Dual Progressive Prototype Network for Generalized Zero-Shot Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generalized Zero-Shot Learning (GZSL) aims to recognize new categories with auxiliary semantic information, *e.g.,* category attributes. In this paper, we handle the critical issue of domain shift problem, *i.e.*, confusion between seen and unseen categories, by progressively improving cross-domain transferability and category discriminability of visual representations. Our approach, named Dual Progressive Prototype Network (DPPN), constructs two types of prototypes which record prototypical visual patterns for attributes and categories, respectively. With attribute prototypes, DPPN alternately searches attribute-related local regions and updates corresponding attribute prototypes to progressively explore accurate attribute-region correspondence. This enables DPPN to produce visual representations with accurate attribute localization ability, which benefits the semantic-visual alignment and representation transferability. Besides, along with progressive attribute localization, DPPN further projects category prototypes into multiple spaces to progressively repel visual representations from different categories, which boosts category discriminability. Both attribute and category prototypes are collaboratively learned in a unified framework, which makes visual representations of DPPN transferable and distinctive. Experiments on four benchmarks prove that DPPN effectively alleviates the domain shift problem in GZSL.

1 Introduction

Deep learning methods depend heavily on enormous manually-labelled data, which limits their further applications [8, 9, 18, 52, 36, 20]. Therefore, Generalized Zero-Shot Learning (GZSL) recently attracts increasing attention, which aims to recognize images from novel categories with only seen domain training data. Due to unavailable unseen domain data during training, GZSL methods introduce category descriptions, such as category attributes [16, 15] or word embedding [5, 37, 26, 41], to associate two domain categories.

A basic framework of embedding-based GZSL is to align global image representations with corresponding category descriptions in a joint embedding space [17, 3, 55, 38, 2, 6], as shown in Fig. 1 (a). Due to the domain shift problem across two domain categories, unseen domain images tend to be misclassified as seen categories. To address this issue, recent methods focus on discovering discriminative local regions to capture subtle differences between two domain categories. For example, AREN [45] and VSE [55] leverage attention mechanism to discover multiple important part regions, which improves feature discrimination. DAZLE [21] and RGEN [46] introduce the semantic guidance, *e.g.*, category attributes, into region localization to narrow the semantic-visual gap. Among existing methods, APN [47] is most related to our approach. Specifically, APN constructs visual prototypes to describe visual patterns for different attributes, and these prototypes are shared across all images to search attribute-matched local regions, as shown in Fig. 1 (b). However, due to image

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

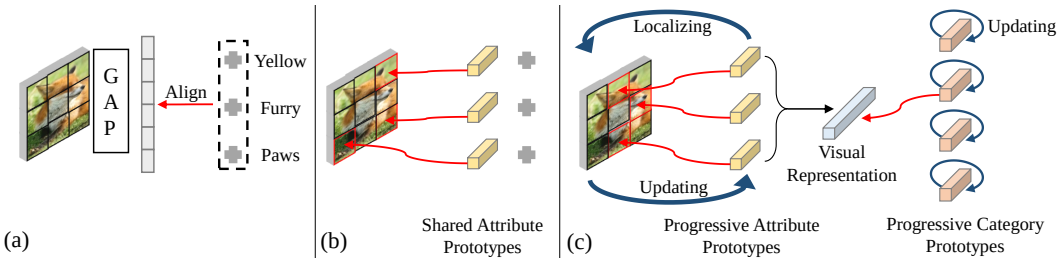


Figure 1: The motivation of DPPN. (a) General GZSL methods directly align global image features with category attributes. (b) A typical part-based method, *i.e.*, APN [47], learns prototypes shared by all images for attribute localization. (c) DPPN progressively updates prototypes according to different images and introduces category prototypes to enhance category discriminability.

variances, the textures corresponding to the same attribute may vary seriously across images. Thus, it is unreasonable for APN to apply shared prototypes to all images.

In this paper, we propose a novel Dual Progressive Prototype Network (DPPN), which constructs two types of progressive prototypes for respective attributes and categories to gradually improve cross-domain transferability and category discriminability of visual representations. Instead of shared prototypes, DPPN dynamically adjust attribute prototypes for each image by alternately localizing attribute regions and updating attributes prototypes, as shown in Fig. 1 (c). Notably, the visual clues in the localized attribute regions can improve the adaptation of attribute prototypes to a target image, and better attribute prototypes will boost attribute localization in turn. This alternation strategy progressively updates attribute prototypes to localize attribute regions more accurately, which narrows the semantic-visual gap and thereby improves representation transferability. Besides, category prototypes contain prototypical visual patterns for different categories. Along with progressively-updated attribute prototypes, DPPN projects category prototypes into multiple spaces to progressively enlarge category margins, which strengthens category discriminability of visual representations. Consequently, with cross-domain transferability and category discriminability, DPPN can effectively bridge the gap between seen and unseen domains.

Experiments on four benchmarks demonstrate that our DPPN alleviates the domain shift problem in GZSL and obtains new state-of-the-art performance. Our contributions can be summarized as three-fold. a) We propose a novel Dual Progressive Prototype Network (DPPN) that constructs progressive prototypes for both attributes and categories to gradually improve cross-domain transferability and category discriminability of visual representations. b) We design an alternation updating strategy to dynamically adjust attribute prototypes according to different images. Besides, DPPN projects category prototypes into multiple spaces to progressively enhance category discriminability. c) DPPN is end-to-end trainable and collaboratively learns attribute and category prototypes in a unified way.

2 Related Work

Generalized Zero-Shot Learning. GZSL aims to recognize new categories using semantic knowledge transferred from seen categories. Early GZSL methods learn a joint embedding space to align global image representations with corresponding category descriptions, *e.g.,* attributes [16, 15, 35] or text descriptions [5, 37]. Since unseen and seen categories share a common semantic space, the semantic-aligned image representations can be transferred from seen to unseen domain. Based on this paradigm, many works focus on improving the discrimination of embedding space by designing elaborate semantic-visual alignment functions [38, 44, 1, 54, 3, 55, 27]. For example, some methods [40, 53, 55, 22, 33] use high-dimensional visual features to span the embedding space, which is proved more discriminative than that spanned by category attributes. Other methods [23, 10, 42, 32] utilize auto-encoders to preserve semantic relationships between categories in the embedding space.

