the parameters are optimized by the Expectation-Maximization (EM) algorithm during training. Once they are learned, the mixture component embedding $\Pi(x)$ can be inferred with Eq.(4).

Semantic-Label Potential ϕ_{YS}: In the GZSL problem, we usually assume a deterministic one-to-one mapping between the semantic signals to class labels provided by a semantic oracle (human annotator). The Semantic-Label potential function is thus simply modeled by an indicator function:

$$\phi_{YS}(y, s_{y'}) = \mathbb{I}(y = y') \tag{9}$$

Visual-Label Potential ϕ_{XY}: To map visual representations to class labels, we construct a classification model D that takes the concatenated part features $[f_m(x)]$ as input and outputs a classification prediction, i.e. $D([f_m(x)]) \in \mathbb{R}^{|\mathcal{O}|}$, where $|\mathcal{O}|$ refers to the number of observed classes. In our implementation, $D(\cdot)$ is simply a fully-connected layer followed by a softmax. Let $\hat{p}(y)$ denote the one-hot encoding of the ground-truth class label y for input image x, the potential ϕ_{XY} is given by the negative cross-entropy between label and prediction:

$$\phi_{XY}([fm(x)], y) = -\text{CE}(D([f_m(x)]), \hat{p}(y)) \tag{10}$$

Visual-Semantic Potential ϕ_{SX}: Existing works often take the raw feature vectors as the visual representation, i.e. $E(x)$ or $f_m(x)$, and suffer from a large discrepancy between the visual and semantic domains. To mitigate such a gap, we propose to adopt the latent mixture component embedding $\Pi(x)$.

Semantic Oracle. In the common GZSL setting where the semantic signals are obtained from a human annotator, we construct a Semantic Mapping model $V(\Pi(x))$ to project $\Pi(x)$ into $\mathcal{S}$, where $V(\cdot)$ is further parameterized by a neural network. Given an imput image x and its semantic attribute s_y, the potential ϕ_{SX} is modeled as:

$$\phi_{SX}(s_y, \Pi(x)) = -\sum_{y' \in \mathcal{O}}[\eta \mathbb{I}(y' = y) + s_{y'}^\top V(\Pi(x)) - s_y^\top V(\Pi(x))]_+ \tag{11}$$

where η is a margin parameter.

Visual Oracle. As discussed by Sec.3.1, we want to evaluate the efficacy of the proposed latent embedding. We thus considers a visual oracle which directly provides $\Pi_{vo}(x)$ as a visually semantic supervision, which is a list of visual part similarity scores. In this case, we no longer need $V(\cdot)$ since both $\Pi(x)$ and $\Pi_{vo}(x)$ are already in the same space. The potential ϕ_{SX} is thus:

$$\phi_{SX}(\Pi_{vo}(x), \Pi(x)) = -|\Pi_{vo}(x) - \Pi(x)|_F^2, \tag{12}$$

where $|\cdot|_F$ is the Frobenius norm.

3.3. Implementation Details

Our model takes input image size as $[448 \times 448]$ and and the output of $E(x)$ is in the size of $14 \times 14 \times 512$. There are 4 parts in the model and in each part, the number of types M is set to 16. λ in Eq.(5) and ζ in Eq.(7) is empirically set to 5 and 0.02. In the semantic-oracle scenario, the smenatic mapping model $V(\cdot)$ is implemented by a two fc-layer neural network with ReLU activation.

In the visual-oracle scenario, the visual oracle is built to provide $\Pi_{vo}(x)$. It consists of the part feature model $f_m(\cdot)$ and a classifier D, where the feature extractor $E(\cdot)$ in $f_m(\cdot)$ is parameterized by the VGG-19 convolutional layers. We choose VGG instead of ResNet backbone to avoid that the learner learns to recover the same parameters in the oracle via the $\Pi_{vo}(x)$. The oracle is first trained by maximizing $\phi_{XY} - L_{prt}([f_m(x)])$ to learn a discriminating $f_m(\cdot)$. Then the EM optimization over $\sum_{m=1}^M L_{mix}(\Theta_m \Pi_m, f_m(x))$ is done to generate $\Pi_{ov}(x)$ for our model. During training, the oracle provide instance level $\Pi_{vo}(x)$ and in the test time, only class-averaged $\bar{\bar{\Pi}}_{vo}$ is revealed to the learner to make the prediction.

