World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:17, No:10, 2023

Shifted Window Based Self-Attention via Swin Transformer for Zero-Shot Learning

Yasaswi Palagummi, Sareh Rowlands

***Abstract*—**Generalised Zero-Shot Learning, often known as GZSL, is an advanced variant of zero-shot learning in which the samples in the unseen category may be either seen or unseen. GZSL methods typically have a bias towards the seen classes because they learn a model to perform recognition for both the seen and unseen classes using data samples from the seen classes. This frequently leads to the misclassification of data from the unseen classes into the seen classes, making the task of GZSL more challenging. In this work, we propose an approach leveraging the Shifted Window based Self-Attention in the Swin Transformer (Swin-GZSL) to work in the inductive GZSL problem setting. We run experiments on three popular benchmark datasets: CUB, SUN, and AWA2, which are specifically used for ZSL and its other variants. The results show that our model based on Swin Transformer has achieved state-of-the-art harmonic mean for two datasets - AWA2 and SUN and near-state-of-the-art for the other dataset - CUB. More importantly, this technique has a linear computational complexity, which reduces training time significantly. We have also observed less bias than most of the existing GZSL models.

***Keywords*—**Generalised Zero-shot Learning, Inductive Learning, Shifted-Window Attention, Swin Transformer, Vision Transformer.

I. Introduction

WITH the help of Deep Learning, humanity has made significant progress in solving many problems related to AI. Significant advances in many tasks, especially visual recognition tasks, are not just due to the use of advanced deep learning architectures, but also due to the use of massive amounts of labelled datasets. While the advancements in deep learning architectures are admirable, dependence on huge volumes of data is sometimes problematic. On one hand, expecting to obtain readily available labelled datasets for each and every problem we attempt to solve is irrational and on the other hand, manually creating annotated datasets is also not practical because it not only requires human labour and domain expertise but also leads towards unrealistically attempting to annotate every kind of image that exists on the planet. If we consider a child who can learn to be able to distinguish between a cat and a dog using a single image of them, an AI system needs many images of those animals to be able to perform the same task with good accuracy. In response to external cues, humans are quick to grasp novel concepts and much quicker to identify variations of those concepts and recall them in different contexts.

General Intelligence is defined as the capability to execute a range of objectives and complete a variety of activities in a variety of scenarios and contexts [1]. Zero-Shot Learning (ZSL) is an attempt to achieve Artificial General Intelligence in AI systems to mimic human general intelligence so that these systems can be deployed in contextually different environments. ZSL is the process of leveraging semantic information (e.g., characteristics) to distinguish between seen and unseen samples when unseen classes are not observed during training. This is achieved by transferring semantic knowledge from seen classes to unseen classes. ZSL attempts to solve tasks where limited training data is available by learning novel classes using the semantic information available. With these methods, a model only needs to be trained once to acquire the capacity to generalize to new tasks with classes that are underrepresented in the training data. In ZSL, the semantic information of both seen and unseen classes are available to us but the images of seen classes are available for training. Using ZSL, we can only classify images from unseen classes or novel classes by leveraging prior knowledge and the semantic information present in the form of dimensional vectors. As mentioned earlier, the focus of this work is centred on Fine-Grained Image classification using GZSL, in which test data can be from both seen and unseen classes, which is closer to real-time practical applications. Fine-grained classification can be defined as recognising classes that are visually very similar. This is an important yet challenging task with a wide range of applications from the fashion industry, e.g., recognition of different types of shoes or cloth, to face recognition and environmental conservation, e.g., recognizing endangered species of birds or plants [2].

In this work, we propose the use of Swin (shifted window) Transformer towards solving GZSL as an approach that achieves state-of-the-art results in a more efficient way. We compare our approach (Swin-GZSL) against the Vision Transformer-based approach [4], which is the current state-of-the-art.