Though effective, these methods suffer from the domain shift problem, *i.e.,* images from two domains have different data distributions. Since only seen domain images are available during training, images of unseen categories tend to be recognized as seen categories. To this end, DVBE [31] and Boundary-based OOD [11] explore out-of-distribution detection to treat seen and unseen domains separately. Some works [28, 21] suppress the seen category confidence when recognizing images

to better distinguish two domain samples. These methods can effectively alleviate the domain shift problem via extra processing, but they ignore the intrinsic potential of local attribute localization in distinguishing two domains.

Part-Based GZSL. Since global image representations contain much noisy background information which is trivial for knowledge transfer, recent part-based methods [49, 45, 47] aim to capture important visual details and localize part regions to better understand the semantic-visual relationship. For example, S^2GA [49], AREN[45], and VSE [55] leverage the attention mechanism to learn semantic-relevant representations by automatically discovering discriminative parts in images. RGEN [46] uses region graph to introduce region-based relation reasoning to GZSL and learns complementary relationships between different region parts inside an image. GEM-ZSL [30] imitates human attention and predicts human gaze location to learn visual attention regions. Usually, the semantic guidance, *e.g.,* category attributes, is used to guide the part localization [29]. Thus their generated local features can better match corresponding category attributes, which can alleviate the domain shift problem. Instead of treating category attributes as an all-in-one vector, DAZLE [21] proposes a dense attribute attention mechanism to produce local attention for each attribute, separately. APN [47] constructs prototypes for separate attributes, which are shared by all images to localize attribute-related regions via region searching. In this paper, instead of using prototypes shared by all images, DPPN dynamically adjusts attribute prototypes according to different images, which learns representations with more accurate attribute localization. Besides, in the proposed DPPN, we also design progressive category prototypes to enhance category discriminability of visual representations.

3 Dual Progressive Prototype Network

3.1 Problem Formulation

The target of GZSL is to recognize images of novel categories trained with only seen domain data. In this paper, we denote $\mathcal{S} = \{X, y, \boldsymbol{a}_y | X \in \mathcal{X}_s, y \in \mathcal{Y}_s, \boldsymbol{a}_y \in \mathcal{A}_s\}$ as seen domain data, where $X \in \mathbb{R}^{C\times N}$ indicates image features extracted by the backbone network, and $X_n \in \mathbb{R}^{C\times 1}$ encodes the local information at the n-th region. y is the corresponding category label, and $\boldsymbol{a}_y \in \mathbb{R}^{N_a\times 1}$ is the category description, such as the category-level vector with N_a attributes. C is the number of feature channels, and $N = W \times H$. The unseen domain data is similarly defined as $\mathcal{U}$, and $\mathcal{Y}_s \cap \mathcal{Y}_u = \phi$. Given $\mathcal{S}$ during training, GZSL aims to recognize images from either $\mathcal{X}_s$ or $\mathcal{X}_u$ during inference. A basic framework is to learn an image representation $f(X)$ that is aligned with corresponding category attributes by minimizing:

$$\mathcal{L}_{v2s} = -\sum_{X\in\mathcal{X}_s} \log \frac{\exp(f(X)^{\mathrm{T}}\boldsymbol{a}_y)}{\sum_{j\in\mathcal{Y}_s}\exp(f(X)^{\mathrm{T}}\boldsymbol{a}_j)}, \tag{1}$$

where $f(\cdot)$ is a visual projection function, which is generally implemented via Global Average Pooling (GAP) and linear projection.

Based on $\mathcal{L}_{v2s}$, APN [47] expects to improve the localization ability of attributes for intermediate feature X. Thus, APN constructs a set of attribute prototypes $\mathcal{P} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{N_a}\}$, where $\boldsymbol{p}_i \in \mathbb{R}^{C\times 1}$ records visual patterns for the i-th attribute, *e.g.,* depicting what attribute "Yellow Wing" looks like. $\mathcal{P}$ is learnable and shared for all images. With $\mathcal{P}$, APN learns attribute localization by minimizing:

$$\mathcal{L}_{apn} = \sum_{x\in\mathcal{X}_s} d(l(X, \mathcal{P}), \boldsymbol{a}_y). \tag{2}$$

$l(X, \mathcal{P})$ is an attribute localization function that searches the most related local feature X_n for $\boldsymbol{p}_i \in \mathcal{P}$, which regresses attributes $\hat{\boldsymbol{a}}_y \in \mathbb{R}^{N_a\times 1}$. $d(\cdot,\cdot)$ is a distance measurement function, *e.g.,* l_2 norm, that aligns the predicted attributes $\hat{\boldsymbol{a}}_y$ and ground truth attributes $\boldsymbol{a}_y$. Finally, the main insight of APN is to minimize $\mathcal{L}_{apn} + \mathcal{L}_{v2s}$, and the inference function is:

$$\hat{y} = \arg\min_{y\in\mathcal{Y}_s\cup\mathcal{Y}_u} d(f(X), \boldsymbol{a}_y). \tag{3}$$

However, due to instance variance, occlusion, and noise, the visual textures that correspond to the same attribute may vary severely across different images. Thus, it is unreasonable to only rely on shared attribute prototypes $\mathcal{P}$ to accurately localize attribute-related regions for each individual image.

