Supplementary Material for Improving Generalized Zero-Shot Learning by Exploring the Diverse Semantics from External Class Names

Yapeng Li[1], Yong Luo[1,2], Zengmao Wang[1] *, Bo Du[1,2*]
[1]National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Wuhan University, Wuhan, China.
[2] Hubei Luojia Laboratory, Wuhan, China.

https://github.com/li-yapeng/DSECN

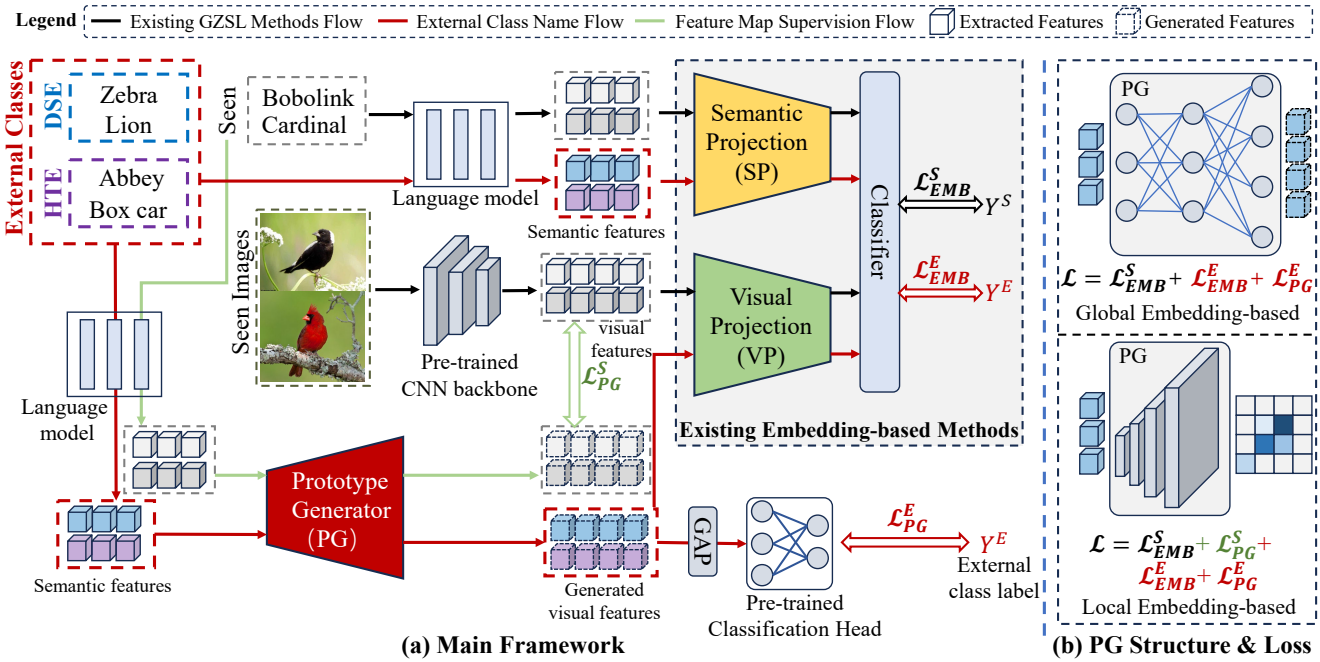


Figure S1. Illustration of integrating DSECN into existing embedding-based GZSL methods (EBGZSL). (a) Main framework: the framework contains three components. The existing GZSL method flow only utilizes the paired semantic features and visual features from seen classes to establish the relations between semantic and visual features, thereby limiting the performance of recognition for dissimilar unseen classes. The external class name flow introduces diverse visual-semantic relations from external class names, thus assisting the recognition of dissimilar unseen classes. The feature map supervision flow is only used when the existing GZSL method belongs to local EBGZSL method, and is used to constrain the generated feature maps of seen classes to be consistent with the GT feature maps. (b) PG Structure & loss: for global EBGZSL methods, we adopt a simple MLP as PG to generate global visual features of external classes, and the training objective includes the original loss $\mathcal{L}^S_{EMB}$ of existing GZSL method, and the loss $\mathcal{L}^E_{EMB} + \mathcal{L}^E_{PG}$ from the external class name flow. In contrast, for local EBGZSL methods, we employ convolutional neural network as PG to generate the feature map from global semantic features. The loss $\mathcal{L}^S_{PG}$ of feature map supervision flow is added to the training objective to constrain the generated feature map in feature map level. The classification labels Y^E of the external classes are the sets of Y^{EB} and Y^{HA} in the main paper, i.e., $Y^E = \{Y^{EB}, Y^{HA}\}$.