4. Experiments

Datasets. We evaluate the performance of our model on three commonly used benchmark datasets for GZSL: *Caltech-UCSD Birds-200-2011* (CUB) [22], *Animals with Attributes 2* (AWA2) [24] and *Attribute Pascal and Yahoo* (aPY) [9]. CUB is a fine-grained dataset which contains 200 different types of birds. CUB has 11,788 images and 312-dim annotated semantic attributes. AWA2 is a coarse-grained dataset which has 37,322 images from 50 different animals. 85 binary and continuous class attributes are provided. aPY is also a coarse-grained dataset with 64 semantic attributes. It has 15,339 images of 20 Pascal classes and 12 Yahoo classes. We did not choose SUN [26] dataset for the reason that the scene images in SUN cannot be easily decomposed into visual parts which are compact and consistent across different scenes, and consequently not expected to benefit from our formulation. The statistics of the datasets are summarized in Table 2.

Setting. We evaluate performance for both GZSL and ZSL settings. Following the protocol in [24], in the GZSL setting, the average-class Top-1 accuracy on unseen classes (ts), seen classes (tr) and the harmonic mean (H) of ts and tr are evaluated; In the ZSL setting, we report the average-class Top-1 accuracy on both Standard Split (SS) and Proposed Split (PS).

We examine impact of different concepts such as visual representation, semantic vs. visual supervision on GZSL performance. As summarized by Table 3, two variants of the proposed model are evaluated: (1) **Ours(S)**: We benchmark performance of proposed latent visual embedding in the conventional GZSL setting. That is, we train a map-

Methods	CUB			AWA2			aPY		
	ts	tr	H	ts	tr	H	ts	tr	H
SJE[2]	23.5	59.2	33.6	8.0	73.9	14.4	3.7	55.7	6.9
SAE[12]	7.8	54.0	13.6	1.1	82.2	2.2	0.4	80.9	0.9
SSE[28]	8.5	46.9	14.4	8.1	82.5	14.8	0.2	78.9	0.4
GFZSL[21]	0.0	45.7	0.0	2.5	80.1	4.8	0.0	**83.3**	0.0
CONSE[19]	1.6	**72.2**	3.1	0.5	90.6	1.0	0.0	**91.2**	0.0
ALE[1]	23.7	62.8	34.4	14.0	81.8	23.9	4.6	73.7	8.7
SYNC[6]	11.5	70.9	19.8	10.0	90.5	18.0	7.4	66.3	13.3
DEVISE[10]	23.8	53.0	32.8	17.1	74.7	27.8	4.9	76.9	9.2
PSRZSL[3]	24.6	54.3	33.9	20.7	73.8	32.3	13.5	51.4	21.4
SP-AEN[8]	**34.7**	70.6	46.6	23.3	**90.9**	37.1	13.7	63.4	22.6
Ours(S)	33.4	**87.5**	**48.4**	**41.6**	**91.3**	**57.2**	**24.5**	72.0	**36.6**
Ours(Π)	**39.5**	68.9	**50.2**	**45.6**	88.7	**60.2**	**43.6**	78.7	**56.2**

Table 1. gZSL learning results on CUB, AWA2 and aPY. ts = test classes (unseen classes), tr = train classes (seen classes), H = harmonical mean. The accuracy is class-average Top-1 in %. The highest accuracy is in red color and the second is in blue (better viewed in color).