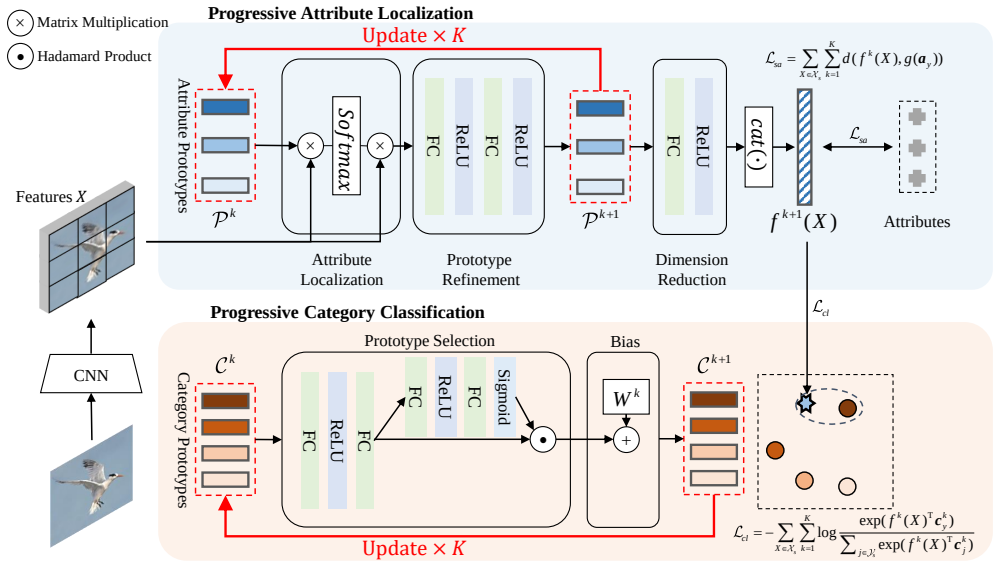


Figure 2: Training pipeline of DPPN. DPPN progressively learns attribute prototypes $\mathcal{P}^k$ and category prototypes $\mathcal{C}^k$ via $\mathcal{L}_{sa}$ and $\mathcal{L}_{cl}$. $\{f^k(X)|k=1,\cdots,K\}$ contain visual representations of K iterations, which gradually capture attribute localization and category discrimination.

Beside, $\mathcal{L}_{apn}$ is applied to intermediate features X as just a constraint. During inference, attribute localization part $l(X,\mathcal{P})$ is discarded, and the GAP in $f(\cdot)$ will damage the attribute localization ability of the final visual representation $f(X)$.

To this end, our Dual Progressive Prototype Network (DPPN) progressively and dynamically explores attribute localization and category discrimination for different images with two modules, *i.e.,* Progressive Attribute Localization and Progressive Category Classification, as shown in Fig. 2.

3.2 Progressive Attribute Localization

The Progressive Attribute Localization (PAL) module aims to dynamically adjust attribute prototypes according to the target image to progressively capture local correspondence between different attributes and image regions.

Define $\mathcal{P}^0 = \{\boldsymbol{p}_1^0, \cdots, \boldsymbol{p}_{N_a}^0\}$ as a set of trainable attribute prototypes, which are randomly initialized and shared across all images. With $\mathcal{P}^0$, PAL first localizes related local regions for each prototype $\boldsymbol{p}_i^0$, and then calculates specific visual features for all $\{\boldsymbol{p}_1^0, \cdots, \boldsymbol{p}_{N_a}^0\}$ by:

$$\mathcal{P}^1 = f_{ar}(XS), S = \hbar(X^{\mathrm{T}}\mathcal{P}^0), \tag{4}$$

where $S \in \mathbb{R}^{N \times N_a}$ is a similarity matrix, and $S_{n,i}$ measures the similarity between the n-th local feature X_n and the i-th attribute prototype $\boldsymbol{p}_i^0$. $\hbar(\cdot)$ is a Softmax normalization along each column. With S, Eq. (4) aggregates related region features in X to calculate attribute-specific features, which produces $\mathcal{P}^1 = \{\boldsymbol{p}_1^1, \cdots, \boldsymbol{p}_{N_a}^1\}$. $f_{ar}(\cdot)$ is a prototype refinement function implemented by two fully-connected (FC) layers as shown in Fig. 2. Compared to the original image feature $X \in \mathbb{R}^{C \times N}$, $\mathcal{P}^1 \in \mathbb{R}^{C \times N_a}$ explicitly captures specific visual patterns of the target image for each attribute, *e.g.,* $\boldsymbol{p}_i^1$ aggregates related local features in X that correspond to the i-th attribute.

Considering instance variance, the shared attribute prototypes $\mathcal{P}^0$ cannot localize accurate attribute-related regions for all images. In Eq. (4), compared to $\mathcal{P}^0$, the visual patterns of $\mathcal{P}^1$, that correspond to different attributes, are more specific to the given image feature X. Thus, PAL further regards $\mathcal{P}^1$ as updated attribute prototypes from $\mathcal{P}^0$ for the target image. By replacing $\mathcal{P}^k$ with $\mathcal{P}^{k+1}$ and repeating Eq. (4), PAL can progressively update attribute prototypes for an input image by:

$$\mathcal{P}^{k+1} = f_{ar}\big(X\hbar(X^{\mathrm{T}}\mathcal{P}^k)\big), \tag{5}$$

where $\mathcal{P}^{k+1}$ leads to better attribute localization than $\mathcal{P}^k$.

For the k-th iteration, since $\mathcal{P}^k \in \mathbb{R}^{C\times N_a}$ contains specific visual patterns for different attributes, PAL concatenates all $\{\boldsymbol{p}_0^k, \cdots, \boldsymbol{p}_{N_a}^k\}$ to produce the visual representation $f^k(X)$:

$$f^k(X) = cat\big(f_{rd}(\mathcal{P}^k)\big), \tag{6}$$

where $f_{rd}(\cdot)$ is a dimension reduction layer and projects each $\boldsymbol{p}_i^k \in \mathbb{R}^{C\times 1}$ into $\mathbb{R}^{D\times 1}$, where $D < C$, to avoid excess calculation complexity. $cat(\cdot)$ concatenates all elements of the input. $f^k(X) \in \mathbb{R}^{N_v\times 1}$, where $N_v = D \times N_a$. With multi-iterative $\{\mathcal{P}^k | k = 1, \cdots, K\}$, PAL generates K visual representations $\{f^k(X)|k = 1, \cdots, K\}$.

