Appendices

A Kmean Initialization

XXX

B Cropping Subnet Initialization

C Implementation details

We implement our approach on the Pytorch Framework. For the mutil-attention subnet, we take the images of size 448 by 448 as input to achieve high-resolution attention maps. For joint feature embedding subnet, we resize all the input images to the size 224 by 224. We consistently adopt VGG19 as the backbone and train the model with the batch size of 32 on two GPUs(TitanX). We use SGD optimizer with the learning rate of 0.05, the moment of 0.9 and weight decay of $5 * 10^{-4}$ to optimize the objectives. The learning rate is decay by 0.1 on the plateau, and the minimum one is set to $5 * 10^{-4}$. Hyper-parameters in our models are obtained by grid search on the validation set. mrgs in Eq.6 and Eq.11 are set to 0.2 and 0.8 respectively.

D Inference from SGMA model

We provide two ways to infer the labels of unseen class images from the JGLF model. The first one is straightforwardly to choose the class label with the maximal overall compatibility score, as the green path in Figure 2. An alternative way is utilizing the features $\phi_{cat}(x)$ learned in the class agent branch, as the purple path in Figure 2. We employ the inner product to measure the distances between the feature of the test image $\phi_{cct}(x)$ and the prototypes of unseen classes Φ^u_{cct}, which can be obtained by the following steps. The prototypes of seen classes Φ^s_{cct} is obtained by averaging the features of all image of each class. We assume the semantic descriptions of unseen classes can be represented by a linear combination of those of seen classes. Let W be the weight matrix of such a combination, and W can be obtained by solving the ridge regression:

$$W = \arg\min_{W} ||\Phi^u - W\Phi^s||_2^2 + \lambda||W||_2^2, \tag{1}$$

where Φ^u and Φ^s are the semantic matrices of unseen and seen classes with each row being the semantic vector of each class. Equipped with the learned W describing the relationship of the semantic vectors of seen and unseen classes, we can obtain the prototypes for unseen classes by applying the same W, $\Phi^u_{cct} = W\Phi^s_{cct}$.

To combine the global and local descriptions of images, we concatenate the visual features generated by different CNNs. Moreover, to combine the inference of two ways, the compatibility scores of the embedding softmax branch and the inner product of the class-agent triplet branch are added as the final prediction scores of the test image w.r.t. unseen classes:

$$y = \arg\min_{y\in\mathcal{Y}_{\mathcal{U}}} (s_y + \langle\phi_{cct}(x), [\Phi^u_{cct}]_y\rangle), \tag{2}$$

where $s_y = \theta(x)^T W\phi(y)$, and $[\cdot]_y$ denotes the row of the matrix corresponding to the class y.

E Evaluation on Different Semantic Embedding

In the section, we evaluate the performance of our model with different semantic embedding. The alternative semantic information we are using is Wikipedia articles, word2vec and fasttext embedding of class names. The work [Elhoseiny *et al.*, 2017] collected Wikipedia articles associated with each class. They extracted the TF-IDF feature as semantic representations of classes. We observe that the high-dimensional TF-IDF features (∼7000D) are sparse, so PCA is employed to compress them to 200D feature for efficiency. We directly use the 500D word2vec feature provided by [Morgado and Vasconcelos, 2017], and extract the 300D fasttext feature from the model trained on Wikipedia-2017. Figure 1 shows the comparison of our MAZSL model with four state-of-the-art methods. Overall, our MAZSL model outperforms others in all cases. It is not surprising that attributes enable the most effective transfer as attributes are defined to be discriminant properties of object classes. Although Wikipedia articles are highly suffering from noise, they are still more informative and discriminative semantic descriptors than word embeddings. Compare the results of two kinds of word embedding methods (i.e., word2vec, fasttext), we can find little difference.