*Corresponding authors: Zengmao Wang, Bo Du

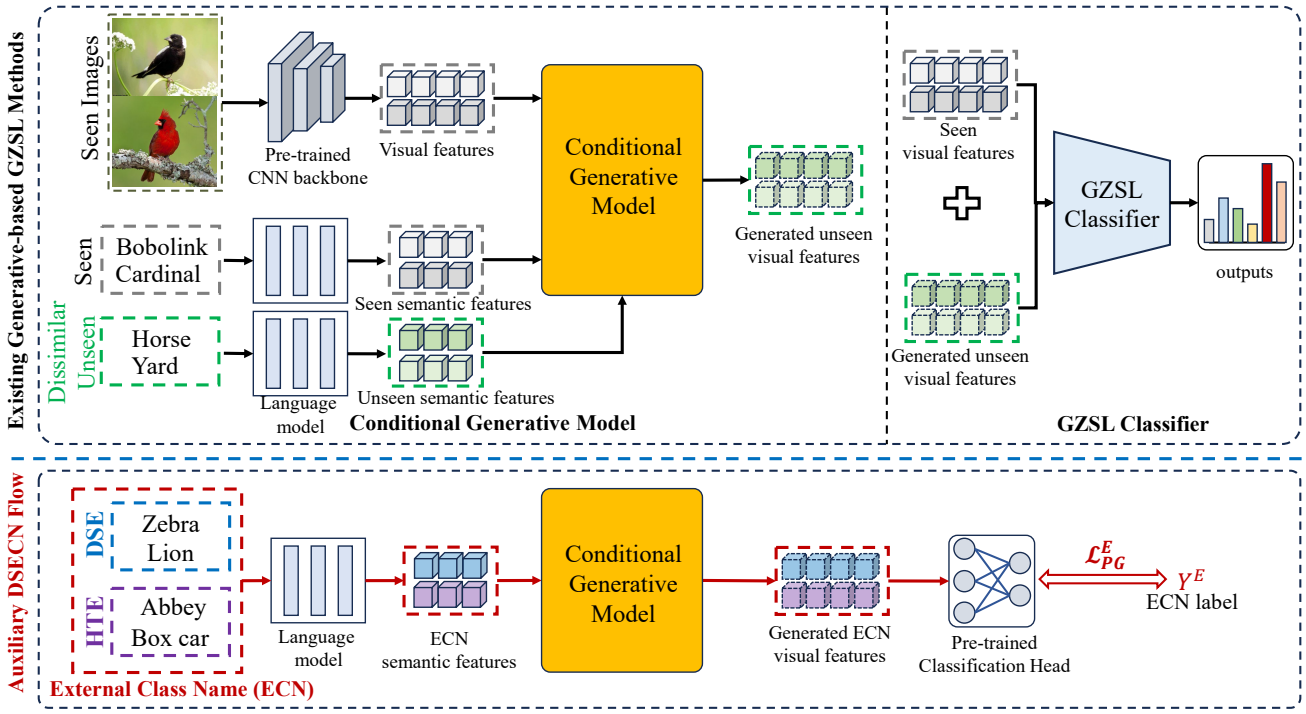


Figure S2. Illustration of integrating DSECN into existing generative-based GZSL methods. The existing generative-based GZSL methods (GBGZSL) learn a conditional generative model based on samples of seen classes conditioned on their semantic features, and then the learned generative model is used to generate visual features of unseen classes using the semantics of unseen classes. Next, the GZSL classifier is trained using the generated visual features of unseen classes and the visual features of seen classes. By introducing the auxiliary DSECN flow into the existing GBGZSL methods, the conditional generative model can utilize the diverse relations between semantics and visual features from external classes, which enables the conditional generative model to generate more accurate visual features of dissimilar unseen classes. The more accurately generated unseen class visual features facilitate the training of the GZSL classifier, thereby assisting in the identification of dissimilar unseen classes.