Finally, $f^k(X)$ is aligned with corresponding attributes in a joint embedding space by:

$$\mathcal{L}_{sa} = \sum_{X\in\mathcal{X}_s} \sum_{k=1}^{K} d\big(f^k(X), g(\boldsymbol{a}_y)\big), \tag{7}$$

where $g(\cdot)$ is a semantic projection function implemented by FC for redundancy reduction, following [31]. In $\mathcal{L}_{sa}$, the semantic alignment supervision is applied to all $\{f^k(X)|k = 1, \cdots, K\}$ for training acceleration. As k increases appropriately, $f^k(X)$ localizes attributes more accurately. Thus, $f^K(X)$ is used as the final visual representation for inference.

Consequently, by dynamically adjusting attribute prototypes according to each individual image, PAL can progressively improve the attribute localization ability of visual representations, as shown in Fig. 3. By modelling the attribute-region correspondence, PAL narrows the semantic-visual gap between category attributes and visual representations, thereby boosting knowledge transfer between seen and unseen domains.

3.3 Progressive Category Classification

Besides exploring the correspondence between attributes and local image regions via PAL, we design a Progressive Category Classification (PCC) module to repel visual representations from different categories, which can enlarge category margins.

Similar to the shared attribute prototypes in PAL, PCC defines a set of learnable category prototypes $\mathcal{C}^0 = \{\boldsymbol{c}_1^0, \cdots, \boldsymbol{c}_{N_c}^0\}$, where $\boldsymbol{c}_j^0 \in \mathbb{R}^{N_v\times 1}$ records the visual representation center for the j-th category. $N_c = |\mathcal{Y}_s|$ is the number of seen categories. Since the attribute prototypes are progressively updated in PAL and K visual representations $\{f^k(X)|k = 1, \cdots, K\}$ are accordingly generated for each image, shared category prototypes $\mathcal{C}^0$ cannot well model visual category differences for all iterations of $f^k(X)$. Thus, PCC is similarly designed to adjust category prototypes for different $f^k(X)$ by:

$$\mathcal{C}^{k+1} = f_{cs}(\mathcal{C}^k) + W^k, \tag{8}$$

where $f_{cs}(\cdot)$ is a selection function as shown in Fig. 2. Since $f^{k+1}(X)$ derives from $f^k(X)$, the category center at the $(k+1)$-th iteration should not deviate from that of the k-th iteration. Thus, $f_{cs}(\cdot)$ actually serves as a gating function implemented by channel attention mechanism, which controls the information flow from $\mathcal{C}^k$ to $\mathcal{C}^{k+1}$. This can ease the training difficulty of PCC by avoiding repetitive learning for $\mathcal{C}^{k+1}$. Beside, W^k is a learnable bias at the k-th iteration, which supplements some specific information for $\mathcal{C}^{k+1}$.

With the visual representation $f^k(X)$ and category prototypes $\mathcal{C}^k = \{\boldsymbol{c}_1^k, \cdots, \boldsymbol{c}_{N_c}^k\}$, PCC uses a contrastive loss to repel different categories by:

$$\mathcal{L}_{cl} = -\sum_{X\in\mathcal{X}_s} \sum_{k=1}^{K} \log \frac{\exp(f^k(X)^{\mathrm{T}} \boldsymbol{c}_y^k)}{\sum_{j\in\mathcal{Y}_s} \exp(f^k(X)^{\mathrm{T}} \boldsymbol{c}_j^k)}. \tag{9}$$

Similar to $\mathcal{L}_{sa}$, the multi-stage supervision is used in $\mathcal{L}_{cl}$. Compared to $\mathcal{L}_{v2s}$ in Eq. (1), $\mathcal{L}_{cl}$ can better repel visual representations from different categories via progressive category prototypes $\{\mathcal{C}^k|k = 1, \cdots, K\}$.

With progressively-updated category prototypes, PCC improves the category discriminability of visual representation $\{f^k(X)|k = 1, \cdots, K\}$.

3.4 Overall Objective

Overall, the objective loss function of DPPN is:

$$\mathcal{L}_{all} = \mathcal{L}_{sa} + \lambda \mathcal{L}_{cl}, \tag{10}$$

where λ is the hyper-parameter to balance $\mathcal{L}_{cl}$. The attribute prototypes and category prototypes are collaboratively trained in a unified framework, which enables the final visual representation $f^K(X)$ to simultaneously capture attribute-region correspondence and category discrimination. During inference, only visual representation $f^K(X)$ at the K-th iteration is used by:

$$\hat{y} = \arg \min_{y \in \mathcal{Y}_s \cup \mathcal{Y}_u} d(f^K(X), g(\boldsymbol{a}_y)). \tag{11}$$

3.5 Discussion

Compared to APN [47], DPPN is a much different and novel method. First, instead of using shared prototypes, DPPN progressively adjusts attribute prototypes according to different images via the novel PAL. Second, corresponding to multi-iterative attribute prototypes, DPPN further constructs progressive category prototypes via the PCC to boost category discriminability. Third, the attribute localization directly involves in the final representation generation in DPPN, instead of a mere supervision during training in APN.

4 Experiments

4.1 Experimental Settings

Datasets. Four public GZSL benchmarks, *i.e.*, Caltech-USCD Birds-200-2011 (CUB) [43], SUN [35], Animals with Attributes2 (AWA2) [24], and Attribute Pascal and Yahoo (APY) [15], are adopted in this paper. CUB contains $11,788$ bird images in 200 species with 312 description attributes. SUN contains $14,340$ scene images in 717 classes with 102 attributes. AWA2 contains $37,322$ animal images in 50 classes with 85 attributes. aPY contains $15,339$ object images in 32 categories with 64 attributes.

Evaluation Metrics. The widely-used harmonic mean $H = (2MCA_u \times MCA_s)/(MCA_u + MCA_s)$ is used to evaluate GZSL performance. MCA_s and MCA_u are the Mean Class Top-1 Accuracy for seen and unseen domains, respectively.

Implementation Details. The input image is resized to 448×448 following [55, 45]. Random cropping and flipping are used for data augmentation. ResNet-101 [19] pretrained on ImageNet [13] is used as the backbone. A two-stage training strategy is adopted, which trains DPPN with the fixed backbone and then fine-tunes the whole network on two 1080ti GPUs. Adam optimizer is used with batch size of 64 and $lr = 2e-4$. $C = 512$. N_a is the number of attributes, which are 312, 85, 64, 102 for respective CUB, AWA2, aPY, and SUN. $K = 3$ and $\lambda = 0.1$. N_v will be discussed in the ablation study.