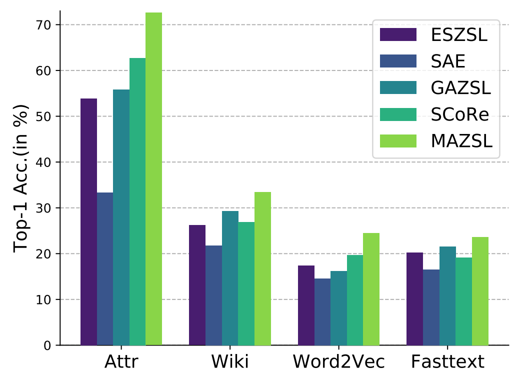

Figure 1: Zero-shot learning results on CUB with various types of semantic embeddings.

F t-SNE Visualizations of Learned Features

We visualize the image features for each class in AwA1 dataset using t-SNE [Maaten and Hinton, 2008] as shown in Figure 5. It is obvious that our trained features are more discriminative cross the classes compared with the feature without finetuning. Another notable observation is that the features from the triplet branch are more intra-class compact and inter-class separatable, explaining the better performance as shown in Table 3.

G More Quantitive Results for Part Detection

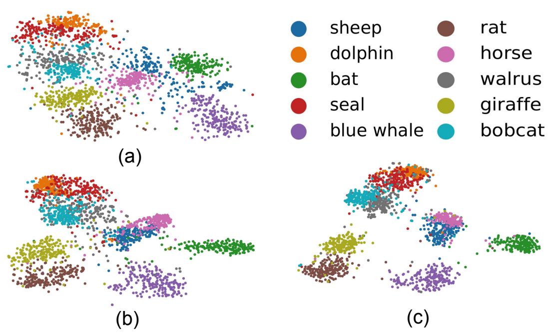

Figure 2: t-SNE visualization of feautures. (a) the features extracted from resnet101 pretrained on Imagenet, (b) the features from embedding softmax embedding branch, (c) the features from class-agent triplet branch.

