Refinement Matters: Textual Description Needs to be Refined for Zero-shot Learning

Chandan Gautam* Institute for Infocomm Research A*STAR gautamc@i2r.a-star.edu.sg

Sethupathy Parameswaran* Indian Institute of Science sethupathyp@iisc.ac.in

Vinay Kumar Verma Duke University vinayugc@gmail.com

Suresh Sundaram Indian Institute of Science vssuresh@iisc.ac.in

Savitha Ramasamy Institute for Infocomm Research, A*STAR ramasamysa@i2r.a-star.edu.sg

Abstract

Zero-Shot Learning (ZSL) has shown great promise at the intersection of vision and language, and generative methods for ZSL are predominant owing to their efficiency. Moreover, textual description or attribute plays a critical role in transferring knowledge from the seen to unseen classes in ZSL. Such generative approaches for ZSL are very costly to train and require the class description of the unseen classes during training. In this work, we propose a non-generative gating-based attribute refinement network for ZSL, which achieves similar accuracies to generative methods of ZSL, at a much lower computational cost. The refined attributes are mapped into the visual domain through an attribute embedder, and the whole network is guided by the circle loss and the well-known softmax cross-entropy loss to obtain a robust class embedding. We refer to our approach as **C**ircle loss guided gating-based **A**ttribute-**R**efinement **Net**work *(CARNet)*. We perform extensive experiments on the five benchmark datasets over the various challenging scenarios viz., Generalized ZSL (GZSL), Continual GZSL (CGZSL), and conventional ZSL. We observe that the CARNet significantly outperforms recent non-generative ZSL methods and most generative ZSL methods in all three settings by a significant margin. Our extensive ablation study disentangles the performance of various components and justifies their importance[1].

1 Introduction

Humans can recognize samples from unseen classes by leveraging the visual information of seen categories and textual descriptions of seen and unseen classes (Larochelle et al., 2008; Palatucci et al., 2009; Lampert et al., 2009). Zero-Shot Learning, inspired by this recognition ability of humans,

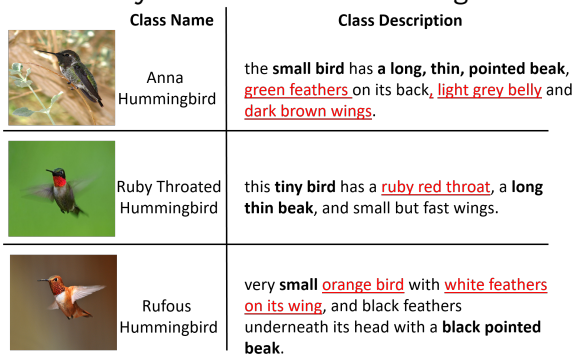


Figure 1: It can be seen that the attributes "small bird" and "long black beak" (words in bold) are common in all three species of the Hummingbird. However, attributes like "ruby red throat" or "orange bird" (words in red) distinguish one species from another. Hence the distinguishing attributes must be given more weight than the common attributes in the class attribute vector. We achieve this through the gating unit in the attribute refinement network.

learns unseen classes through the textual description (also referred to as side-information or class attribute vector or semantic information) (Xian et al., 2017). A typical ZSL algorithm does not need training samples from unseen classes. However, it requires the class description for both seen and unseen classes (Zhang and Saligrama, 2015; Reed et al., 2016).

The generative model has recently been the most popular approach for ZSL. It uses generators like VAE (Mishra et al., 2018; Schonfeld et al., 2019) or GAN (Narayan et al., 2020; Vyas et al., 2020) to generate synthetic samples for unseen classes using the class attribute vector. However, despite their promising results, such methods are not very efficient due to the following reasons: (i) The method requires the knowledge of the number of unseen classes and respective attribute vectors during training which is not always feasible, and (ii) retraining of the classifier with seen and unseen samples, with each new unseen classes. On the other hand, non-generative approaches for ZSL alleviates the

*These authors contributed equally to this work

[1]The source code is available at https://github.com/Sethup123/CARNet

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6156 - 6169
December 7-11, 2022 ©2022 Association for Computational Linguistics

above problems but shows inferior accuracies. Typically, the non-generative models learn mapping in three ways: (i) visual to attribute space (Xian et al., 2016) or (ii) attribute to visual space (Zhang et al., 2017; Li et al., 2019), or (iii) joint embedding of attribute and visual space (Cacheux et al., 2019). It is to be noted that most of the existing non-generative or embedding-based ZSL approaches are formulated to learn embedding from visual to attribute space. They assume that the seen and unseen classes share the same representational characteristics and are linked in the attribute space (Frome et al., 2013; Wang et al., 2019; Chen et al., 2021a). However, this approach leads to the well-known hubness problem (Dinu et al., 2014), where the representations are skewed towards seen classes (Zhang et al., 2017; Li et al., 2019). Another problem with this approach is that it implicitly loses the discriminative power of visual features that are generally extracted from a powerful pre-trained deep learning model (like ResNet (Xian et al., 2018a, 2016) and GoogleNet (Song et al., 2018)) but are then mapped to a different smaller attribute space (Li et al., 2019, 2018).

Although the aforementioned issues in the non-generative model are mitigated by mapping the attribute to visual features (Zhang et al., 2017; Li et al., 2019; Skorokhodov and Elhoseiny, 2021), they have lower classification accuracies. In this work, we propose a non-generative method with an Attribute-Refinement Network (ARN) that leverages the gating mechanism. The ARN enables highly robust representation of the description/attribute vector for the seen and unseen classes. In recent years, the gating mechanism has shown good performance without any complex architecture in supervised learning tasks (Srivastava et al., 2015; Dauphin et al., 2017; Sandler et al., 2018; Wu et al., 2018; Liu et al., 2021). In this paper, we propose a gating mechanism for refining the textual description in the ZSL task. The ARN learns to refine the attribute in a self-weighing manner from the seen class attribute (Fig.1). These refined attributes are mapped onto the visual space using an attribute embedder (AE) to obtain the class prototype vector of each class. The class prototype vector is then combined with the visual features in the feature-prototype combiner (FPC) to obtain classifications. The ARN, AE, and FPC are trained jointly using the circle loss and standard softmax cross-entropy in such a way that it minimizes inter-class and maximizes intra-class similarity. The circle loss achieves better within-class compactness and between-class discrepancy compared to triplet loss and adaptive margin softmax loss, as it unifies both classification and pair-wise similarity representation objectives (Sun et al., 2020). We refer to our approach as **C**ircle loss guided gating-based **A**ttribute-**R**efinement **Net**work *(CARNet)*.