Dataset	Num Att	$\mathcal{Y}$	$\mathcal{O}$	$\mathcal{U}$	Image
CUB[22]	312	200	150	50	11788
AWA2[24]	85	50	40	10	37322
aPY[9]	64	32	20	12	15339

Table 2. Statistics for CUB[22], AWA2[24] and aPY[9]. Number of semantic attributes, number of class for all($\mathcal{Y}$), seen($\mathcal{O}$) and unseen($\mathcal{U}$), and the number of images are listed.

Method	Supervision	Evaluation	Representation
Others	S_O	$\mathcal{S}$	$\mathcal{F}$
Ours(S)	S_O	$\mathcal{S}$	Π
Ours(Π)	Π_{vo}	Π	Π

Table 3. Comparison of the supervision, evaluation embedding and feature representations for our model and others. $\mathcal{S}$: semantic embedding; Π: latent visual part similarity embedding; $\mathcal{F}$: raw visual feature embedding; S_O: semantic oracle; Π_{vo}: visual oracle.

ping to project the learned latent representation Π into the semantic space $\mathcal{S}$ (Eq.11) under the supervision provided by the semantic oracle, and during test time the semantic attributes for unobserved classes $\{s_o | o \in \mathcal{O}\}$ are also revealed. For this model, solely semantic supervision is leveraged, as same as all the competing methods. (2) **Ours(Π)**: We quantify the drawbacks of semantic information for GZSL by replacing semantic signals with visual signals Π_{vo} generated by our visual oracle, which is then used for both supervision and evaluation.

Training Details. To train our models, we take an alternative optimization approach where in each epoch, we update the weights in two steps. In step 1, only the weights of $G(\cdot)$ is updated by minimizing L_{prt}. In step 2, we freeze the weights of $G(\cdot)$ and update all the other modules. The semantic model (Ours(S)) and the visual oracle is trained by ϕ_{XY} while Ours(Π) model is trained by $\phi_{SX} + \phi_{XY}$ in step 2. Adam optimizer is used to optimize the loss in each step. The learning rate for step 1 and step 2 is set to 1e-6 and 1e-5, respectively.

Our models are trained for 80, 60 and 70 epochs on CUB, AWA2 and aPY, respectively. As for the visual oracle, it is trained to 70 epochs on AWA2 and 60 epochs on CUB and aPY. The feature extractor $E(\cdot)$ is initialized with ImageNet pretrained weights. The learning rate for $V(\cdot)$ and η in Eq.(11) is selected via cross-validation. For the optimization of L_{mix}, the EM algorithm is terminated if the loss did not change or after 300 steps.

Competing Methods. To validate the benefits of the proposed latent visual embedding, we compare against other state-of-the-art methods which also utilize semantic supervision and visual representation. Ten competitors are compared: SJE[2], ALE[1], and DEVISE[10] which use structured loss to learn a linear compatibility between visual and semantic space; SSE[28] learns the compatibility function in a latent common space for visual and semantic embedding; GFZSL[21] models the the class-conditional distribution as multi-variate Gaussian; CONSE[19] and SYNC[6] learns maps the unseen image into semantic representation via combination of seen classes or phantom classes; SAE[12] learns the mapping from semantic to visual embedding; PSRZSL[3] and SP-AEN[8] try to preserve the semantic relations in the mapping by encoder-decoder network or adversarial training.

4.1. Generalized Zero Shot Learning Evaluation

The results for the GZSL setting are shown in Table. 1. Observe that the proposed methods, Ours(S) and Ours(Π) consistently outperforms state-of-the-art methods in the GZSL setting. Specifically, the harmonic mean of the accuracy for seen (tr) and unseen (ts) classes with Ours(S) and Ours(Π) reaches 48.4%, 50.2% on CUB, 57.2%, 60.2% on AWA2, and 36.6%, 56.2% on aPY, which dominate other competing methods and often surpass the third-best result by a very large margin, e.g. a $> 20\%$ improvement on AWA2, and a $> 10\%$ improvement on aPY. While several competing methods (e.g. [21, 12, 19, 28]) only perform well on the seen classes and obtain close-to-zero accuracy on unseen classes, we are able to classify both seen and unseen