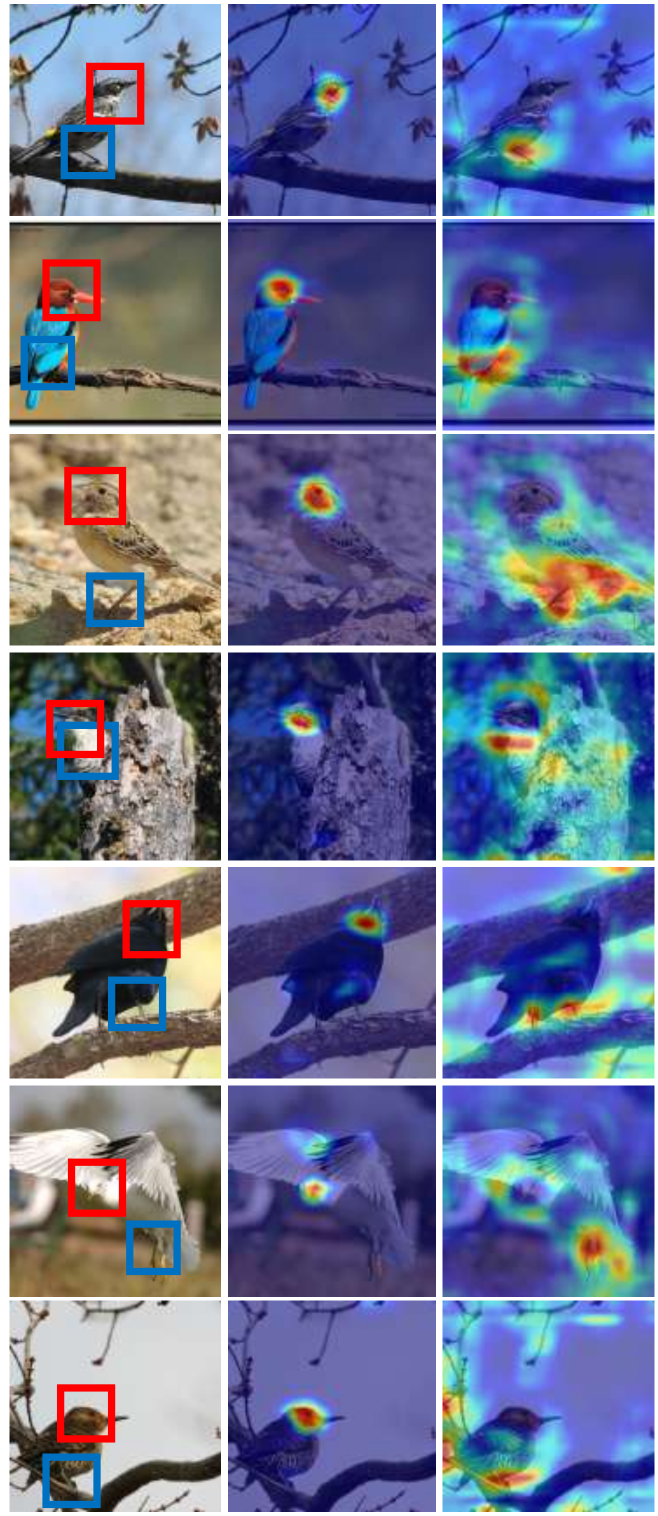

Figure 3: Part detection results on **CUB**

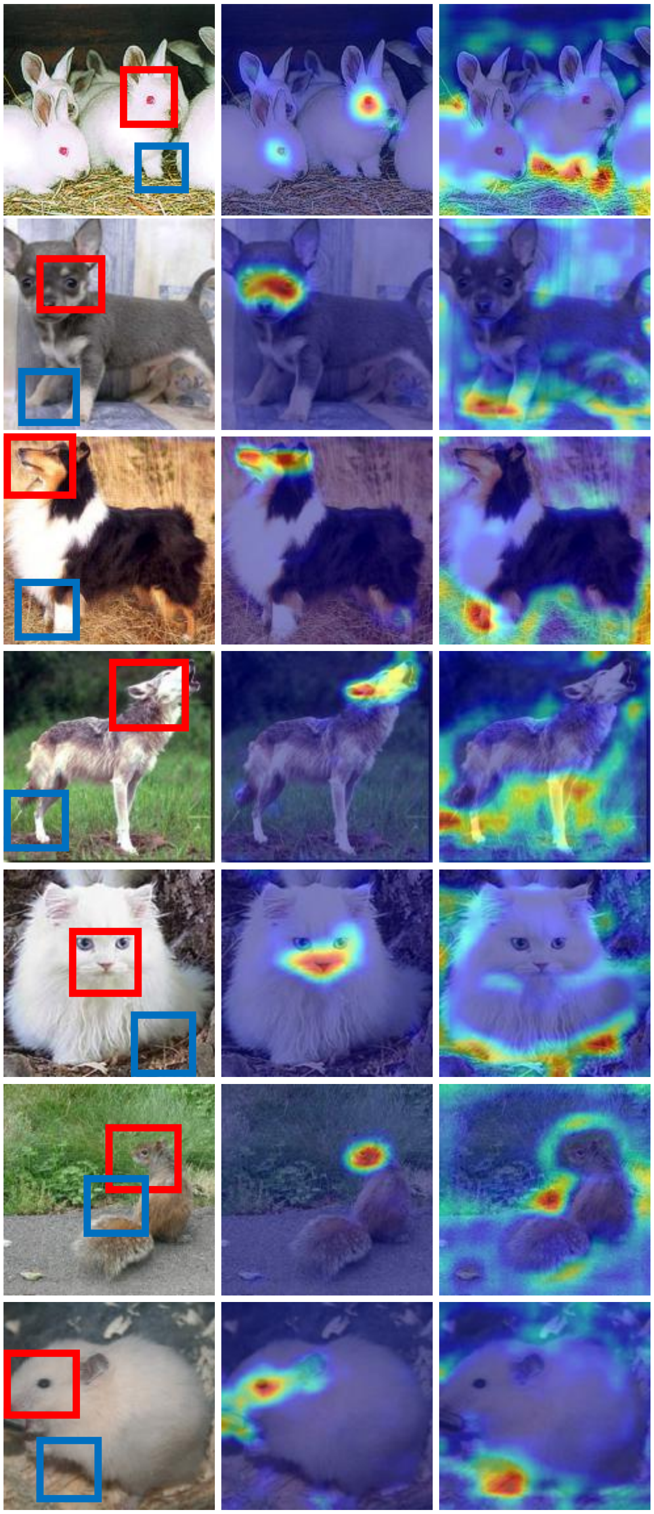

Figure 4: Part detection results on **AWA1**

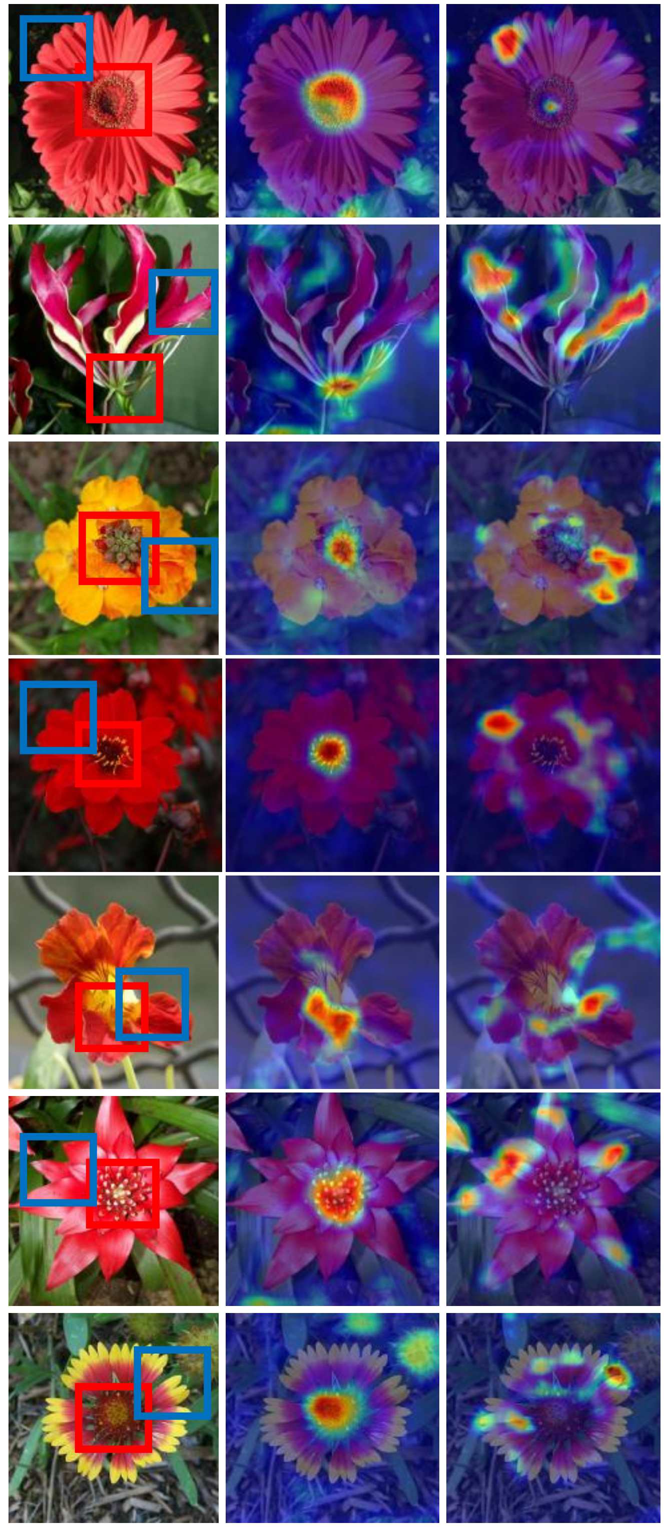

Figure 5: Part detection results on **FLO**

References

[Elhoseiny *et al.*, 2017] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the "beak": Zero shot learning from noisy text description at part precision. In *CVPR*, 2017.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Morgado and Vasconcelos, 2017] Pedro Morgado and Nuno Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *CVPR*, 2017.