We evaluate the performance of CARNet in three scenarios: (a) Conventional Zero-Shot Learning (ZSL), where only the unseen classes are available during inference (b) Generalized Zero-Shot Learning (GZSL), where both the seen and unseen classes are available during inference and (c) Continual Generalized Zero-Shot Learning (CGZSL), where data arrives as a sequence of tasks and only current task data is available during training, with the challenge of handling catastrophic forgetting of the past tasks. The performance of CARNet for conventional ZSL and GZSL is evaluated on five standard datasets. The CGZSL method is evaluated for the challenging CUB and SUN datasets. The extensive experiment shows that CARNet outperforms the recent generative (unlike the generative model, we do not require the unseen class description during training) and the non-generative model by a significant margin. Our ablation study emphasizes the significance of each component of the proposed learning algorithm. The main contributions of our work are summarised as follows: (i) We propose a gating-based attribute-refinement network (ARN) to enhance the class description/attribute for zero-shot learning. (ii) The ARN and AE are guided by circle loss to achieve better within-class compactness and between-class discrepancy. (iii) We propose a highly competitive, simple, and fast non-generative method. Our model achieves $\sim 70\times$ speedup compared to generative ZSL methods.

2 Related Work

The proposed CARNet is evaluated for the three kinds of ZSL settings: conventional ZSL, GZSL, and CGZSL. We provide a brief survey on all these three settings. ZSL aims to construct the recognition model for the samples from unseen classes using the textual description (i.e., attribute information) of unseen classes. These attribute information can be obtained through various ways, like human-annotated attributes (Farhadi et al., 2009), textual descriptions (Reed et al., 2016), and word vectors (Socher et al., 2013; Frome et al., 2013). In recent

years, there has been a surge of interest in this area. The whole literature of ZSL can be broadly categorized into two parts: generative and non-generative (i.e., embedding-based) approaches.

The first popular category is the generative approach, which solves the ZSL problem by synthesizing the unseen class samples. To synthesize the samples from the unseen classes, models leverage on a powerful generative models like conditional variational autoencoder (VAE) (Mishra et al., 2018; Kumar Verma et al., 2018) or generative adversarial network (GAN) (Vyas et al., 2020; Xian et al., 2018b; Felix et al., 2018; Keshari et al., 2020; Verma et al., 2020) or a combination of VAE and GAN (Xian et al., 2019; Narayan et al., 2020).

Another popular category is the non-generative approach, and it does not need class attribute information of unseen classes during training. In the early ZSL work (Lampert et al., 2009; Farhadi et al., 2009; Lampert et al., 2013), models directly predict the attribute confidence from images. Methods based on this approach can be further divided into three groups. In the first group, we project visual feature into attribute (i.e., semantic) space (Lampert et al., 2013; Socher et al., 2013; Frome et al., 2013; Akata et al., 2016; Fu and Sigal, 2016). In the second group, both visual and attribute data are projected into intermediate space (Akata et al., 2015; Fu et al., 2014; Lei Ba et al., 2015; Romera-Paredes and Torr, 2015; Cacheux et al., 2019). In the third group, visual space is spanned by attribute to visual mapping (Zhang et al., 2017; Li et al., 2019; Skorokhodov and Elhoseiny, 2021). ZSL methods developed based on the projection from attribute space to visual space approach are more suitable for mitigating the hubness problem, and recent works (Zhang et al., 2017; Li et al., 2019; Skorokhodov and Elhoseiny, 2021) show promising results for the ZSL and GZSL setting. Surprisingly, despite the fast, accurate, and realistic setting, this approach has not been explored much in the past. In this work, we consider the non-generative model for further exploration and learn the mapping from the attribute space to the visual space similar to (Li et al., 2019; Skorokhodov and Elhoseiny, 2021).

The above-discussed ZSL methods can handle data only in an offline setting, and cannot be used in a setting with a streaming sequence of tasks (Delange et al., 2021), known as Continual GZSL (CGZSL). Only a handful of research is available for CGZSL (Chaudhry et al., 2019a; Wei et al., 2020; Skorokhodov and Elhoseiny, 2021; Gautam et al., 2021a, 2020, 2021b). For the extensive evaluation, apart from the conventional ZSL and GZSL, CARNet is also evaluated on the CGZSL setting as proposed in Skorokhodov and Elhoseiny (2021).

3 Problem Definition

In this section, we define the problem formally and introduce the notations. The objective of ZSL is to learn a model that can generalize the novel classes (i.e., unseen classes) with the help of side information (attribute/descriptions) without training data for the novel classes. The attribute vector of each class is constructed by either using a word embedding vector generated from a language model or manually defining the key features like color, size, shape, pattern, etc. Primarily, the ZSL setting consists of two sets of classes known as seen and unseen classes. Let $\mathcal{D}^s_{tr}$ and $\mathcal{D}_{ts}$ be the training and testing data, respectively, for the C^s seen and C^u unseen classes. We also have set of seen ($\{C^s\}$) and unseen ($\{C^u\}$) classes where $\{C^s\} \cap \{C^u\} = \phi$ i.e. seen and unseen classes set are disjoint. It is to be noted that $\{C^s\}$ and $\{C^u\}$ denote the set of seen and unseen classes, while C^s and C^u denote the number of seen classes and unseen classes, respectively. Corresponding to each seen class i ($i \in \{C^s\}$) and unseen class j ($j \in \{C^u\}$), there is a d-dimensional class attribute vector, i.e., $A^s_i \in \mathbb{R}^d$ and $A^u_j \in \mathbb{R}^d$, respectively. In the ZSL training, data is represented as: $\mathcal{D}^s_{tr} = \{x_i, y_i, A^s_{y_i}\}^N_{i=1}$ where N is the number of seen class images and $\{x_i, y_i\}$ is the image and label pair. During inference for conventional ZSL, we have $\mathcal{D}_{ts} = \{x_j\}^M_{j=1}$ with attribute set $A = A^u$ where $\forall j, x_j$ belongs to the unseen class. However, in GZSL, we have $\mathcal{D}_{ts} = \{x_j\}^M_{j=1}$ with attribute set $A = A^s \cup A^u$ where $\forall j, x_j$ belongs to either seen or unseen class. Here A^s and A^u are the seen and unseen class attribute information, respectively. Overall, our objective is to develop a model based on the training dataset $\mathcal{D}_{tr}$ (i.e., seen data), and it needs to be generalized over all class labels $\{C\}$ where $\{C\} = \{C^s\} \cup \{C^u\}$ and the total number of classes in $\{C\}$ is C.