II. Background

GZSL mainly suffers from two problems - Bias and Hubness. As explained above, GZSL methods are generally biased towards seen classes i.e., their performance on seen classes tends to be a lot better than that of unseen classes. In addition to this, GZSL methods tend to push the correct labels down their neighbour list, to the neighbour class with a high number of items. These feature vectors that have a high number of items mapped to them are called hubs and this is called the Hubness

Yasaswi Palagummi is with University of Exeter, UK (e-mail: yasaswi.palagummi@gmail.com).

problem [5] where instead of the class label with maximum similarity, its closest hub is returned as the predicted label. Various techniques based on attention, advanced post-processing, feature de-correlation and hubness-repellent mapping can be used to solve these problems and achieve better-performing GZSL systems [6]-[8].

When deep learning architectures were not as prevalent, GZSL methods [9]-[11] were developed to learn a mapping between the semantic and visual space that enables the transfer of knowledge of semantic dimensional vectors from seen classes to unseen classes. The performance of these models was poor due to the fact that not only were the models simplistic, but they also relied on global characteristics. The performance of GZSL models, however, has been vastly enhanced by recent developments in the field of deep learning. Socher et al. [12] was the first to introduce the concept of GZSL in 2013, as discussed in [13]. Although many attempted to solve GZSL, it did not gain traction until 2016, when Chao et al. empirically showed that techniques under ZSL setting cannot perform well under the GZSL setting [50]. Most attempts to solve the GZSL task can be broadly categorised into two classes - Embedding based and Generative based methods as shown in Fig. 1 [13]. In order to link the low-level visual characteristics of observed classes with the appropriate semantic vectors, embedding-based approaches learn an embedding space. Generative-based methods learn a method for generating images for unseen classes from the semantic representations of both classes and visual features of seen classes thereby converting the GZSL problem into a conventional supervised learning problem [13]. As our proposed methodologies are Embedding-based, we perform research on a range of Embedding-based techniques and present them below.

A. Embedding Based Methods

GZSL models are trained using both visual features - containing only seen samples, and semantic features - containing both seen and unseen samples. In order to correlate the visual features of the viewed class to their corresponding semantic feature vectors, most ZSL methods use an embedding function. Upon optimisation, this function gains the ability to recognise new classes by comparing the embedding space representations of the prototype and the predicted. The embedding space is divided into three categories: semantic embedding, visual embedding, and latent space embedding. As opposed to semantic embedding, which performs classification in the semantic space by mapping attributes from the visual space to the semantic space (forward-projection), visual embedding performs classification in the visual space by mapping the semantic representations to the visual space (backward-projection). In order to discover cross-modal shared semantic characteristics, some works propose learning an intermediate space shared by the visual features and semantic features [11]-[13]. This is accomplished by projecting the visual and semantic features into a common Latent space L (intermediate projection).

B. Early Approaches

Deep Visual-Semantic Embedding, or DeViSe, is a model that was first developed in 2013 that is capable of recognising visual objects by making use of both labelled and semantic information [14]. This was one of the first methods that demonstrated that semantic information could be used to predict novel classes. Later, Convex combination of Semantic Embeddings (ConSe) was introduced in 2014, with the aim of building an embedding model from an already-existing image classifier and a semantic word embedding model that includes the n-class labels in its vocabulary [15]. This method does not require any extra training as it maps the images into the semantic embedding space using a complicated combination of class label embedding vectors [15]. Another important paper from 2015 presented methods to enhance ZSL by reducing the hubness problem through the use of the proximity distribution of possible neighbours across several mapped vectors and the substitution of globally adjusted closest neighbour queries [16].

C. Auto Encoder Based Approaches

In 2018, an approach to preserving the semantic relations in the embedding space with the help of objective functions that induce semanticity to the embedding space was proposed [28]. This approach focuses on effectively utilising the semantic space by introducing relations between classes based on the similarity of their semantic content. Another novel approach - Low-rank Embedded Semantic AutoEncoder - that leverages the low intrinsic dimensionality of data has been proposed which links visual features with their semantic representations using low-rank mappings [29]. While the encoder aims to learn the low rank mapping that links visual features to semantic space, the decoder's task is to reconstruct the original data using the mapping that the encoder learns [29]. Furthermore, techniques like *Meta Learning* [22]-[24], *Knowledge Graphs* [25]-[27], and *Bi-directional Learning* [30], [31] that utilise cutting-edge techniques in diverse contexts have been proposed to enhance GZSL task.