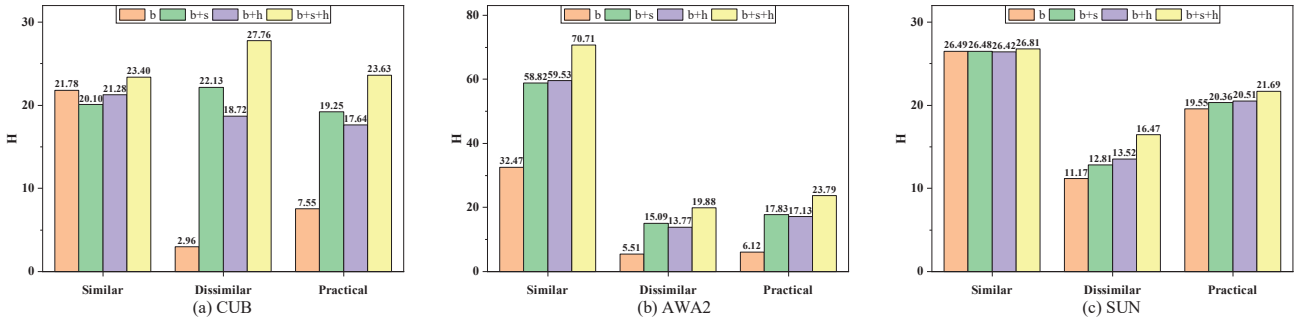
A. Integrating into Existing GZSL Methods

The proposed DSECN can be easily integrated into other GZSL approaches and improve their robustness for dissimilar unseen classes. In the following subsections, we describe the details of how to integrate DSECN into embedding-based and generation-based GZSL methods.

A.1. Embedding-based Methods

As shown in Fig. S1, the existing embedding-based GZSL methods (EBGZSL) align the semantic features and visual features of seen classes to a common space, and then the learned embedding space is used to perform recognition. However, when the unseen classes are dissimilar to seen classes, they perform poorly since they can only transfer little information from seen classes to recognize the dissimilar unseen classes. The proposed DSECN can easily be integrated into the existing EBGZSL methods to introduce diverse semantic-visual relations of external classes, thus improving their performance for identifying dissimilar unseen classes. The existing EBGZSL methods can be broadly categorized into global EBGZSL and local EBGZSL. The global EBGZSL methods, such as CN [7], take the paired global visual features and semantic features as inputs to establish the relations between semantic and visual features. In contrast, the local EBGZSL methods, such as TransZero [2], extract the feature map using a pre-trained backbone, and utilize semantic descriptions as guidance to discover the discriminative local features between seen classes and unseen classes. Next we describe how to integrate DSECN into the current mainstream global and local EBGZSL methods.

Global EBGZSL methods. The existing GZSL flow denotes the original pipeline of existing GZSL methods. DSECN is integrated into the existing global EBGZSL methods by the external class name flow (ECNF). Specifically, the diverse semantics from external classes are fed into the prototype generator (PG) to generate the visual features of external classes. Then we constrain that the generated semantic features should be able to be accurately classified with a frozen pre-trained classification head. Next, the generated visual features and semantic features of external classes are used to train the existing EBGZSL methods via $\mathcal{L}_{EMB}^{E}$. This enriches the visual-semantic relationships available to the trained GZSL model, thus assisting in the



(a) CUB (b) AWA2 (c) SUN

Figure S3. Effect of diverse semantic enhancement (s) and hierarchy taxonomy enhancement (h) for GZSL. We remove these two components as our baseline (b). In the ablation study, we add DSE (s) and HTE (h) step by step to show their effect on GZSL.

identification of dissimilar unseen classes.