4 Proposed Method

Zero-Shot Learning (ZSL) aims at classification in the absence of input images for unseen classes, using textual description, namely attribute vectors. In this section, we propose the CARNet for zero-shot

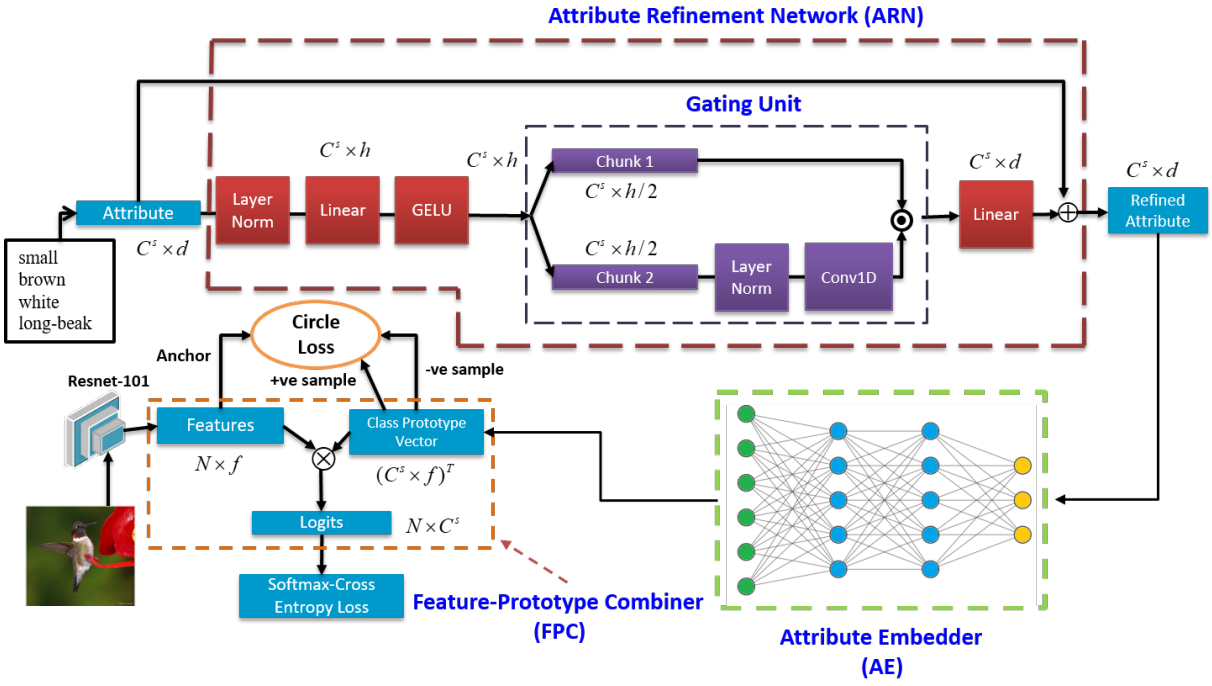


Figure 2: CARNet: circle loss guided gating-based attribute refinement network for ZSL. It primarily consists of three blocks: (i) ARN, (ii) AE, and (iii) FPC

learning. CARNet refines the class attribute vectors for better representation with the help of end-to-end joint loss. Here, a gating-based attribute refinement network (ARN) is used to refine the class attribute vectors. These refined class attribute vectors are mapped through an attribute embedder (AE) to extract an efficient class prototype vector corresponding to each class. The visual features are extracted using a pre-trained ResNet-101 model. These visual features are then combined with the class prototype vectors in the feature-prototype combiner (FPC) for classification, as shown in Fig. 2. The ARN, the AE, and the FPC are trained end-to-end based on the sum of two losses, namely, the circle loss and the softmax cross-entropy loss. These two losses guide the gating unit to yield refined attributes, which lead to better class prototype vectors through AE. In this section, we present a detailed description of our model CARNet.

4.1 Gating-based Attribute Refinement Network (ARN)

The class attribute vector plays a very crucial role in ZSL, as there are no visual samples for unseen classes during training. Moreover, the attribute vector is the only information that is available for both the seen and unseen classes. Therefore, it is highly important that the attribute representation has minimal noise and highlights its prime dimensions. The objective of the ARN is to obtain an accurately representative class attribute vector with high weight on its key dimensions, as shown in Fig. 2. Let $A^s \in \mathbb{R}^{C^s \times d}$ be the class attribute matrix for C^s seen classes where each row corresponds to the d dimensional class attribute vector of the corresponding class. The ARN consists of the following stack of operations.

We first normalize the input A^s across the dimension d for each class independently using layer-norm (Ba et al., 2016), as $A^s_L = LayerNorm(A^s)$ Over the layer-norm, we perform the linear projection followed by the Gaussian error linear unit (GELU) (Hendrycks and Gimpel, 2016) activation function as $A^s_P = GELU(A^s_L W_1)$. Here, the linear projection helps in the expansion of the current dimension of the class attribute vector. Here, $W_1 \in \mathbb{R}^{d \times h}$ denotes weight for the linear projection and $A^s_P \in \mathbb{R}^{C^s \times h}$.

Further, we apply the gating unit, which helps achieve a better representation of the attribute information. In the ARN, the gating unit performs cross-feature learning on the higher dimensional class attribute information vector (A^s_P). For this purpose, we split the A^s_P into two parts, each with half the dimension, i.e. $A^s_{P1} \in \mathbb{R}^{C^s \times h/2}$ and $A^s_{P2} \in \mathbb{R}^{C^s \times h/2}$. Both halves are processed as

follows:

$$A^s_G = A^s_{P1} \odot Conv1\text{D}(LayerNorm(A^s_{P2})), \quad (1)$$

where $Conv1\text{D}$ represents 1-D Convolution, which enables the spatial projection, $A^s_G \in \mathbb{R}^{C^s \times h/2}$, and $\odot$ represents Hadamard product (i.e., element-wise multiplication) which is a linear gating. Also, we can interpret this operation as self-weighting on each dimension of the attribute vector. The important dimensions will get high weight while the other has lower weight. The Hadamard product enables the refinement network to keep both information (i.e., raw A^s_{P1} and spatial projection of A^s_{P2}) in the output of the gating unit, i.e., A^s_G. During training, the $Conv1\text{D}(LayerNorm(A^s_{P2}))$ is initialized as an identity matrix. Finally, the output of the ARN is obtained through residual learning, as shown below:

$$A^s_R = A^s_G W_2 \oplus A^s \quad (2)$$

where $W_2 \in \mathbb{R}^{h/2 \times d}$ denotes weight for the linear projection, $\oplus$ denotes element-wise-addition, and $A^s_R \in \mathbb{R}^{C^s \times d}$ is the final refined class attribute information. The linear projection helps A^s_G to have the same dimension as A^s.

Overall, the refinement network stacks the above-mentioned operations one after the other, as shown in Fig. 2, and this set of operations can be repeated multiple times. As the set of operations can be repeated multiple times for better attribute refinement, it can cause the vanishing gradient problem that is very common in typical gating units. However, the residual learning in Eq. (2) helps to alleviate this issue.

4.2 Attribute Embedder (AE)

After getting the refined attribute for seen classes, we perform attribute to visual mapping using the AE (as shown in Fig. 2) to obtain the class prototype matrix P^s for the C^s seen classes, where each row corresponds to the class prototype vector p^s_c of the respective seen class $c \in \{C^s\}$.

$$P^s = AE(A^s_R), \;\; P^s \in \mathbb{R}^{C^s \times f}, \quad (3)$$

Overall, the AE is a simple 3-layered multi-layer perceptron (MLP) architecture, which is used to perform the attribute-to-visual mapping. Here, f denotes the dimension of the visual feature vector.