D. Recent Approaches

To produce semantically relevant representations for images, in 2018, a Stacked Semantic Directed Attention Model that progressively assigns weights for distinct regional features based on semantic descriptions has been suggested [17]. A framework using Latent Feature Guided Attribute Attention has also been proposed which uses both global class-level features and low-level visual information to perform object-based attribute attention for semantic disambiguation, where attention is used to integrate both low-level and global features from semantic space [18]. Goal-Oriented Gaze Estimation is another unique approach that uses visual attention by predicting the human gaze location using semantic query-guided attention to identify unseen class objects [19]. Leveraging both attention and transformers, approaches like [4], [20] and [21] have been proposed very recently to solve the GZSL task.

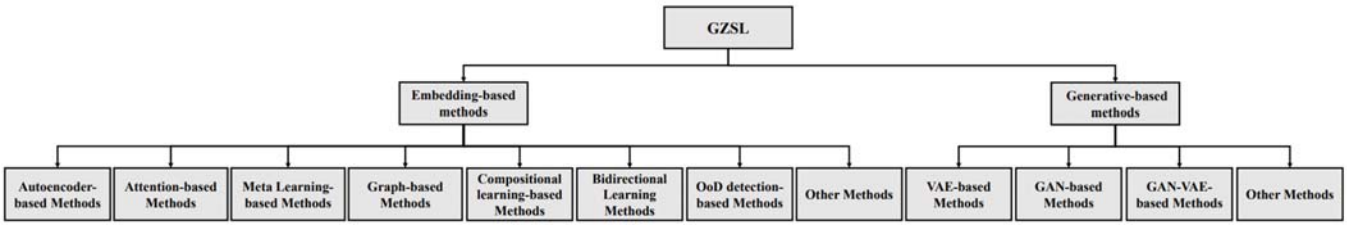

Fig. 1 The taxonomy of GZSL models [13]

III. Objectives

Even the state-of-the-art methods used to solve the problem of GZSL are seen to exhibit high amounts of bias in their results. Models tend to recognise images from seen classes with far better accuracy than images from unseen classes. Our primary objective is to build a GZSL model to bridge the gap in performance across seen and unseen classes, that is able to generalise well on unseen data i.e., show less bias in its performance at the same time performing similar to or better than the existing state-of-the-art models. Moreover, the current state-of-the-art technique has quadratic computational complexity which makes it quite time-consuming during the training process. Our secondary objective is to reduce this computational complexity as much as possible. In order to achieve these objectives, we use the Swin Transformers with Shifted window based Self Attention. As Swin-Transformer based approaches have shown to outperform the Vision Transformer (ViT) model in terms of semantic segmentation and COCO object recognition [32], we wanted to implement it in the ZSL setting and measure its performance. We build a deep learning classifier implementing the Swin architecture in order to see the effectiveness of them in solving the GSZL. We perform experiments on the three benchmark datasets: Animals with Attrubites2 (AWA2), Caltech-UCSD-Birds (CUB) and SUN.

IV. Proposed Approach

A. Architecture Overview

Swin Transformer is a novel transformer design developed in 2021 by researchers at Microsoft and is published by Liu et al. in the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" that can effectively serve as a general-purpose backbone for computer vision applications [32]. "Swin" basically stands for Shifted Windows, implying that Swin Transformer essentially is a hierarchical Transformer that uses Shifted Windows to compute image feature representations by combining image patches in deeper layers. The shifted windowing technique improves performance by enabling cross-window connectivity while restricting self-attention computation to non-overlapping local windows. This also means that the computational complexity of the Swin transformer is linear to the input image size while the other vision transformers have a quadratic computational complexity to input size [33], as the attention module is only computed within local windows, unlike other vision transformers where it is computed globally.