4.2 Ablation Study

Analysis of Attribute Localization and Category Discrimination. The core motivation of DPPN is to learn visual representations that simultaneously explore category discrimination and attribute-region correspondence via the proposed PCC and PAL, thus we analyse how PCC and PAL affect GZSL performance in this part. For simple comparison, we set $K = 1$ in PCC and PAL. Results are listed in Table 1. "Base-V2S" is the baseline method trained by Eq. (1), which is trained by $\mathcal{L}_{v2s}$. "+PCC" adds PCC module to Base-V2S which introduces stronger category discrimination constraint $\mathcal{L}_{cl}$. We can observe that "+PCC" obtains 1.1% and 4.7% gains on H over "Base-V2S" on CUB and aPY. This is because that PCC explicitly pushes representations away from different categories via category prototypes, thus enlarging category margins for more accurate category classification and boosting GZSL. The third model "+PAL" replaces $f(\cdot)$ in "Base-V2S" with PAL, which enables visual representations with attribute localization ability. Compared with "Base-V2S", PAL module brings 4.9% and 6.3% gains on CUB and aPY. This derives from that PAL captures attribute-region correspondence by utilizing attribute prototypes to localize attribute-related local regions and produce

Method	$\mathcal{L}_{v2s}$	$\mathcal{L}_{cl}$	$\mathcal{L}_{sa}$	K	CUB			aPY			GFLOPs
					MCA_u	MCA_s	H	MCA_u	MCA_s	H	
Base-V2S	✓			-	50.5	84.4	63.2	30.4	42.6	35.5	62.396
+PCC		✓		1	59.3	70.2	64.3	33.2	50.9	40.2	62.440
+PAL			✓	1	64.7	71.8	68.1	34.2	53.8	41.8	62.809
+PCC&PAL		✓	✓	1	69.2	71.4	70.3	35.6	59.0	44.4	62.842
+PCC&PAL		✓	✓	3	70.2	77.1	73.5	40.0	61.2	48.4	62.900

Table 1: Effect of PCC and PAL on CUB and aPY datasets. GFLOPs is calculated with input size 448×448 on the CUB dataset.

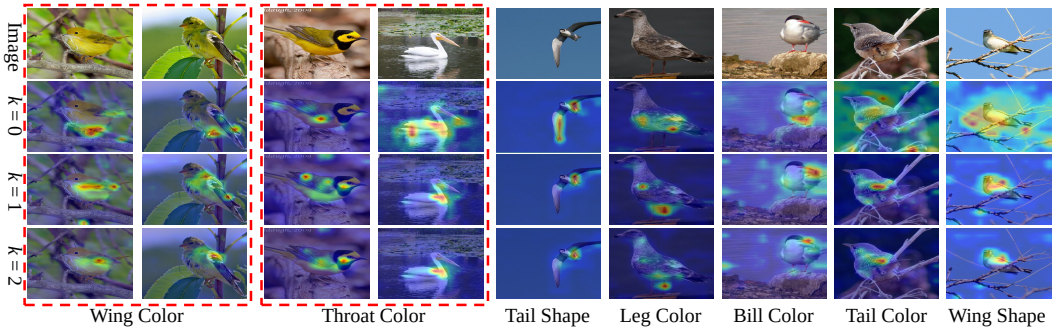


Figure 3: Visualization of attribute localization at different iterations. The localization gets more and more accurate as k increases from 0 to 2.

attribute-specific visual representations. Finally, "+PCC&PAL" simultaneously considers category discrimination and attribute localization by incorporating both PCC and PAL, which obtains 7.1% and 8.9% gains. This proves that both attribute-region correspondence and category discrimination are critical to GZSL and complementary to each other.

Notably, PAL and PCC bring negligible additional computation, even with $K = 3$, because $f_{rd}(\cdot)$ in Eq. (6) controls the dimension of visual representation $f^k(X)$ to limit the computation burden. We visualize the difference of representation distribution between Base-V2S and our DPPN, which is shown in the Appendix.

Effect of Progressive Prototype Updating. DPPN progressively updates attribute and category prototypes to learn more transferable and distinctive representations. Here, we analyse how such a progressive learning strategy impacts the attribute localization ability and category recognition by evaluating varying K in PAL and PCC. The results are given in Fig. 4.

As K rises from 1 to 3, H on all the four datasets gradually increases. The best $H = 73.5\%$, 73.1%, 48.4%, and 41.0% on CUB, AWA2, aPY, and SUN is obtained when $K = 3$. This demonstrates that, with updating category and attribute prototypes, the visual representations from DPPN become more discriminative and transferable to the unseen categories. Here, to intuitively present the attribute localization progressively learned by PAL, we visualize the attribute localization results of PAL at different iteration k when setting $K = 3$. As shown in Fig. 3, the localization gets more and more precise as attribute prototypes gradually update. With progressively updated attribute prototypes, the PAL module can finally accurately localize corresponding attribute-related visual regions. Besides, as updating, the prototype for the same attribute gets more specific to the target images in the first four columns, reflecting that progressive updating can adapt prototypes according to different images. This proves that attribute prototypes can capture the attribute-region correspondence, and progressive updating makes prototypes more specific and distinctive. When $K > 3$, H drops. The reason may be that the over-updated prototypes become unstable and hard to train. Thus, we set $K = 3$ for the rest experiments.

In summary, both quantitative and qualitative results demonstrate that progressive updating can improve attribute and category prototypes, which captures better attribute-region correspondence and category discrimination for different images.