Local EBGZSL methods. DSECN is integrated into the existing local EBGZSL methods by the external class name flow (ECNF) and the feature map supervision flow (FMSF). The ECNF pipeline for local and global EBGZSL is similar. The difference is that the structure of PG uses a convolutional neural network to generate feature maps instead of using MLP to generate global feature vectors. Another difference is that the generated feature maps need to be converted into global visual features required by the frozen pre-trained classification head through a global average pooling (GAP) layer. Considering that the feature map generated by PG should be as close as possible to the real feature map, the FMSF is further introduced into the existing GBGZSL methods. Specifically, the PG takes the semantic features of seen classes as inputs to generate the visual features of seen classes, and then the generated visual feature maps are constrained to have the smallest mean square error with the real seen class feature maps extracted by pre-trained backbone.

A.2. Generative-based Methods.

As shown in Fig. S2, the mainstream generative-based GZSL methods (GBGZSL), such as DGZ [1], learn a generative model to generate the visual features of unseen classes. Then a GZSL problem can be converted into a conventional supervised learning problem. Therefore, the core of the GBGZSL method is the accurate generation of unseen class visual features. However, the existing GBGZSL methods heavily rely on the semantics and visual features from seen classes to learn the conditional generative model. This makes these models perform poorly on dissimilar unseen classes, as little information can be transferred from seen to dissimilar unseen classes. Hence, the proposed DSECN method is integrated into the existing GBGZSL methods to assist in the learning of the conditional generative model. Specifically, the auxiliary DSECN flow introduces the diverse semantics from external class names. Then the external class visual features generated by the conditional generative model are constrained to be as consistent as possible with the real visual features through the frozen pre-trained classification head. This enables the learned generative model to simultaneously utilize the semantic-visual relationships of seen classes and external classes to generate visual features of dissimilar unseen classes, thereby improving the recognition performance of the GBGZSL methods for dissimilar unseen classes.

B. Implementation Details

We use the data splits proposed by [8] and extract visual features (with $d^v = 2048$) for each image by the ResNet101 [4] backbone pre-trained on ImageNet-1K [3]. The semantic embedding of classes are extracted by the text encoder of CLIP VIT/B-32 [6] and W2V [5]. Following [7], the scale factor γ of the visual classifier VC is set with 5. The Adam optimizer is used to train our model. The learning rate and weight decay of Adam are 5×10^{-4} and 10^{-4}. The training epochs and batch size are set to 50 and 256. We use PyTorch to implement our model. All runs are conducted on the same hardware: NVIDIA GeForce RTX 3090 GPU, $\times 64$ Intel Xeon Gold 6226R CPU and 256 GB RAM.

C. Ablation Study on W2V Embedding

The results are presented in Fig. S3, with further analysis provided in Sec. 4.3 of the main paper.

D. Qualitative Insight

To provide additional qualitative insight into the improvements that can be gained using diverse semantic enhancement s and hierarchy taxonomy enhancement h, we randomly select five seen classes from the CUB dataset. The five unseen classes are randomly selected from the complementary set of CUB ($du = \complement_{Set} CUB = \{AWA2, SUN\}$), which ensures that the unseen classes are dissimilar to seen classes. The PCA visualization of the features (X^u, V^u, V^s) for these classes are shown in Fig. S4. From the results, we can see that when the baseline b only utilizes the semantics from seen classes, the synthesized visual prototypes of unseen classes V^u (★) are close

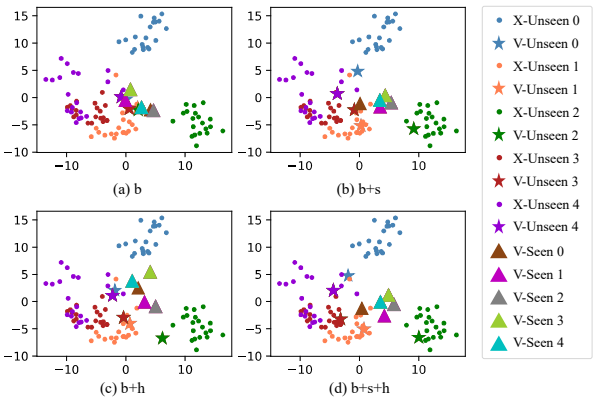


Figure S4. Qualitative evaluation with PCA visualization. The • denotes the visual features of test samples from unseen classes X^u. The ★ and ▲ denote the synthesized class-level visual prototypes of unseen classes V^u and those of seen classes V^s, respectively. The synthesized class-level visual prototypes V are extracted by the semantic-to-visual network (S2V). We use 10 colors to denote randomly selected 5 seen classes and 5 unseen classes. Refer to § D.