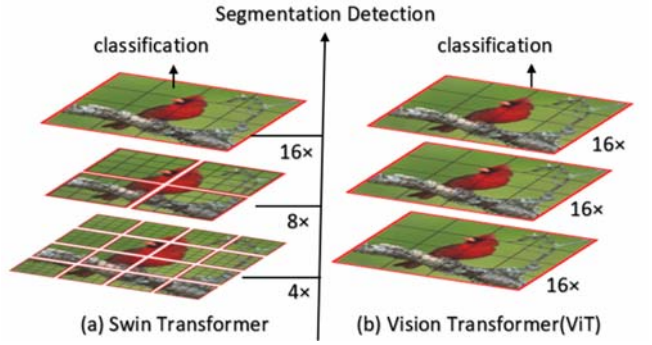

Fig. 2 Hierarchical feature mapping of Swin and ViT architectures [32]

As seen in Fig. 2, the swin transformer generates a hierarchical representation of the image by beginning with relatively tiny portions/patches of the image, then gradually merging adjacent patches to generate larger patches as the transformer layers become deeper. Because the number of patches inside each window is fixed throughout the architecture regardless of the depth of layers, the complexity of this approach will always be proportional to the image size.

B. Shifted Window Based Attention Module

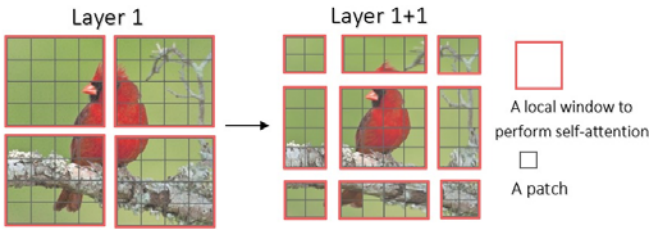

Fig. 3 Shifted-Window method to compute self-attention [32]

One of the important design elements of the Swin Transformer is the scheme of shifting the window partition between consecutive self-attention layers (Fig. 3). Shifted windows is not entirely a novel idea - it is something that is an essential part of Convolutional Neural Networks (CNN) which has made CNN so powerful and accurate. But, as far as transformers are concerned, Swin is the first architecture to implement the this. The shifting of the window partition that occurs between consecutive self-attention layers is one of the essential components of the design of the Swin Transformer. Shifting windows allows us to partition the images into distinct patches in which we can bridge the windows of the previous layer, enabling connections between windows and increasing modelling capabilities. Unlike sliding-window-based self-

attention methods, which are troubled by latency, shifted-window methods have significantly less latency while retaining high modelling capacity.

A window-based self-attention module's modelling capacity is limited because there is no information transfer between windows. To address this, the Swin architecture proposes shifted-windows based self-attention, which enables cross-window communication. Unlike a normal Vision Transformer where attention is fixed on certain regions, in Swin, the attention window keeps shifting with respect to the previous layer, just like in a strided convolution. Patches that were bounded by different windows in a layer cannot communicate in that layer, although they are adjacent. But Swin facilitates communication in the next layer or deeper layers. Each transformer, as depicted Fig. 3, contains two modules. In the first module, partitioning begins at the top-left pixel and it evenly divides the 8×8 feature map into four windows of 2×2 size each. In the next module, the window partitioning is displaced by $(\lfloor\frac{M}{2}\rfloor, \lfloor\frac{M}{2}\rfloor)$ pixels, relative to the previous module. Consecutive Swin Transformer blocks in this shifted window approach are calculated as follows [32]:

$$\hat{z}^l = W - MSA\left(LN(\hat{z}^{l-1})\right) + \hat{z}^{l-1}$$

$$\hat{z}^l = MLP\left(LN(\hat{z}^l)\right) + \hat{z}^l$$

$$\hat{z}^{l+1} = SW - MSA\left(LN(\hat{z}^l)\right) + \hat{z}^l$$

$$\hat{z}^{l+1} = MPL\left(LN(\hat{z}^{l+1})\right) + \hat{z}^{l+1} \quad (1)$$

where z^l is the MLP module for block l; $\hat{z}^l$ is the output features generated by the shifted window module (SW-MSA); W-MSA stands for self-attention calculated using regular window partitioning; SM-MSA stands for self-attention calculated using shifted window. partitioning. A general Vision Transformer [33] has the standard multi-head self-attention (MSA) whereas the Swin transformer block (Fig. 4) has the shifted window-based attention module (SW-MSA) replacing MSA. All the other layers between these two architectures remain the same. The MSA and SW-MSA modules are connected to a 2-layer MLP with GELU non-linearity function in between them, before finally connecting to a dropout layer in the MLP [32]. Layer Normalization is applied after each stage i.e., before MSA and MLP modules, in each of the individual units in block.

Within MSA and SW-MSA, self-attention is calculated for all the patches in patch embeddings separately. Query(Q), Key(K) and Value(V) are created by multiplying the patch embeddings by three weight matrices trained during the training process to calculate self-attention [4]. The dot product of Q and K is calculated which results in the attention intensity matrix which describes the amount of attention a patch embedding has to dedicate to other embeddings. Finally, a softmax function is applied to the score matrix and then multiplied by values, as shown in [34]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where Q, K, and V are Query, Key and Value; d_k is the number of dimensions of the Key Vector; V is the value vectors [32].

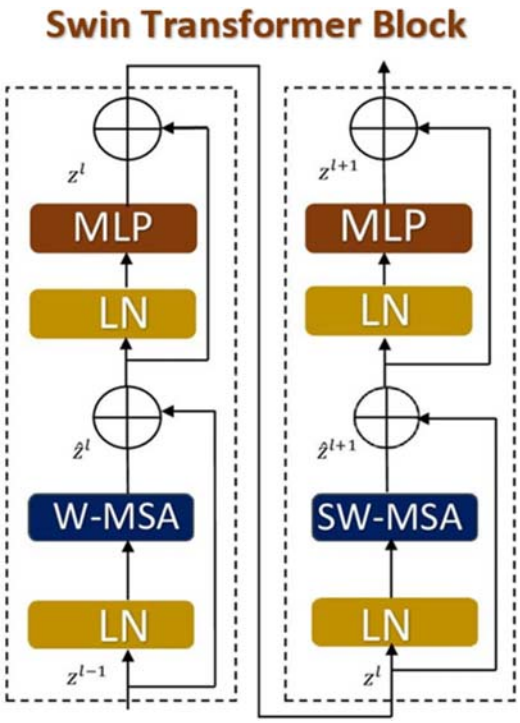

Fig. 4 Two Successive Swin Transformer Blocks with Multihead attention and Shifted Window Attention [32]

Many embedding based approaches learn an embedding space between the global visual features and semantic representations. But the main focus of Attention based approaches is to learn the most important regions of an image that provide maximum discriminative features. So, the task of visual attention is to divide an image I into R regions denoted by $\{I^r\}_{r\epsilon R}$, which can be either equal-sized or arbitrary grids and generate features from the most relevant regions of the image [3]. Given region features $\{f^r\}_{r\epsilon R}$, the goal of the attention module g(.) is to find the most relevant regions for the specific task at hand [3]. This is done by finding an attention feature, f, which is defined by [3]:

$$f = g(f^1, \ldots, f^R) = \sum_{r=1}^{R} \alpha_r (f^r) f^r \quad (3)$$

V. Experiment Design

A. Implementation Details

Swin-L, a large variant of Swin, has been implemented using PyTorch. Image size of 224 * 224 is used, and no data augmentations have been performed on the dataset. The input patch size is 4 * 4 with 192 connections in each of the fully connected layers. This architecture contains 564 attention heads across all four stages of the Transformer Block - 12, 24, 432, 96 heads respectively in each stage. The window size of all

attention modules is considered to be 7 * 7. On the whole, this architecture comprises of 197M parameters, and we used Adam optimizer to fine-tune the model with a weight decay of 0.05, learning rate of 0.0001 and batch size of 64. The architecture has been implemented in PyTorch using the cloud platform - Gradient Paperspace, using A-6000 with 48GB GPU and 45GB RAM.