Methods	CUB		AWA2		aPY	
	SS	PS	SS	PS	SS	PS
SJE[2]	55.3	53.9	69.5	61.9	32.0	32.9
SAE[12]	33.4	33.3	80.7	54.1	8.3	8.3
SSE[28]	43.7	43.9	67.5	61.0	31.1	34.0
GFZSL[21]	53.0	49.3	79.3	63.8	51.3	38.4
CONSE[19]	36.7	34.3	67.9	44.5	25.9	26.9
ALE[1]	53.2	54.9	80.3	62.5	30.9	39.7
SYNC[6]	54.1	55.6	71.2	46.6	39.7	23.9
DEVISE[10]	53.2	52.0	68.6	59.7	35.4	39.8
PSRZSL[3]	-	56.0	-	63.8	-	38.4
SP-AEN[8]	-	55.4	-	58.5	-	24.1
Ours(S)	**63.7**	**66.7**	**90.7**	**69.1**	**52.1**	**50.1**
Ours(Π)	**68.8**	**71.9**	**92.4**	**84.4**	**54.4**	**65.4**

Table 4. Zero shot learning results on CUB, AWA2 and aPY. SS = standard split, PS = proposed split. The results are class-average Top-1 accuracy in %. The highest accuracy is in red color and the second is in blue (better viewed in color).

improving upon existing works in the GZSL setting.

State-of-art comparison with Semantic Supervision. Note that under identical conditions of semantic supervision, the gain in our method (ours(S)) can be attributed primarily to our latent visual embedding ($\Pi(\cdot)$). Different from the conventional visual representation, which is a high-dimensional deep CNN feature vector, and not semantically meaningful, $\Pi(\cdot)$ intrinsically describes the input image by a common vocabulary of prototypical parts. These prototypical parts are estimated by the latent mixture model with training images and the components of Π quantify the existence of a prototypical part in the instance. Such a representation resembles the semantic similarity of semantic vectors and leads to mitigating the visual-semantic gap.

Semantic vs. Visual Supervision. Observe that Ours(Π) consistently achieves better performance than Ours(S), e.g. 1.8%, 3.0% and 19.6% absolute improvement in the harmonic mean on the three datasets. This comparison shows that, although the proposed latent visual embedding Π is able to reduce the Visual→Semantic gap, the semantic attributes are noisy in that they contain information that are difficult to transfer from the visual domain (e.g. 'smelly', 'agility', 'weak'). Using semantic supervision and evaluation for GZSL thus fundamentally limits attaining high accuracy. By switching to visual supervision and evaluation provided by our visual oracle, we see the potential to further improve GZSL accuracy. This comparison is fair since our visual oracle provides only a list of similarity scores without any other identifying high-dimensional features to the learner. This is similar to the case of a semantic oracle providing attribute annotations.

Issue with aPY. Finally, observe that on aPY most existing methods fail to recognize unseen classes achieving nearly zero accuracy, while we get a significant improvement from Ours(S) and Ours(Π). The reason is that aPY attributes
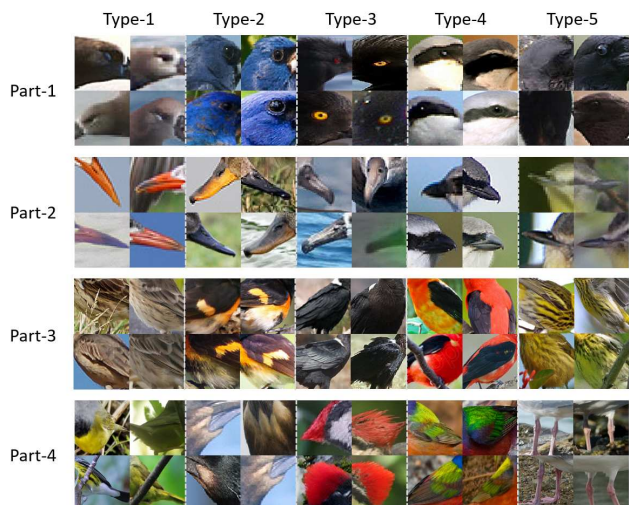

Figure 3. Example of types in each part on CUB dataset. Each two rows belong to a part and each two columns belong to a type. In m-th part, the example can be labeled with a scalar $\pi(k|m)$ indicating its probability of belonging to the k-th type. The example in the type has the largest $\pi(k|m)$ among all types. Note that these types are semantically meaningful and visually distinguishable.

are extremely noisy and are not visually representative (e.g. bus-car attributes nearly identical). Thus, the semantic supervision cannot provide useful information for training a GZSL model.