4.3 Feature-Prototype Combiner (FPC)

The visual features $V^s_{tr} \in \mathbb{R}^{N \times f}$ are extracted by passing the images of the seen classes ($\mathcal{D}^s_{tr}$) through a pretrained ResNet-101 model (no fine-tuning). These visual features are combined with the class prototype vectors in the FPC through scaled cosine similarity between the P^s and the V^s_{tr} (Skorokhodov and Elhoseiny, 2021). The scaled cosine similarity ($scos$) scales and normalizes the class prototype vectors and the extracted visual features before computing the dot product between them as follows:

$$scos(v^s_{tr}, p^s_c) = \left(\beta \cdot \frac{v^s_{tr}}{\|v^s_{tr}\|}\right)^\top \left(\beta \cdot \frac{p^s_c}{\|p^s_c\|}\right), \quad (4)$$

where $v^s_{tr} \in V^s_{tr}$ is the f-dimensional extracted visual feature for a sample, p^s_c is the class prototype vector of class $c \in \{C^s\}$, β is the scaling hyper-parameter, which has the same impact as setting a high temperature of β^2 in softmax (Liu et al., 2018). Here, normalization reduces the variance of the class prototype vectors and the visual features, which helps in achieving better performance.

4.4 Training of the CARNet using only Seen Classes

The CARNet is trained by minimizing the circle loss and the softmax cross-entropy loss over the end-to-end network comprising the ARN and the AE. We present these loss functions and the learning algorithm of CARNet in this subsection. Without loss of generality, let us assume that $v^s_{tr} \in V^s_{tr}$ be the extracted visual feature of a sample, which belongs to the seen class $k \in \{C^s\}$.

Circle Loss: Generally, two kinds of losses are involved in the literature: one kind of losses, like L2-softmax, AM-softmax, and angular softmax are good candidates for classification, while the other kind of losses, like triplet loss, N-pair loss, contrastive loss, and the margin loss are good candidates for pair-wise similarity. The circle loss (Sun et al., 2020) aims to unify both classification and pair-wise similarity representation. Hence, it is a good candidate for optimizing ARN, AE, and FPC. Moreover, it enhances the feature learning and better separability by using flexible optimization and definite convergence target (Sun et al., 2020). The main objective of feature learning is to increase within-class similarity s_p while reducing between-class similarity s_n. The circle loss unifies both

class-level labels and pair-wise similarity with K within-class similarity scores (s_p) and L between-class similarity scores (s_n) and is defined as:

$$\mathcal{L}_{circle} = \log\Big[1 + \sum_{j=1}^{L} \exp(\gamma \alpha_n^j (s_n^j - \Delta_n)) \sum_{i=1}^{K} \exp(-\gamma \alpha_p^i (s_p^i - \Delta_p))\Big], \tag{5}$$

where $\alpha_p^i = [1 + m - s_p^i]_+$, and $\alpha_n^j = [s_n^j + m]_+$ are weighting factors, $\alpha_n^j > 0$, $\alpha_p^i > 0$, and γ is a scaling factor. Here $[.]_+$ denotes cut-off at Zero. $\Delta_n = m$ and $\Delta_p = 1 - m$ are the between-class and within-class margins, respectively.

In CARNet, we use the sample v_{tr}^s as the anchor and the corresponding class prototype vector p_k^s of class k as the positive sample and the remaining class prototype vectors p_j^s of the seen classes as negative samples. The cosine similarity is used to determine the positive similarity s_p and negative similarity s_n as follows:

$$s_p^k = \frac{v_{tr}^s . p_k^s}{\|v_{tr}^s\| \|p_k^s\|} \tag{6}$$

$$s_n^j = \frac{v_{tr}^s . p_j^s}{\|v_{tr}^s\| \|p_j^s\|}, \quad \text{where } k, j \in \{C^s\} \text{ and } k \neq j \tag{7}$$

Hence, the circle loss in Eq. (5) is modified in CARNet as:

$$\mathcal{L}_{circle} = \log\Big[1 + \exp(-\gamma \alpha_p^k (s_p^k - \Delta_p)) \sum_{\substack{j \in \{C^s\} \\ k \neq j}} \exp(\gamma \alpha_n^j (s_n^j - \Delta_n))\Big] \tag{8}$$

Softmax Cross-Entropy Loss: To improve the classification, the softmax cross-entropy loss ($\mathcal{L}_{soft-ce}$) is applied over the computed scaled cosine similarity in Eq. (4), as shown below:

$$\mathcal{L}_{soft-ce} = -\log \frac{e^{scos(v_{tr}^s, p_k^s)}}{\sum_{i \in \{C_s\}} e^{scos(v_{tr}^s, p_i^s)}} \tag{9}$$

Thus, the training of the CARNet is achieved by learning the weights of the ARN and the AE using the losses in Eq. (8) and Eq. (9), as shown below:

$$\mathcal{L}_{CARNet} = \mathcal{L}_{soft-ce} + \lambda \mathcal{L}_{circle} \tag{10}$$

Thus, the loss in Eq. (10) is optimized during training. It is to be noted that only the seen class information ($\mathcal{D}_{tr}^s, \{C^s\}, A^s$) is used during training.

4.5 Inference: Seen and Unseen Classes

The proposed CARNet method is based on the fixed body and dynamic head (classification layer) architecture. As the model is trained with only seen classes, the classification layer has neurons corresponding to the seen classes only, i.e., C^s neurons. Further, we simply modify the output head and enable it for unseen classes ($\{C^u\}$) using its class attribute information (A^u) as per the following procedure:

1. Pass the unseen class attribute information A^u to the trained ARN and get the output A_R^u.
2. Pass the A_R^u to the trained AE and get the unseen class prototype vectors $P^u \in \mathbb{R}^{C^u \times f}$ for C^u unseen classes.
3. The unseen class prototype vectors (P^u) are stacked with seen class prototype vectors (P^s) as follows:

$$P = \begin{bmatrix} P^s \\ P^u \end{bmatrix}, \tag{11}$$

where $P \in \mathbb{R}^{C \times f}$.

After computation of P, we compute the scaled cosine similarity score $scos(v_{ts}, p_i)$ as follows:

$$scos(v_{ts}, p_i) = \left(\beta \cdot \frac{v_{ts}}{\|v_{ts}\|}\right)^\top \left(\beta \cdot \frac{p_i}{\|p_i\|}\right), \tag{12}$$

where $p_i \in P$ is the class prototype vector of class $i \in \{C^s\} \cup \{C^u\}$, and $v_{ts} \in V_{ts}^s \cup V_{ts}^u$. Here, V_{ts}^s and V_{ts}^u are the extracted visual features using pretrained Resnet-101 model for the test images of seen and unseen classes, respectively. Finally, we perform a traditional way of classification and choose the class based on the highest cosine similarity score.