Evaluation of Components of PAL. PAL aims to learn visual representations with accurate attribute localization. Thus, we evaluate two important components of PAL, *i.e.*, $f_{ar}(\cdot)$ in Eq. (5) and $cat(\cdot)$

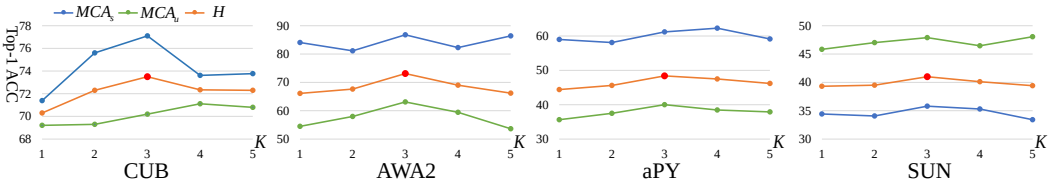


Figure 4: Effect of progressive updating with varying K on four datasets.

	PAL			PCC	CUB			aPY		
	$f_{ar}(\cdot)$	$cat(\cdot)$	$sum(\cdot)$	$f_{cs}(\cdot)$	MCA_u	MCA_s	H	MCA_u	MCA_s	H
DPPN	✓	✓		✓	70.2	77.1	73.5	40.0	61.2	48.4
		✓		✓	67.4	78.2	72.4	36.3	60.9	45.5
	✓		✓	✓	67.0	70.5	68.7	35.6	62.0	45.2
	✓	✓			68.3	76.0	71.9	38.4	57.1	45.9

Table 2: Evaluation of components in DPPN.

in Eq. (6). The results are given in Table 2. $f_{ar}(\cdot)$ is a refinement function between $\mathcal{P}^k$ and $\mathcal{P}^{k+1}$ to improve prototype quality. As shown in the second row of Table 2, without $f_{ar}(\cdot)$, H drops by 1.1% and 2.9% on CUB and aPY, respectively. This demonstrates that $f_{ar}(\cdot)$ benefits attribute prototype updating, thereby boosting attribute localization ability.

$cat(\cdot)$ is used to produce visual representation $f^k(X)$ by concatenating attribute prototypes $\mathcal{P}^k = \{\boldsymbol{p}_1^k, \cdots, \boldsymbol{p}_{N_a}^k\}$. Compared to summing $\{\boldsymbol{p}_1^k, \cdots, \boldsymbol{p}_{N_a}^k\}$ up, $cat(\cdot)$ better preserves attribute-region correspondence. The third row of Table 2 shows that H drops by 4.8% and 3.2% on CUB and aPY as replacing $cat(\cdot)$ with summing up operation. This proves that $cat(\cdot)$ benefits local correspondence preservation.

Effect of $f_{cs}(\cdot)$ in PCC. $f_{cs}(\cdot)$ in Eq. (8) serves as a gating function to ease the training of category prototypes. In this part, we analyse the impact of $f_{cs}(\cdot)$. In Eq. (8), $f_{cs}(\cdot)$ removes redundancy in $\mathcal{C}^k$. Without $f_{cs}(\cdot)$, the category prototype updating becomes $\mathcal{C}^{k+1} = \mathcal{C}^k + W^{k+1}$, which passes all information in $\mathcal{C}^k$ to $\mathcal{C}^{k+1}$. The results are listed in the last row of Table 2. Without $f_{cs}(\cdot)$, H drops by 1.6% and 2.5% on CUB and aPY. This proves that $f_{cs}(\cdot)$ helps to ease the training process of PCC, which can learn more discriminative visual representations.

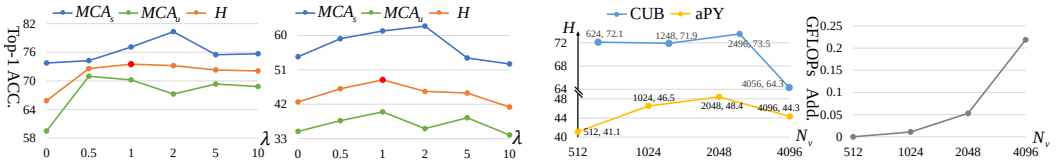
Effect of λ. λ is the hyper-parameter to balance $\mathcal{L}_{cl}$. Here, we evaluate the effect of λ as shown in Fig. 5 (a) and (b). As λ rises from 0.0 to 1.0, *i.e.*, category discrimination supervision $\mathcal{L}_{cl}$ is introduced, H increases on both CUB and aPY. The best H is obtained when $\lambda = 1.0$. This proves the effectiveness of category discrimination brought by PCC. When $\lambda > 1.0$, H starts to drop. Thus, we set $\lambda = 1.0$ for better results.

Effect of N_v. N_v is the dimension of $f^k(X)$ and $N_v = D \times N_a$. N_a is the number of attributes, which are 312, 85, 64, 102 for CUB, AWA2, aPY, and SUN, respectively. Here, we set different D to evaluate the effect of N_v on recognition performance and calculation addition. Fig. 5 (c) shows the values of H as N_v varies. Fig. 5 (d) shows GLOPs addition over $N_v = 512$. When N_v is around $2,048$, best performance is obtained with a relatively small calculation complexity addition. Thus, we set $N_v = 2,496$, $2,125$, $2,048$, $2,142$ and $D = 8$, 25, 32, and 21 for CUB, AWA2, aPY, and SUN, respectively.

4.3 Comparison with State-of-the-Art Methods

We compare DPPN with the state-of-the-art GZSL methods, of which the results are given in Table 3.

Among the existing methods, APN [47] is the most related method which also utilizes visual prototypes to localize visual parts. Different from APN that counts on only shared prototypes for all images, our DPPN adjusts attribute prototypes dynamically according to different images and exploits category prototypes to enhance category discrimination. Thus, DPPN learns specific and distinctive visual representations and surpasses APN by a large margin, *i.e.*, 6.3%, 7.6%, and 3.4% for H on CUB, AWA2, and SUN datasets, respectively. APN does not provide codes and is approximately similar to PAL with $K = 1$. As shown in Fig. 3, it qualitatively implies that progressive updating attribute prototypes can learn visual representations with better attribute localization ability. Besides,



(a) Top-1 ACC. - λ on CUB (b) Top-1 ACC. - λ on aPY (c) H - N_v on CUB and aPY (d) GFLOPs-Add. - N_v

Figure 5: Effect of hyper-parameters λ and N_v.