4.2. Zero Shot Learning Evaluation

We next evaluate the results for the traditional ZSL setting, where only unseen classes are included during testing. The results are reported in Table 4. Observe that many competing methods, which are ineffective in the GZSL setting (e.g. [21, 12, 19, 28]), realize a huge performance gain in the ZSL setting. However, Ours(S) and Ours(Π) is robust and still outperforms the competing methods. Again, this can be attributed to our proposed latent visual embedding. Ours(S) model consistently obtains superior performance, improving the state-of-the-art accuracy from 55.3% to 63.7% for SS on CUB, from 80.7% to 90.7% for SS on AWA2, and from 51.3% to 52.1% for SS on aPY. A similar improvement can also be observed on the PS split. This observation shows that, with the same level of semantic supervision, our low-dimensional latent visual embedding is more semantically meaningful than the traditional high-dimensional visual features, and thus effectively bridges the Visual→Semantic gap. Additionally, with the visual supervision provided by our visual oracle, Our(Π) obtains even better performance on all datasets and splits, which reveals the drawbacks of leveraging primarily semantic attributes for supervision and evaluation.

4.3. Analysis and Discussion

Visualization of discovered Latent Prototypical Part Types. To verify that our parameterized Π is able to learn

Methods	CUB					AWA2					aPY				
	ts	tr	H	SS	PS	ts	tr	H	SS	PS	ts	tr	H	SS	PS
Ours(Π_{flat})	38.4	**69.8**	49.6	66.8	69.5	42.6	**88.7**	57.6	91.7	84.0	36.5	**88.7**	51.7	53.6	62.9
Ours(Π)	**39.5**	68.9	**50.2**	**68.8**	**71.9**	**45.6**	**88.7**	**60.2**	**92.4**	**94.4**	**43.6**	78.7	**56.2**	**54.4**	**65.4**

Table 5. Test accuracy for different visual representations on CUB, AWA2 and aPY. Ours(Π_{flat}): our model with flatten visual representation. Ours(Π): our model with structured visual representation. The accuracy class-average Top-1 in %. Both gZSL (ts, tr, H) and ZSL (SS, PS) performances are reported.

Methods	CUB			AWA2		
	ts	tr	H	ts	tr	H
DEVISE (S)	23.8	53.0	32.8	17.1	74.7	27.8
DEVISE (Π_{vo})	**24.6**	**53.3**	**33.7**	**28.3**	**75.3**	**41.2**

Table 6. gZSL performances for DEVISE[10] using semantic supervision and visual oracle supervision (Π_{vo}).

diverse parts and discriminating types, we visualize some exemplar parts and types from CUB dataset in Figure 3. We observe that the examples in each type are visually similar to each other, but distinguishable to humans across different classes. When provided with the examples in each type, humans can score the existence of a type, i.e., $\pi_x(k|m)$, thereby bypassing the proposed visual oracle. Noticeably, the part feature model $f_m(\cdot)$ is able to detect some semantic parts. For example, in Figure 3, Part-1 detects the face (or eye), Part-2 detects the beak and Part-3 tends to detect the body texture representation of birds. These semantic parts are easier to be linked to the semantic attributes, and hence our visual semantic embedding is able to close the gap between the high-dimensional visual feature and the semantic space.