5 Experiments

We conduct extensive experiments on five benchmark ZSL datasets (description given in Table 5 in appendix) to evaluate the performance of ZSL in two settings, i.e., conventional ZSL and GZSL. In conventional ZSL, test samples only consist of unseen classes, and we compute Top-1 accuracy for unseen classes (Acc) during inference. In GZSL, test samples consist of both seen and unseen classes, and we compute Top-1 accuracy for both seen (SA) and unseen (UA) classes. Further, we compute its corresponding harmonic mean (HM) using SA and UA, which is defined as HM. We also evaluate the model on the continual GZSL (CGZSL) setting.

	Methods	SUN				CUB				AWA1				AWA2				APY			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			APY	GZSL		
		Acc	UA	SA	HM	Acc	UA	SA	HM	Acc	UA	SA	HM	Acc	UA	SA	HM	Acc	UA	SA	HM
Generative Methods	SGAL (Yu and Lee, 2019)	-	42.9	31.2	36.1	-	47.1	44.7	45.9	-	52.7	75.7	62.2	-	55.1	81.2	65.6	-	-	-	-
	DASCN (Ni et al., 2019)	-	42.4	38.5	40.3	-	45.9	59.0	51.6	-	59.3	68.0	63.4	-	-	-	-	-	**39.7**	59.5	**47.6**
	CIZSL (Elhoseiny and Elfeki, 2019)	-	-	-	27.8	-	-	-	-	-	-	-	-	-	-	-	24.6	-	-	-	-
	TF-VAEGAN (Narayan et al., 2020)	-	45.6	40.7	43.0	-	52.8	64.7	58.1	-	-	-	-	-	59.8	75.1	66.6	-	-	-	-
	F-VAEGAN-D2 (Xian et al., 2019)	**64.7**	45.1	38.0	41.3	61.0	48.4	60.1	53.6	-	-	-	-	**71.1**	57.6	70.6	63.5	-	-	-	-
	CADA-VAE (Schonfeld et al., 2019)	-	47.2	35.7	40.6	-	51.6	53.5	52.4	-	57.3	72.8	64.1	-	55.8	75.0	63.9	-	-	-	-
	EPGN (Yu et al., 2020)	-	-	-	-	-	52.0	61.1	56.2	-	62.1	**83.4**	**71.2**	-	52.6	83.5	64.6	-	-	-	-
	LsrGAN (Vyas et al., 2020)	-	44.8	37.7	40.9	-	48.1	59.1	53.0	-	-	-	-	-	54.6	74.6	63.0	-	-	-	-
	ZSML (Verma et al., 2020)	-	21.7	**45.1**	29.3	-	**60**	52.1	55.7	-	57.4	71.1	63.5	-	58.9	74.6	65.8	-	-	-	-
	IZF (Shen et al., 2020)	**68.4**	**52.7**	**57**	**54.8**	**67.1**	52.7	68	**59.4**	**74.3**	61.3	80.5	**69.6**	**74.5**	**60.6**	77.5	**68.0**	**44.9**	**42.3**	60.5	**49.8**
	FREE (Chen et al., 2021b)	-	47.4	37.2	41.7	-	55.7	59.9	57.7	-	**62.9**	69.4	66.0	-	60.4	75.4	67.1	-	-	-	-
Non-generative Methods	DEVISE (Frome et al., 2013)	56.5	16.9	27.4	20.9	52.0	23.8	53.0	32.8	54.2	23.8	53.0	32.8	59.7	17.1	74.7	27.8	39.8	4.9	**76.9**	9.2
	ESZSL (Romera-Paredes and Torr, 2015)	54.5	11.0	27.9	15.8	53.9	12.6	63.8	21.0	58.2	6.6	75.6	12.1	58.6	5.9	77.8	11.0	38.3	2.4	70.1	4.6
	LATEM (Xian et al., 2016)	55.3	14.7	28.8	19.5	49.3	15.2	57.3	24.0	55.1	7.3	71.7	13.3	55.8	11.5	77.3	20.0	35.2	0.1	73.0	0.2
	SYNC (Changpinyo et al., 2016)	56.3	7.9	**43.3**	13.4	55.6	11.5	**70.9**	19.8	54.0	8.9	**87.3**	16.2	46.6	10.0	**90.5**	18.0	23.9	7.4	66.3	13.3
	SAE (Kodirov et al., 2017)	40.3	8.8	18.0	11.8	33.3	7.8	54.0	13.6	53.0	1.8	77.1	3.5	54.1	1.1	82.2	2.2	8.3	0.4	**80.9**	0.9
	DEM (Zhang et al., 2017)	40.3	20.5	34.3	25.6	51.7	19.6	57.9	29.2	68.4	32.8	**84.7**	47.3	67.2	30.5	**86.4**	45.1	35.0	11.1	75.1	19.4
	ZSKL (Zhang and Koniusz, 2018)	-	20.1	31.4	24.5	-	21.6	52.8	30.6	-	18.3	79.3	29.8	-	18.9	82.7	30.8	-	10.5	**76.2**	18.5
	DCN (Liu et al., 2018)	61.8	25.5	37.0	30.2	56.2	28.4	60.7	38.7	65.2	-	-	-	-	25.5	84.2	39.1	**43.6**	14.2	75.0	23.9
	SP-AEN (Chen et al., 2018)	59.2	24.9	38.6	30.3	55.4	34.7	**70.6**	46.6	-	-	-	-	58.5	23.3	**90.9**	37.1	24.1	13.7	63.4	22.6
	CDL (Jiang et al., 2018)	-	21.5	34.7	26.5	-	23.5	55.2	32.9	-	-	-	-	-	-	-	-	-	19.8	48.6	28.1
	PSR (Annadani and Biswas, 2018)	61.4	20.8	37.2	26.7	56.0	24.6	54.3	33.9	-	-	-	-	63.8	20.7	73.8	32.2	38.4	13.5	51.4	21.4
	RelNet (Sung et al., 2018)	-	-	-	-	55.6	38.1	61.4	47.0	68.2	31.4	91.3	46.7	64.2	30.9	93.4	45.3	-	-	-	-
	COSMO (Atzmon and Chechik, 2019)	-	44.9	37.7	41.0	-	44.4	57.8	50.2	-	-	-	-	-	-	-	-	-	-	-	-
	CRNet (Zhang and Shi, 2019)	-	34.1	36.5	35.3	-	45.5	56.8	50.5	-	52.6	78.8	63.1	-	58.1	74.7	65.4	-	32.4	68.4	44.0
	MLSE (Ding and Liu, 2019)	-	20.7	36.4	26.4	-	22.3	**71.6**	34.0	-	-	-	-	-	23.8	83.2	37.0	-	12.7	74.3	21.7
	DLFZRL (Tong et al., 2019)	-	-	-	24.6	-	-	-	37.1	-	-	-	-	-	-	-	45.1	-	-	-	31.0
	Triplet (Cacheux et al., 2019)	-	47.9	30.4	36.8	-	55.8	52.3	53.0	-	-	-	-	-	48.5	83.2	61.3	-	-	-	-
	CVC-ZSL (Li et al., 2019)	62.6	36.3	42.8	39.3	54.4	47.4	47.6	47.5	**70.9**	62.7	77.0	69.1	**71.1**	56.4	81.4	66.7	26.5	26.5	74.0	39.0
	APNet (Liu et al., 2020)	62.3	35.4	40.6	37.8	57.7	48.1	55.9	51.7	68.0	59.7	76.6	67.1	68.0	54.8	83.9	66.4	41.3	32.7	74.7	45.5
	DAZLE (Huynh and Elhamifar, 2020)	-	**52.3**	24.3	33.2	-	**56.7**	59.6	**58.1**	-	-	-	-	-	60.3	75.7	67.1	-	-	-	-
	DVBE (Min et al., 2020)	-	45.0	37.2	40.7	-	53.2	60.2	56.5	-	-	-	-	-	**63.6**	70.8	67.0	-	32.6	58.3	41.8
	CNZSL (Skorokhodov and Elhoseiny, 2021)	-	44.7	41.6	43.1	-	49.9	50.7	50.3	-	**63.1**	73.4	67.8	-	60.2	77.1	**67.6**	-	-	-	-
	HSVA (Chen et al., 2021c)	**63.8**	48.6	39.0	**43.3**	**62.8**	52.7	58.3	55.3	70.6	59.3	76.6	66.8	-	56.7	79.8	66.3	-	-	-	-
	CARNet (Ours)	63.1	**49.4**	40.5	**44.5**	**73.1**	**65.0**	59.6	**62.2**	**75.0**	**69.5**	74.7	**72.0**	**73.7**	**65.7**	79.7	**72.0**	**45.3**	**39.9**	65.9	**49.7**