	Methods	CUB			AWA2			aPY			SUN		
		MCA_u	MCA_s	H	MCA_u	MCA_s	H	MCA_u	MCA_s	H	MCA_u	MCA_s	H
GEN.	IZF-Softmax[39]	52.7	68.0	59.4	60.6	77.5	68.0	42.3	60.5	49.8	52.7	57.0	54.8
	TF-VAEGAN[34]	63.8	79.3	70.7	-	-	-	-	-	-	62.4	47.1	53.7
	E-PGN[50]	52.0	61.1	56.2	52.6	83.5	64.6	-	-	-	-	-	-
	GCM-CF[51]	61.0	59.7	60.3	60.4	75.1	67.0	37.1	56.8	44.9	47.9	37.8	42.2
	CE-GZSL[18]	63.9	66.8	65.3	63.1	78.6	70.0	-	-	-	48.8	38.6	43.1
EMB.	MLSE[14]	22.3	71.6	34.0	23.8	83.2	37.0	12.7	74.3	21.7	20.7	36.4	26.4
	COSMO[4]	44.4	57.8	50.2	-	-	-	-	-	-	44.9	37.7	**41.0**
	PREN[48]	32.5	55.8	43.1	32.4	88.6	47.4	-	-	-	35.4	27.2	30.8
	VSE-S[55]	33.4	87.5	48.4	41.6	91.3	57.2	24.5	72.0	36.6	-	-	-
	LFGAA[29]	43.4	79.6	56.2	50.0	90.3	64.4	-	-	-	20.8	34.9	26.1
	AREN[45]	63.2	69.0	66.0	54.7	79.1	64.7	30.0	47.9	36.9	40.3	32.3	35.9
	CosineSoftmax[25]	47.4	47.6	47.5	56.4	81.4	66.7	26.5	74.0	39.0	36.3	42.8	39.3
	RGEN[46]	73.5	60.0	66.1	76.5	67.1	71.5	48.1	30.4	37.2	31.7	44.0	36.8
	DAZLE[21]	56.7	59.6	58.1	60.3	75.7	67.1	-	-	-	52.3	24.3	33.2
	APN[47]	65.3	69.3	67.2	56.5	78.0	65.5	-	-	-	41.9	34.0	37.6
	GEM-ZSL[30]	64.8	77.1	70.4	64.8	77.5	70.6	-	-	-	38.1	35.7	36.9
	DPPN	70.2	77.1	**73.5**	63.1	86.8	**73.1**	40.0	61.2	**48.4**	47.9	35.8	**41.0**

Table 3: Results of GZSL on four classification benchmarks. Our DPPN belongs to embedding-based methods (EMB.). Generative methods (GEN.) utilize extra synthetic unseen domain data for training. The best result is bolded, and the second best is underlined.

DAZLE [21] uses dense attribute attention to focus on relevant regions, which is inferior than DPPN by 15.4%, 6.0%, and 7.8% on CUB, AWA2, and SUN, respectively. This proves the effectiveness of designing progressive attribute and category prototypes in GZSL.

Comparing to other embedding-based methods, our method surpasses the best one by respectively 3.1%, 1.6%, and 9.4% for H on CUB, AWA2, aPY datasets, and obtains comparable best H performance on SUN dataset. The results demonstrate that progressively exploring attribute-region correspondence and category discrimination can effectively enhance cross-domain transferability and category discriminability of visual representations. The reason for relative small improvement on SUN may be that #categories is large while #images in each category is small in SUN, leading to difficulties for DPPN to learn accurate attribute localization and category discrimination.

Comparing to generative methods that utilize additional unseen category labels during training, DPPN can achieve comparable, even better results, especially on CUB and AWA2 datasets. This reveals that with the assistance of progressive attribute localization and category discrimination, DPPN is able to surpass generative methods without complex GAN training.

5 Conclusion

In this paper, we propose a Dual Progressive Prototype Network (DPPN) to progressively explore both attribute-region correspondence and category discrimination for GZSL. Specifically, DPPN constructs progressive prototypes for both attributes and categories. Target to different images, DPPN alternatively locates attribute-related visual regions and updates attribute prototypes, which improves attribute localization ability and cross-domain transferability of visual representations. According to progressive attribute prototypes, DPPN progressively projects category prototypes to multiple spaces to enforce visual representations away from different categories, thus enhancing category discriminability. Extensive experimental results on four public datasets demonstrate the effectiveness of our DPPN.

References

[1] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(7), 1425–1438 (2016)

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

[3] Annadani, Y., Biswas, S.: Preserving semantic relations for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7603–7612 (2018)

[4] Atzmon, Y., Chechik, G.: Adaptive confidence smoothing for generalized zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11671–11680 (2019)

[5] Ba, J.L., Swersky, K., Fidler, S., salakhutdinov, R.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4247–4255 (2015)

[6] Bucher, M., Herbin, S., Jurie, F.: Improving semantic embedding consistency by metric learning for zero-shot classiffication. In: European Conference on Computer Vision. pp. 730–746 (2016)

[7] Cacheux, Y.L., Borgne, H.L., Crucianu, M.: Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10333–10342 (2019)

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

[9] Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: Proceedings of the European Conference on Computer Vision. pp. 52–68 (2016)

[10] Chen, L., Zhang, H., Xiao, J., Liu, W., Chang, S.F.: Zero-shot visual recognition using semantics-preserving adversarial embedding network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1043–1052 (2018)

[11] Chen, X., Lan, X., Sun, F., Zheng, N.: A boundary based out-of-distribution classifier for generalized zero-shot learning. In: Proceedings of the European Conference on Computer Vision (2020)

[12] Chen, Y., Bai, Y., Zhang, W., Mei, T.: Destruction and construction learning for fine-grained image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5157–5166 (2019)

[13] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)

[14] Ding, Z., Liu, H.: Marginalized latent semantic encoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6191–6199 (2019)

[15] Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1785 (2009)

[16] Ferrari, V., Zisserman, A.: Learning visual attributes. Advances in neural information processing systems **20**, 433–440 (2007)