It is worth noting that in Part-4, different semantic parts, like head, chin, wins and legs, are discovered in different types. We find it reasonable since the same semantic parts may not appear in different classes. The situation could be even more common in coarse-grained recognition, like a chair is not likely to have an engine. Moreover, our model tries to learn the most discriminating part via the loss $\phi_{XY}(f_m(x), y)$. The same semantic part which is the most discriminating to one class is possibly not important to another class. This phenomenon also won't cause any problem for supervision because the visual oracle is based on visual features, while for a human being this is unlikely to arise.

Structured vs. Flat Visual Supervision In our formulation, both the feature model $f_m(x)$ and the mixture model $\Pi(x)$ are *structured*. That is, each part feature $f_m(x)$ is represented by a unique mixture model Π_m conditioned on the part m. The visual oracle's supervision Π_{vo} is also structured in a similar way. However, a different strategy is to take a flat representation and supervision: drop the part-based representation by replacing $f_m(x)$ with the global feature $E(x)$, and collapse the structured Π_{vo} into a single list representation. Such a flat supervision requires no part-wise features and its result is reported in Table 5. We observe that flattening the latent structured visual embedding as a single vector, which mirrors the common usage of

semantic attributes, suffers from a slight performance drop from Ours(Π) since the rich part structure information is lost. However, note that, compared to the noisy semantic supervision (see competitors in Table 1 and 4), the flat visual supervision still dominates competing methods.

Visual vs. Semantic Supervision. To further justify the effectiveness of the proposed latent visual embedding for GZSL supervision, we took an existing state-of-the-art approach, DEVISE[10], and re-trained it under the visual supervision Π_{vo}. As shown in Table 6, the proposed visual supervision also boosts DEVISE's GZSL performance, especially on AWA2, e.g. a 13.4% absolute improvement in the harmonic mean. The result demonstrates that the proposed latent visual embedding, as a supervision type, is effective and generalizable even to non-attention methods.

5. Conclusion

In this paper we proposed a novel Zero-Shot learning (ZSL) method. Our method unlike many existing works neither synthesizes unseen examples nor uses any unseen semantic information during training. We claim that semantic gap exists because visual features employed in prior work is not semantic leading to significant drop in accuracy. To bridge this semantic gap we proposed a new statistical model for embedding a visual instance into a low-dimensional probability matrix. Our insight is based on the fact that analogous to how a semantic component measures the likeliness of the attribute arising in an object, so also, our mixture component conveys visual likeliness by scoring how similar a part type is relative to proto-typical part types of other instances in the training set. To further reduce semantic noise we propose a novel visual oracle for supervision in lieu of semantic supervision. We tabulate results on a number of benchmark datasets demonstrating significant improvement in accuracy over state-of-art under both semantic and visual supervision.

Acknowledgement

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2016. 6, 7

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015. 6, 7

[3] Y. Annadani and S. Biswas. Preserving semantic relations for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 6, 7

[4] S. Antol, C. L. Zitnick, and D. Parikh. Zero-shot learning via visual abstraction. In *ECCV*, pages 401–416. Springer, 2014. 1

[5] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 2015. 1

[6] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016. 6, 7

[7] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 1, 2

[8] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2018. 2, 6, 7

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009. 5, 6

[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 2, 6, 7, 8

[11] H. Jiang, R. Wang, S. Shan, and X. Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[12] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017. 2, 6, 7

[13] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. 1

[14] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[15] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 36(3):453–465, 2014. 1, 2, 3

[16] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[17] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang. Zero-shot recognition using dual visual-semantic mapping paths. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[18] Y. Li, J. Zhang, J. Zhang, and K. Huang. Discriminative learning of latent features for zero-shot recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[19] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013. 6, 7

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1

[21] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 792–808. Springer, 2017. 6, 7

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 5, 6

[23] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[24] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 2, 5, 6

[25] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[26] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1):3–22, Aug. 2016. 5

[27] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[28] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174, 2015. 2, 6, 7

[29] H. Zheng, J. Fu, T. Mei, and J. Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Int. Conf. on Computer Vision*, volume 6, 2017. 4

[30] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2