Learning To Identify Seen, Unseen And Unknown In Open World: A Practical Setting for Zero-Shot Learning

A. Dataset Description

For our experiments, we consider three commonly used zero-shot learning (ZSL) datasets, namely a course-grained dataset: Animals with Attributes 1 (AWA1) [7] which contains images of different animals like polar bear, giraffe, seal etc., and three fine-grained datasets: Caltech-UCSD Birds-200-2011 (CUB) [1] which contains images of different species of birds, Oxford Flowers (FLO) [8] which consists of images of different kinds of flowers and SUN attribute [9] which contains images of different scene categories. It is worth noting that AWA1 dataset provides only the features extracted using ResNet-101 pre-trained on ImageNet dataset [5] and has not made the images publicly available. Furthermore, these ZSL datasets originally contain the seen and unseen splits for training and testing. We modify these datasets for the Open-Set Zero-Shot Learning (OZSL) by randomly choosing half of the original unseen classes as unseen classes, and the other half of the original unseen classes are taken as unknown samples. The seen classes are maintained as provided in the original split. As ResNet-101 pre-trained on the ImageNet dataset is used as the backbone to extract the visual features, care was taken in the original ZSL split so that unseen classes are not present in the ImageNet dataset. Hence, for our experiments as well, we consider unknown samples only from the original unseen classes split.

B. Settings and Implementation Details

As proposed in [14], we use ResNet-101, pre-trained on the Imagenet dataset, as the backbone to extract the visual features. A Multilayer perceptron (MLP) with one hidden layer and with ReLU activation is used as the encoder and decoder for both modalities for all the two stages. Further, the hidden layer dimension is fixed at 512, and the latent dimension is fixed at 64 for both modalities. Moreover, the classifiers in the latent space of both stage I and stage II follow a linear LogSoftmax architecture. Stage I classifier is used to classify seen class samples, and Stage II classifier is used to classify unseen class samples. Furthermore, inspired by [2, 4, 15] we use von Mises-Fisher (vMF) distribution as the prior and the posterior distribution in the VAE modules as it enhances the representation power by using a hypersphere as the latent space. Adam optimizer with a learning rate of 10^{-4} is used for training the model. As suggested in [2], we set λ_{cr} and λ_{cls} to 1 and Wasserstein distance weight λ as 0.1 for both stages. Stage I is trained using the seen class train data. GSM-Flow [3], a state-of-the-art ZSL approach, is used to generate synthetic samples for unseen classes for training Stage II. Further, while training Stage II, the weight of the proposed distribution retainment loss is set to 0.1. We implement our model using PyTorch. All our experiments are run on RTX 3090 GPU cards. We report the average results of the methods on five random unseen-unknown splits. It is important to note that the ECCV version of [2] followed the version of benchmark dataset in which there was a leakage of train seen data and test seen data (https://drive.google.com/file/d/1p9gtkuHCCCyjkyezSarCw-1siCSXUykH/view). The archive version of [2] provides the results with the fixed benchmark datasets.

B.1. Threshold Determination For Inference

As we don't have the actual visual features for the unseen classes and the unknown samples during training, and the generated synthetic unseen samples are only an approximation of the actual unseen samples, directly determining the threshold is difficult. Hence, we use one random split to determine the threshold value and use the same threshold value for other random splits for each dataset. For Stage I, We set γ^{I} as 0.98, 0.7, 0.85, 0.5 for AWA1, CUB, FLO and SUN Datasets respectively. Furthermore, for Stage II, we set γ^{II} as 0.98 for AWA1 and 0.65 for CUB, FLO and SUN datasets. It is interesting to note that course-grained datasets have a higher threshold and fine-grained datasets have a lower threshold. This is because, it is easier to distinguish samples of different classes in course-grained datasets as compared to fine-grained datasets.

B.2. Illustration of Threshold

As can be seen in Fig. 1, the proposed model is trained such that the unseen class samples are clustered around the unseen class attributes, whereas the unknown samples are

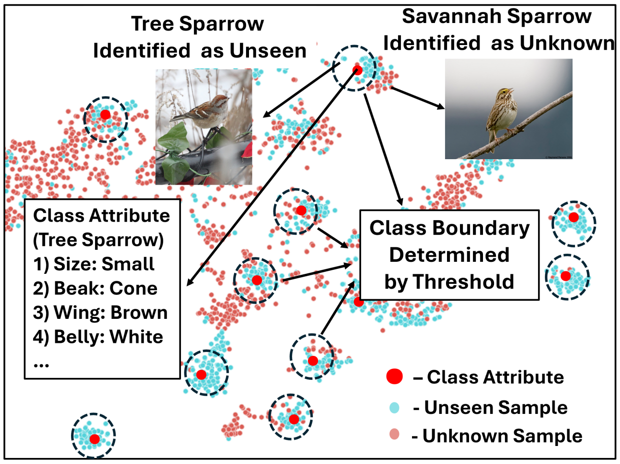

Figure 1. Illustration of unknown sample detection using the t-SNE plot of Stage II latent sapce for CUB dataset. The big red dots denote the attributes, the bule dots detnote the unseen class clusters and the scattered orange points denote the unknown samples.