Table 1: ZSL and GZSL results (%) on the ZSL benchmark datasets. The best, the second best, and the third best results are made as bold. The best results are underlined. '-' denotes that results are not available in the paper.

5.1 Comparison with Baseline Methods

In this section, the performance of CARNet is evaluated against strong baseline models for three ZSL settings. Results for both the ZSL settings are provided in Table 1.

Conventional ZSL: From Table 1, it is observed that the proposed CARNet outperforms all non-generative ZSL methods in conventional ZSL setting, by 10.3%, 4.1%, 2.6%, 1.7% and 1.3% absolute gain for CUB (Welinder et al., 2010), AWA1 (Lampert et al., 2009), AWA2 (Lampert et al., 2013), and aPY (Farhadi et al., 2009) and SUN (Patterson and Hays, 2012) datasets, respectively. Also, in comparison to generative methods of ZSL, the proposed approach shows 6% and 0.4% absolute gain over CUB and AWA1 datasets, respectively. On the remaining datasets, the model outperforms all the generative models but shows competitive performance to IZF IZF (Shen et al., 2020).

Generalized zero-shot learning: In this setting, the proposed non-generative CARNet model outperforms all non-generative ZSL methods with the absolute gain of 1.2%, 4.1%, 2.9%, 4.4%, and 4.2% for SUN, CUB, AWA1, AWA2, and aPY datasets, respectively. Moreover, CARNet yields the best HM compared to all generative/non-generative methods for CUB, AWA1, and AWA2 datasets and better/similar HM on aPY dataset. The performance of CARNet is outperformed by IZF (Shen et al., 2020) on the SUN dataset. It should be noted that IZF is an invertible flow-based generative model which learns from the bi-directional mapping between the visual and the

Methods	CUB			SUN		
	mUA	mSA	mHM	mUA	mSA	mHM
Seq-CARNet	4.5	11.3	5.9	4.3	11.8	6.1
Seq-CNZSL (Skorokhodov and Elhoseiny, 2021)	-	-	23	-	-	14
Seq-CVAE (Mishra et al., 2018)	8.6	24.7	12.2	11.4	16.9	13.4
Seq-CADA-VAE (Schonfeld et al., 2019)	14.4	40.8	21.1	16.2	25.9	20.1
CNZSL-AGEM (Skorokhodov and Elhoseiny, 2021)	-	-	23.8	-	-	14.2
CNZSL-EWC-online (Skorokhodov and Elhoseiny, 2021)	-	-	23.3	-	-	14.3
CNZSL-MAS-online (Skorokhodov and Elhoseiny, 2021)	-	-	23.8	-	-	14.2
GRCZSL (Gautam et al., 2021a)	14.1	41.9	20.5	11.5	17.7	13.7
CZSL-CV+res (Gautam et al., 2020)	13.5	44.9	20.2	14.1	24.0	17.6
CZSL-CA+res (Gautam et al., 2020)	32.8	44.0	36.1	21.7	27.1	22.9
Tf-GCZSL (Gautam et al., 2021b)	32.4	46.6	36.3	**24.7**	28.1	24.8
CARNet-ER (Ours)	43.0	45.8	43.4	23.3	30.3	25.6
CARNet-ER+CBR (Ours)	**43.4**	**47.4**	**44.2**	23.6	**30.9**	**26.0**

Table 2: Continual Generalized Zero-shot Learning Results

attribute space, enabling it to have better performances than other generative ZSL methods. However, CARNet outperforms IZF for CUB, AWA1, and AWA2 by a significant margin of 2.8%, 2.4%, 3.7%, respectively, and yields similar result for aPY dataset (only a difference of 0.1%). In addition to performance gains, the proposed approached CARNet is characteristically advantageous over generative ZSL approaches in that, while the CARNet uses only the attribute vectors of seen classes during training, the generative ZSL methods use the attribute vectors of both seen and unseen classes during training which is not a realistic scenario in a dynamic environment.

Table 3 presents the computational time required to train the various ZSL methods. It can be observed from this table that the CARNet is at least $68\times$, $68\times$, $21\times$, and $31\times$ times faster than generative methods for SUN, CUB, AWA1, and AWA2, as observed in Table 3. This can be attributed to the fact that CARNet only needs to process class attribute vectors through ARN and AE.

Thus, the proposed CARNet is a desirable candidate for conventional and generalized ZSL, owing to its performance, data requirements and computational speed.

Continual generalized zero-shot learning (CGZSL): While ZSL assumes data for all tasks to be available apriori, data may arrive in a sequential manner in real-world, and collecting all the data in memory is cumbersome. Hence, we further evaluate the performance of CARNet for the highly challenging CGZSL setting proposed in Skorokhodov and Elhoseiny (2021). This setting assumes that the data arrives in a sequence of tasks and only the current task data is available for training. Thus, after training for a sequence of $[1, \ldots, t]$ tasks, all classes in the $[1, \ldots, t]$ tasks are considered as seen classes and classes from $(t+1)$ onward are considered as unseen classes. As experience replay-based methods generally outperform regularization-based methods in the literature (Delange et al., 2021), CARNet is equipped for CGZSL using experience replay (ER) (Chaudhry et al., 2019b) strategy with class-balanced reservoir (CBR) sampling (Chrysakis and Moens, 2020). We measure the performance of CGZSL method using SA, UA, and HM at each task. Further, we compute the mean of SA, UA, and HM of overall tasks and denote it as mSA, mUA, and mHM (Skorokhodov and Elhoseiny, 2021). We present the CGZSL results in Table 2, along with the state-of-the-art CGZSL methods. Our method outperforms all existing methods by an absolute gain of 7.1% and 0.4% for CUB and SUN datasets, respectively. We also provide the performance of CARNet with CBR sampling (CARNet-ER+CBR) and without CBR sampling (CARNet-ER) in Table 2.