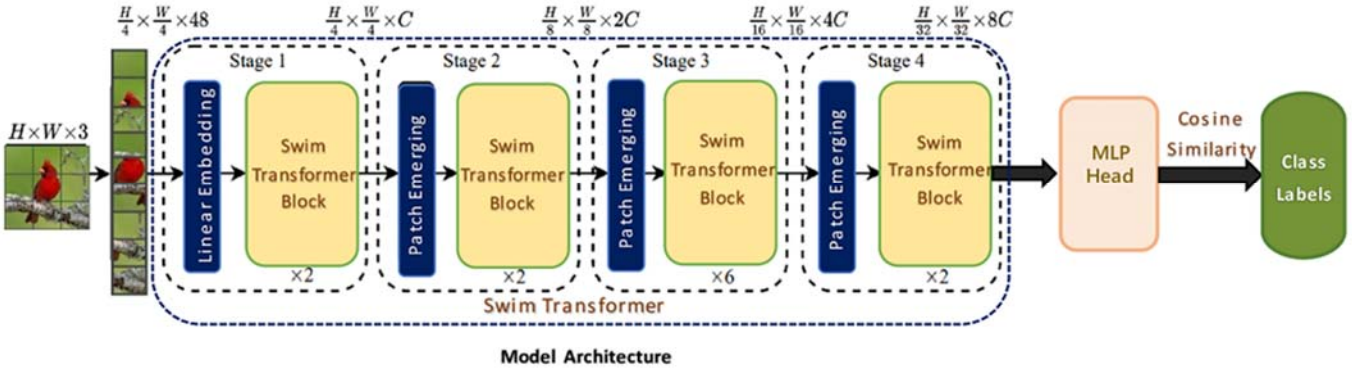

Fig. 5 Swin Transformer Architecture for GZSL (proposed approach)

TABLE I
SUMMARY OF DATASETS

Attribute	AwA2	CUB	SUN
#Total	37,322	11,788	14,340
#Classes	50	200	717
#Attribute Length	85	312	102
#Seen Classes	40	50	645
#Unseen Classes	10	50	72
#Train Images	23,527	7,057	10,320
#Validation Images	5882	1,764	2,580
#Test Images	7,913	2,967	1,440

B. Datasets

We have used the datasets mentioned below for the purposes of training, validating, and testing our model. These datasets are freely available public benchmarks that have been utilized to evaluate ZSL and GZSL for many years. Therefore, data privacy is not an issue. To evaluate our model, we conduct experiments on the following datasets.

1. CUB: Caltech-UCSD-Birds [48] is a fine-grained image dataset that contains images from 150 seen classes and 50 unseen classes. This dataset contains images of 200 birds belonging to 200 classes with a total of 11,788 images. Each class is represented using a continuous semantic vector of 312 dimensions. This dataset includes attribute location annotations, which will make it easier to locate objects by utilizing small discriminative regions. Our technique is weakly supervised method and does not require location annotations, hence we do not use these labels.
2. SUN: SUN [47] is a dataset that contains 14,340 images with a total of 645 seen classes and 72 unseen classes. It contains the most classes among all the 3 datasets with each class containing a 102D continuous semantic vector.
3. AWA2: Animals with Attributes [41] is a dataset of animal images with 37,322 images. It has 40 seen and 10 unseen classes in total and an 85D semantic vector associated with each class.

In [48], Xian et al. propose a new way to split all the above three datasets ensuring that none of the test classes appear in the ImageNet 1K dataset, to ensure no data leakage. This helps us to evaluate accurately not only the performance of the model but also the generalization capability of the model. We partition the datasets based on these proposed splits for the purposes of training and testing our model.