scattered across the latent space. If a sample lies closer to the unseen class attributes, then it is categorized as unseen whereas if it lies further away from the unseen class attributes, it is categorized as unknown. The threshold determines the class boundary by setting how close a sample must lie near the unseen class attributes for it to be considered as unseen. In our approach, we use cosine similarity as the distance metric. As the sample from the Tree Sparrow class (unseen class) lies inside the class boundary (closer to the class attribute vector) it is identified as Unseen. Whereas, the sample from the Savannah Sparrow class (Unknown class) lies outside the class boundary and is flagged as unknown.

C. Baselines

We compare our proposed method to the following baselines, which are broadly from the following three categories, namely Zero-Shot Learning (ZSL), Open-Set Recognition (OSR) and Open-Set Zero-Shot Learning (OZSL). For the baselines from the ZSL category, we consider the state-of-the-art generative method, namely GSMFlow [3]. GSMFlow is a flow-based generative ZSL approach that generates synthetic samples for the unseen classes based on the unseen class attribute vectors. For the baseline from the OSR category, we consider MSP [6], ViM [13], KNN [11]. MSP is a simple threshold method that determines whether a sample is from the seen category or unknown based on a threshold on the softmax probability predicted by a classifier. ViM and KNN perform OSR by determining the probability a sample is unknown. Finally, for the baseline from the OZSL category, we consider the proposed naïve approaches, namely GSMFlow-Threshold, GSMFlow-ViM, GSMFlow-KNN. GSMFlow-Threshold is based on (MSP) [6] wherein we apply a threshold on the maximum similarity score predicted by GSMFlow to determine whether a sample is from in-distribution (seen + unseen) or out-of-distribution (unknown). If the sample is determined to be from in-distribution, then it is assigned the class for which it has the maximum similarity score. GSMFlow-ViM and GSMFlow-KNN applies ViM [13] and KNN [11] on the seen class train data and the synthetic unseen class data generated by GSMFlow in order to estimate the probability a sample is from out-of-distribution. Additionally, we also consider Contrastive Language-Image PreTraining (CLIP) [10]. However, CLIP is not originally designed for our setting as it is pre-trained on instance-level textual information, whereas other baselines and our proposed method are trained on a single description for each in-distribution class. Nevertheless, the pre-trained CLIP can still be used for ZSL. Likewise, CLIP-threshold is an extension that employs a threshold to the cosine similarity measure and can be used for OZSL. Lastly, a variant of CLIP, called CLIPN [12], is designed to handle unknown samples in the OZSL setting. As proposed in [12], we consider the ‘Agreeing to Disagree’ strategy for CLIPN.

D. Generator Description

As discussed in Section B, we use GSMFlow [3] as the generative model for generating synthetic samples for unseen classes. GSMFlow is a state-of-the-art flow based generative method that tries to address the following generation shifts, namely i) Semantic Inconsistency, ii) Variance Collapse and iii) Structure Disorder, in order to generate better synthetic samples. GSMFlow is first trained on the seen class train data conditioned on the corresponding seen class attribute vectors. Once trained, we condition a random vector sampled from the normal distribution with unseen class attribute vectors and pass it to the GSMFlow model to generate the synthetic samples for the unseen classes.

E. Source Of The Image In The Illustration

The example illustration image used in Fig.1 of the main paper is obtained from chatGPT4 using the prompt ”Generate a photo-realistic real world scenario consisting of a bicycle, motorcycle and a cat”

F. The standard VAE Loss

The standard VAE loss for the Visual and attribute VAE module can be written as:

$$L^{F}_{\text{VAE}} = \log p_{D^F}(\mathbf{x}|\mathbf{z}_{\mathbf{x}}) - \lambda d(q_{E^F}(\mathbf{z}_{\mathbf{x}}|\mathbf{x}) \| q_{E^A}(\mathbf{z}_{\mathbf{a}}|\mathbf{a})), \tag{1}$$

$$L^{A}_{\text{VAE}} = \log p_{D^A}(\mathbf{a}|\mathbf{z}_{\mathbf{a}}) - \lambda d(q_{E^A}(\mathbf{z}_{\mathbf{a}}|\mathbf{a}) \| q_{E^F}(\mathbf{z}_{\mathbf{x}}|\mathbf{x})), \tag{2}$$