5.2 Ablation Study: Significance of Individual Components in CARNet

In this section, to emphasize the significance of individual components of CARNet, we perform an extensive ablation study over all the components and hyperparameters.

We study the effect of individual components of the CARNet, namely, (i) ARN (ii) AE (iii) circle loss (iv) softmax cross-entropy loss (v) scaled cosine similarity or Dot Product. We present the

Methods	SUN	CUB	AWA1	AWA2
RelNet (Sung et al., 2018)	-	25 min	40 min	40 min
DCN (Liu et al., 2018)	40 min	50 min	-	55 min
CIZSL (Elhoseiny and Elfeki, 2019)	3 Hr	2 Hr	3 Hr	3 Hr
CVC-ZSL (Li et al., 2019)	3 Hr	3 Hr	1.5 Hr	1.5 Hr
LsrGAN (Vyas et al., 2020)	1.1 Hr	1.25 Hr	-	1.5 Hr
TF-VAEGAN (Narayan et al., 2020)	1.5 Hr	1.75 Hr	-	2 Hr
CNZSL (Skorokhodov and Elhoseiny, 2021)	20 sec	20 sec	30 sec	30 sec
CARNet (Ours)	35 sec	22 sec	110 sec	77 sec

Table 3: Training time comparison. CNZSL (Skorokhodov and Elhoseiny, 2021) is a non-generative model and remaining are generative models.

results of this study in Table 4. The softmax cross-entropy loss is an imperative loss for the model, as the proposed CARNet has to perform classification. Therefore, we kept it in all cases in the component analysis of Table 4. It is very evident from the results that attribute refinement significantly boosts the performance of CARNet. Moreover, the scaled-cosine similarity is another important component, and helps to outperform the model with another potential candidate, namely, dot product by a large margin.

$\mathcal{L}_{Soft-ce}$	✔	✔	✔	✔	✔
ARN		✔		✔	✔
$\mathcal{L}_{Circle}$			✔	✔	✔
Scaled-Cosine Similarity	✔	✔	✔		✔
Dot-product				✔	
SUN	43.1	43.9	43.4	37.8	44.5
CUB	58.1	58.8	59.4	42.7	62.2
AWA1	67.4	71.2	70.1	27.4	72.0
AWA2	69.6	70.5	70.0	17.3	72.0
APY	43.9	45.2	47.3	17.3	49.7

Table 4: Component Analysis

6 Conclusion

In this work, we developed the circle loss guided gating-based attribute-refinement network for handling ZSL, GZSL, and continual-GZSL tasks. CARNet refines the attribute through a gating unit where it improves the attribute representation by learning a self-weight on each attribute dimension in a projected space. These refined attributes improve the embedding, which helps to overcome the model bias towards the seen classes. The whole model is guided by the circle loss along with the standard softmax cross-entropy loss, which maximizes the inter-class separability and intra-class similarity. Also, unlike the generative method, CARNet does not require the attribute vector of the unseen classes during training. The proposed method is quite fast, as the attribute refinement network and the attribute embedder need to process only the class attribute vectors during training. This work shows that a simple MLP-based architecture can outperform various highly computationally expensive ZSL methods. This approach needs to be explored with generative methods and other applications of ZSL, like zero-shot for sketch-based image retrieval, action recognition, and natural language processing.

7 Limitations

One major limitation is that the inference data must be from the same domain, as the proposed model cannot handle data from the other domains on which the model is not trained. Another limitation of the proposed method is that it requires task id during training in the CGZSL setting, without which CARNet cannot optimize the proposed model properly. However, in realistic scenarios, it is not necessary for the data to arrive with well-defined task-boundaries. Hence, the requirement of task id during training is a drawback of our proposed model.

Acknowledgements

We would like to thank the Wipro IISc Research and Innovation Network (WIRIN, Grant No-99325T) and National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-027) for funding this research.

References

Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. 2016. Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 59–68.

Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. 2015. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2927–2936.

Yashas Annadani and Soma Biswas. 2018. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612.

Yuval Atzmon and Gal Chechik. 2019. Adaptive confidence smoothing for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11671–11680.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Yannick Le Cacheux, Herve Le Borgne, and Michel Crucianu. 2019. Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10333–10342.

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019b. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Ian Horrocks, Jeff Z. Pan, and Huajun Chen. 2021a. Knowledge-aware zero-shot learning: Survey and perspective. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4366–4373.

Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. 2018. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, pages 1043–1052.

Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. 2021b. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 122–131.

Shiming Chen, Guosen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. 2021c. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. *Advances in Neural Information Processing Systems*, 34.

Aristotelis Chrysakis and Marie-Francine Moens. 2020. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhengming Ding and Hongfu Liu. 2019. Marginalized latent semantic encoder for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6191–6199.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

Mohamed Elhoseiny and Mohamed Elfeki. 2019. Creativity inspired zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5784–5793.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE.

Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. 2018. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129.

Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. 2014. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, pages 584–599. Springer.

Yanwei Fu and Leonid Sigal. 2016. Semi-supervised vocabulary-informed learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5337–5346.

Chandan Gautam, Sethupathy Parameswaran, Ashish Mishra, and Suresh Sundaram. 2020. Generalized continual zero-shot learning. *arXiv preprint arXiv:2011.08508*.

Chandan Gautam, Sethupathy Parameswaran, Ashish Mishra, and Suresh Sundaram. 2021a. Generative replay-based continual zero-shot learning. *arXiv preprint arXiv:2101.08894*.

Chandan Gautam, Sethupathy Parameswaran, Ashish Mishra, and Suresh Sundaram. 2021b. Online lifelong generalized zero-shot learning. *arXiv preprint arXiv:2103.10741*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Dat Huynh and Ehsan Elhamifar. 2020. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4483–4493.

Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2018. Learning class prototypes via structure alignment for zero-shot recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 118–134.

Rohit Keshari, Richa Singh, and Mayank Vatsa. 2020. Generalized zero-shot learning via over-complete distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13300–13308.

Elyor Kodirov, Tao Xiang, and Shaogang Gong. 2017. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3174–3183.

Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3.

Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE international conference on computer vision*, pages 4247–4255.

Kai Li, Martin Renqiang Min, and Yun Fu. 2019. Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3583–3592.

Yan Li, Junge Zhang, Jianguo Zhang, and Kaiqi Huang. 2018. Discriminative learning of latent features for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7463–7471.