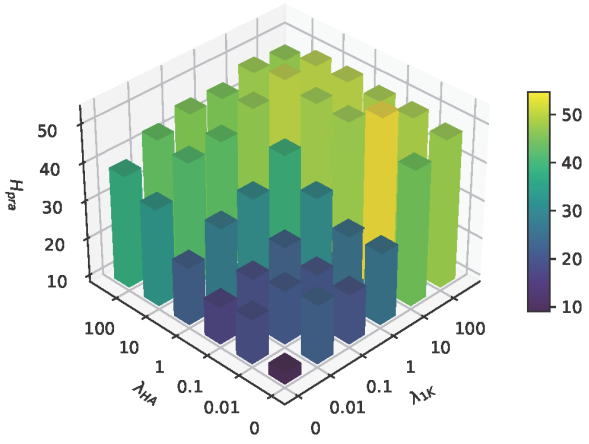


Figure S5. Hyper-parameter analysis on AWA2 datasets.

to those of seen classes V^s (▲) and far from the real visual features of unseen classes X^u (•). This means that if the GZSL model only utilizes the semantics from seen classes, the GZSL model tends to misclassify unseen classes into seen classes. By adding the s or h, the distance between V^u (★) and X^u (•) is reduced. It demonstrates that introducing additional semantic information exploited from external class names, the similarities between synthesized visual prototypes V^u (★) and visual features X^u (•) of samples from unseen classes can be improved, and thus boost the recognition ability of unseen classes. The $b+s+h$ adds both s and h to the baseline, and the distance between V^u (★) and X^u (•) is the closest. This indicates that the diverse semantic enhancement s and hierarchy taxonomy enhancement h are complementary to each other for GZSL.

E. Hyper-Parameter Analysis

There are two main hyper-parameters in our algorithm, i.e., the trade-off factor (λ_{EB}) and (λ_{HA}). We conduct the hyper-parameter analysis on the AWA2 dataset with CLIP semantic embedding. λ_{EB} and λ_{HA} are chosen from $\{0, 0.01, 0.1, 1, 10, 100\}$. The harmonic mean accuracy under practical GZSL setting H_{pra} is chosen as the evaluation metric. The results are reported in Fig. S5. It can be observed that: 1) when $\lambda_{EB} = 0$ and $\lambda_{HA} = 0$, the diverse semantic enhancement (DSE) and hierarchy taxonomy enhancement (HTE) are not used. It leads to very poor performance; 2) as λ gradually increases, the overall performance becomes better, which demonstrates the effectiveness of DSE and HTE; 3) the proposed method can obtain satisfactory performance at a wide range of λ greater than 10. This indicates that our algorithm is not quite sensitive to these hyper-parameters.

References

[1] Dubing Chen, Yuming Shen, Haofeng Zhang, and Philip HS Torr. Deconstructed generation-based zero-shot model. In *AAAI*, pages 295–303, 2023. 3

[2] Shiming Chen, Ziming Hong, Yang Liu, Guo-Sen Xie, Baigui Sun, Hao Li, Qinmu Peng, Ke Lu, and Xinge You. Transzero: Attribute-guided transformer for zero-shot learning. In *AAAI*, pages 330–338, 2022. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 3

[6] 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 *ICML*, pages 8748–8763. PMLR, 2021. 3

[7] Ivan Skorokhodov and Mohamed Elhoseiny. Class normalization for (continual)? generalized zero-shot learning. In *ICLR*, 2021. 2, 3

[8] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 41(9): 2251–2265, 2018. 3