where $\mathbf{x}$ and $\mathbf{a}$ are the visual features and the corresponding class attributes of a training instance, respectively; $\mathbf{z}_{\mathbf{x}}$ and $\mathbf{z}_{\mathbf{a}}$ denote the latent variable of the visual and attribute VAE modules, respectively. Furthermore, $q_{E^F}(\mathbf{z}_{\mathbf{x}}|\mathbf{x})$ and $q_{E^A}(\mathbf{z}_{\mathbf{a}}|\mathbf{a})$ are the posterior distributions of the visual and attribute VAE modules, modeled by the visual encoder E^F and the attribute encoder E^A, respectively. Meanwhile, $p_{D^F}(\mathbf{x}|\mathbf{z}_{\mathbf{x}})$ is modeled by the decoder network D^F of the visual VAE module, while $p_{D^A}(\mathbf{a}|\mathbf{z}_{\mathbf{a}})$ is modeled by the decoder network D^A of the attribute VAE module. Note that, in the two losses, the first term represents the reconstruction error of a single modality (visual features and attributes, respectively); the second terms represents the alignment error across the two modalities. Here, $d(\cdot\|\cdot)$ measures the Wasserstein distance between two distributions, and λ is a hyperparameter balancing the two terms.

G. VAE Based visual and Attribute Module

As discussed in Section B, we use von Misses-Fisher (vMF) distribution as the prior and posterior distribution for our VAE-based visual and attribute module. The VAE module models the latent space by predicting the mean μ and the concentration parameter κ of the vMF distribution. Fig. 2 illustrates various losses used in our model. L_R denotes simple reconstruction loss where the decoder must reconstruct the input feature from the latent variable z sampled from the vMF distribution with parameter μ and κ predicted by the encoder. L_{cr} denotes the cross-reconstruction loss where the latent variable sampled from one module is passed to the decoder of the other module to reconstruct the corresponding input feature of that module. L_w denotes the loss that minimizes the Wasserstein distance between the latent space of the visual module and the attribute module. L_{cls} is the classification loss applied to the latent space of the two modules. It is interesting to note that stage I is trained using the seen class visual train data and the corresponding seen class attributes. However, since we do not have any visual train data for unseen classes, stage II is trained using synthetic unseen class samples and corresponding unseen class attributes.

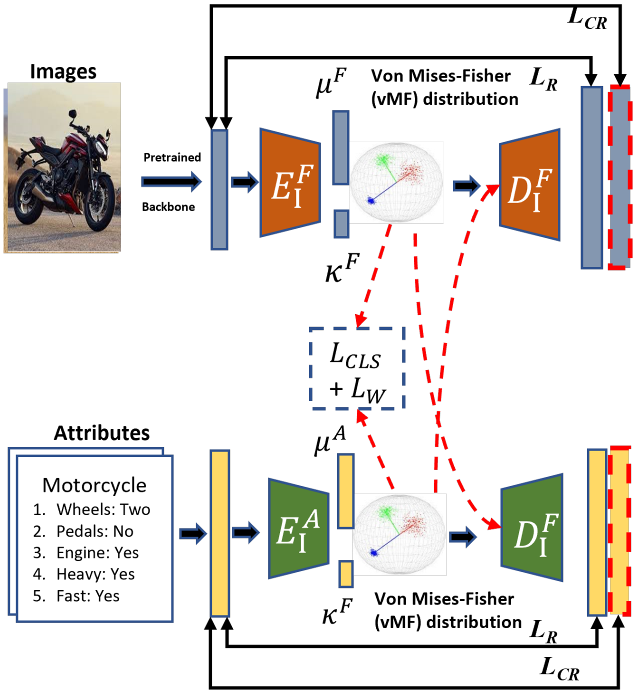

Figure 2. Illustration of the VAE based Generative model using vMF distribution

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, 2015. 1

[2] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. *arXiv e-prints*, pages arXiv–2008, 2020. 1

[3] Zhi Chen, Yadan Luo, Sen Wang, Jingjing Li, and Zi Huang. Gsmflow: Generation shifts mitigating flow for generalized zero-shot learning. *IEEE Transactions on Multimedia*, 2022. 1, 2

[4] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence*, pages 856–865, 2018. 1

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[6] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2016. 2

[7] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 1

[8] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth*

Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729, 2008. 1

[9] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108:59–81, 2014. 1

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 2

[11] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022. 2

[12] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1802–1812, 2023. 2

[13] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4921–4930, 2022. 2

[14] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5542–5551, 2018. 1

[15] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, 2018. 1