[17] Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al.: Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems. pp. 2121–2129 (2013)

[18] Han, Z., Fu, Z., Chen, S., Yang, J.: Counterfactual zero-shot and open-set visual recognition. In: CVPR (2021)

[19] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[20] Huang, H., Wang, C., Yu, P.S., Wang, C.D.: Generative dual adversarial network for generalized zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 801–810 (2019)

[21] Huynh, D., Elhamifar, E.: Fine-grained generalized zero-shot learning via dense attribute-based attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4483–4493 (2020)

[22] Jiang, H., Wang, R., Shan, S., Chen, X.: Transferable contrastive network for generalized zero-shot learning. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[23] Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3174–3183 (2017)

[24] Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 951–958 (2009)

[25] Li, K., Min, M.R., Fu, Y.: Rethinking zero-shot learning: A conditional visual classification perspective. In: The IEEE International Conference on Computer Vision (October 2019)

[26] Li, Y., Wang, D., Hu, H., Lin, Y., Zhuang, Y.: Zero-shot recognition using dual visual-semantic mapping paths. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3279–3287 (2017)

[27] Liu, L., Zhou, T., Long, G., Jiang, J., Dong, X., Zhang, C.: Isometric propagation network for generalized zero-shot learning. In: International Conference on Learning Representations (2021)

[28] Liu, S., Long, M., Wang, J., Jordan, M.I.: Generalized zero-shot learning with deep calibration network. In: Advances in Neural Information Processing Systems. pp. 2005–2015 (2018)

[29] Liu, Y., Guo, J., Cai, D., He, X.: Attribute attention for semantic disambiguation in zero-shot learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6698–6707 (2019)

[30] Liu, Y., Zhou, L., Bai, X., Huang, Y., Gu, L., Zhou, J., Harada, T.: Goal-oriented gaze estimation for zero-shot learning. In: CVPR (2021)

[31] Min, S., Yao, H., Xie, H., Wang, C., Zha, Z.J., Zhang, Y.: Domain-aware visual bias eliminating for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

[32] Min, S., Yao, H., Xie, H., Zha, Z.J., Zhang, Y.: Domain-specific embedding network for zero-shot recognition. In: Proceedings of ACM International Conference on Multimedia. pp. 2070–2078 (2019)

[33] Min, S., Yao, H., Xie, H., Zha, Z.J., Zhang, Y.: Domain-oriented semantic embedding for zero-shot learning. IEEE Transactions on Multimedia (2020)

[34] Narayan, S., Gupta, A., Khan, F.S., Snoek, C.G., Shao, L.: Latent embedding feedback and discriminative features for zero-shot classification. In: Proceedings of the European Conference on Computer Vision. pp. 1–23 (2020)

[35] Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2751–2758 (2012)

[36] Rahman, S., Khan, S., Porikli, F.: A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Transactions on Image Processing **27**, 5652–5667 (2018)

[37] Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 49–58 (2016)

[38] Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning. pp. 2152–2161 (2015)

[39] Shen, Y., Qin, J., Huang, L.: Invertible zero-shot recognition flows. In: Proceedings of the European Conference on Computer Vision. pp. 614–631 (2020)

[40] Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., Matsumoto, Y.: Ridge regression, hubness, and zero-shot learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 135–151 (2015)

[41] Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: Advances in neural information processing systems. pp. 935–943 (2013)

[42] Tong, B., Wang, C., Klinkigt, M., Kobayashi, Y., Nonaka, Y.: Hierarchical disentanglement of discriminative latent features for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11467–11476 (2019)

[43] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. In: California Institute of Technology. pp. 1–8 (2011)

[44] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 69–77 (2016)

[45] Xie, G.S., Liu, L., Jin, X., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9384–9393 (2019)

[46] Xie, G.S., Liu, L., Zhu, F., Zhao, F., Zhang, Z., Yao, Y., Qin, J., Shao, L.: Region graph embedding network for zero-shot learning. In: Proceedings of the European Conference on Computer Vision (2020)

[47] Xu, W., Xian, Y., Wang, J., Schiele, B., Akata, Z.: Attribute prototype network for zero-shot learning. In: Advances in Neural Information Processing Systems. pp. 21969–21980 (2020)

[48] Ye, M., Guo, Y.: Progressive ensemble networks for zero-shot recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11728–11736 (2019)

[49] Yu, Y., Ji, Z., Fu, Y., Guo, J., Pang, Y., Zhang, Z.: Stacked semantic-guided attention model for fine-grained zero-shot learning. arXiv preprint arXiv:1805.08113 (2018)

[50] Yu, Y., Ji, Z., Han, J., Zhang, Z.: Episode-based prototype generating network for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14035–14044 (2020)

[51] Yue, Z., Wang, T., Zhang, H., Sun, Q., Hua, X.S.: Counterfactual zero-shot and open-set visual recognition. In: CVPR (2021)

[52] Zhang, H., Long, Y., Guan, Y., Shao, L.: Triple verification network for generalized zero-shot learning. IEEE Transactions on Image Processing **28**(1), 506–517 (2019)

[53] Zhang, H., Koniusz, P.: Zero-shot kernel learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7670–7679 (2018)

[54] Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

[55] Zhu, P., Wang, H., Saligrama, V.: Generalized zero-shot recognition based on visually semantic embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2995–3003 (2019)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See abstract and introduction parts.
   (b) Did you describe the limitations of your work? [Yes] See Section 4.3.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Our DPPN can alleviate the burden of labelling data, which may cause unemployment of data annotators.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] I have carefully read and ensured that this paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not include theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] We do not include theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See implementation details and ablation study in Sec. 4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We did not report error bars for it would bring high computational cost.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See implementation details and ablation study in Sec. 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See datasets and implementation details in Sec. 4
   (b) Did you mention the license of the assets? [No] We were unable to find the license for the dataset we used.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We only use existing data.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] All data used in this paper are publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] All data used in this paper are publicly available and have no relations with privacy information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not use crowdsourcing or conduct research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We do not use crowdsourcing or conduct research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We do not use crowdsourcing or conduct research with human subjects.