Hanxiao Liu, Zihang Dai, David So, and Quoc Le. 2021. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34.

Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2020. Attribute propagation network for graph zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4868–4875.

Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018. Generalized zero-shot learning with deep calibration network. In *Advances in neural information processing systems*, pages 2009–2019.

Shaobo Min, Hantao Yao, Hongtao Xie, Chaoqun Wang, Zheng-Jun Zha, and Yongdong Zhang. 2020. Domain-aware visual bias eliminating for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12664–12673.

Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. 2018. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2188–2196.

Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. 2020. Latent embedding feedback and discriminative features for zero-shot classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 479–495. Springer.

Jian Ni, Shanghang Zhang, and Haiyong Xie. 2019. Dual adversarial semantics-consistent network for generalized zero-shot learning. *Advances in Neural Information Processing Systems*, 32:6146–6157.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. *Advances in neural information processing systems*, 22.

Genevieve Patterson and James Hays. 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758. IEEE.

Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58.

Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

Yuming Shen, Jie Qin, Lei Huang, Li Liu, Fan Zhu, and Ling Shao. 2020. Invertible zero-shot recognition flows. In *European Conference on Computer Vision*, pages 614–631. Springer.

Ivan Skorokhodov and Mohamed Elhoseiny. 2021. Class normalization for (continual)? generalized zero-shot learning. In *International Conference on Learning Representations*.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. 2018. Transductive unbiased embedding for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1024–1033.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Bin Tong, Chao Wang, Martin Klinkigt, Yoshiyuki Kobayashi, and Yuuichi Nonaka. 2019. Hierarchical disentanglement of discriminative latent features for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11467–11476.

Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. 2020. Meta-learning for generalized zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6062–6069.

Maunil R Vyas, Hemanth Venkateswara, and Sethuraman Panchanathan. 2020. Leveraging seen and unseen semantic relationships for generative zero-shot learning. In *European Conference on Computer Vision*, pages 70–86. Springer.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37.

Kun Wei, Cheng Deng, and Xu Yang. 2020. Lifelong zero-shot learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 551–557.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-ucsd birds 200.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2018. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. 2016. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018a. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018b. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591.

Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10275–10284.

Hyeonwoo Yu and Beomhee Lee. 2019. Zero-shot learning via simultaneous generating and learning. *Advances in Neural Information Processing Systems*, 32:46–56.

Yunlong Yu, Zhong Ji, Jungong Han, and Zhongfei Zhang. 2020. Episode-based prototype generating network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14035–14044.

Fei Zhang and Guangming Shi. 2019. Co-representation network for generalized zero-shot learning. In *International Conference on Machine Learning*, pages 7434–7443. PMLR.

Hongguang Zhang and Piotr Koniusz. 2018. Zero-shot kernel learning. In *CVPR*, pages 7670–7679.

Li Zhang, Tao Xiang, and Shaogang Gong. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2030.

Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174.

A Appendix

A.1 ZSL Datasets

Five common benchmark datasets used in ZSL are Scene UNderstanding (SUN) dataset, Caltech-UCSD Birds (CUB) dataset, Animals With Attributes (AWA1, AWA2) dataset and Attribute Pascal and Yahoo (aPY) dataset. A brief description of these datasets is provided in Table 5.

A.2 Continual Learning Setup

In a continual learning setup, all the classes (both seen and unseen classes) of a given dataset are grouped together and split into T tasks. Further, all classes up to the current task t are taken as seen classes. All classes of the remaining tasks, namely t+1 to T, are taken as unseen classes. At task t, the input consists of training images for all classes in task t. During testing, the test data consists of images belonging to all classes of tasks 1 to T. The Task split for the CUB and SUN dataset are as follows:

1. The CUB dataset consists of 200 classes. The dataset is split into 20 tasks with 10 classes in each task.

2. The SUN dataset consists of 717 classes. The dataset is split into 15 tasks, with 47 classes in the first 3 tasks and 48 classes in the remaining 12 tasks.

A.3 Implementation Details

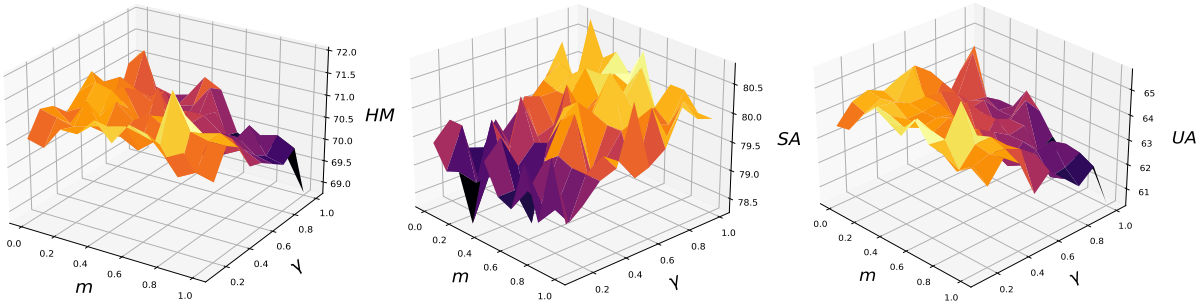
We use ResNet-101 as a pretrained model, which is pretrained on ImageNet as the backbone for visual feature extraction. CARNet is trained using the Adam optimizer with a learning rate of 0.001 for APY and 0.0005 for all remaining datasets. Further, weight decay of 0.001 for APY and SUN, and 0.0001 for other datasets are used. We choose $\beta = 5$ across all datasets for computing scaled cosine similarity and m, γ, λ are taken as shown in Table 6. We performed all our experiments on RTX 2080 GPU with i7 processor and 32 GB RAM.

A.4 Impact of Hyper-parameters (γ and m)

In Fig. 3, we provide three 3-D plots for HM, SA, and UA on AWA2 dataset to study the effects of m and γ on CARNet. From the figure, it is seen that HM, SA, and UA vary only 3%, 2%, and 4% with changes in m and γ, and still outperforms most generative and all non-generative ZSL methods.

Dataset	Attribute Dimension	#Seen Classes	#Unseen Classes	Total Classes	Description
SUN	102	645	72	717	Fine-grained
CUB	1024	150	50	200	Fine-grained
AWA1	85	40	10	50	Coarse-grained
AWA2	85	40	10	50	Coarse-grained
aPY	64	20	12	32	Coarse-grained

Table 5: Zero-shot learning benchmark datasets



(a) HM (b) SA (c) UA

Figure 3: Impact of hyperparameters on proposed CARNet model for AWA2 dataset on GZSL setting with $\lambda = 0.8$

Thus, it can be observed that the CARNet is robust to large changes in the γ and m.

Hyperparameter	SUN	CUB	AWA1	AWA2	APY
m	0.4	0.1	0.3	0.4	0.2
γ	0.5	0.9	1.0	0.5	1.0
λ	1.2	0.7	0.2	0.5	1.0

Table 6: Values taken by the Hyperparameters for different datasets