C. Evaluation Metrics

We use Harmonic Mean as the assessment metric in order to assess the performance of models and contrast them with current techniques. The accuracy of the model on seen class and unseen class is calculated separately and the harmonic mean of these values is considered as the final accuracy. Nevertheless, we do report seen class and unseen class accuracy along with the harmonic mean so that further inferences can be drawn. Per-class accuracy (Acc_C) of both seen and unseen classes is calculated as:

$$Acc_C = \frac{1}{C}\sum_{i=1}^{|C|} A\,cc_{C_i} \tag{4}$$

where the accuracy of the model Acc_C on test samples of class C_i is calculated using the formula:

$$Acc_{C_i} = \frac{\#\left(pred\left(x_{C_i}^{(j)}\right)==y_{C_i}^{j}\right)}{|C_i|} \tag{5}$$

Average accuracy on all classes is considered the final per-class accuracy of the model. To calculate the harmonic mean, we first calculate accuracy on unseen classes, denoted as $GZSL_U$, using (5). Similarly, we calculate seen class accuracy, denoted as $GZSL_S$. Harmonic Mean, which is the final accuracy, denoted as $GZSL_H$, is calculated using the formula:

$$GZSL_H = \frac{2\times GZSL_U\times GZSL_S}{GZSL_U+GZSL_S} \tag{6}$$

Additionally, we used calibration stacking to assess these

techniques, where the calibration value depends on the dataset.

VI. Results

In Table II, we produce the performance of Swin-GZSL on Generalised Zero shot classification on the three selected datasets. Because this is "generalised", images from both unseen classes and seen classes can be seen during the testing. Along with the performance of Swin-GZSL, these results showcase the performance of some of the earliest approaches that tried to solve GZSL, along with some other well-known, and recent approaches. We compare our results against the results of ViT-ZSL, which is the state-of-the-art method at the time of writing this paper.

Swin-GZSL has achieved a better harmonic mean than the current state-of-the-art, ViT-ZSL, on AWA2 and SUN datasets. It also performed well and achieved near state-of-the-art accuracy on the CUB dataset. Compared to ViT-ZSL, the accuracy of Swin-GZSL is a little less on seen classes, but the Unseen accuracy of Swin is greater than that of ViT on all three datasets. This essentially means that we have improved the generalisation capacity of the model and reduced the bias, although not completely eliminated it.

VII. Critical Discussion

Swin-GZSL is better suited to solve the GZSL than ViT-ZSL, the current state-of-the-art, because of multiple reasons:

4. It achieved state-of-the-art harmonic mean accuracy of seen and unseen class data on two datasets, and it achieved near state-of-the-art on the third dataset - $< 1\%$ difference
5. Swin-GZSL demonstrates better generalisation and less bias. When compared to ViT-ZSL, it has the advantage of being superior at recognising unseen class data, making it a better choice for real-world use cases.
6. The computational time of Swin-GZSL is less compared to that of ViT-ZSL. As ViT-ZSL computes attention globally, the computational complexity of ViT-ZSL is Quadratic i.e., the computational time increases quadratically with respect to the image. In contrast, the shifted windowing scheme used in Swin-GZSL brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection [32]. Hence, the computational complexity of Swin-GZSL grows linearly with respect to image making this a better choice both in terms of speed and accuracy.

Vision Transformers partition an input image into distinct, uniformly sized patches and processes those patches independently. Because of this, it is possible that information at the boundaries of the patch will be lost, which will cause them to perform poorly on tasks that involve a fine-grained evaluation of the pixels inside patches. This issue does not arise in Swin Transformer because Swin does not process patches independently. To understand how computing local attention instead of global attention causes such a significant difference in computational complexity when compared to a Vision transformer, let's look at how a Swin Transformer operates.

A. Computational Complexity

Let us consider that a standard RBG image of size 224×224, across 3 channels is sent as input to Swin-GZSL. This image is now segmented into 4×4 size patches resulting in 48 pixels per patch. In Swin-GZSL, we have used a large variant of the Swin transformer which has a linear embedding of size 192, which means that each patch will be represented in by an embedding vector of length 192, resulting in 3,136 patches. Each image is broken into non-overlapping windows such that each window contains $M \times M$ patches, which in our case is 7×7. This results in the entire image getting divided into 64 windows with 49 patches in each window. Here is where local Attention comes into picture - Attention is only calculated using those 49 patches within a window. This computation of local attention does not take into account any other patches that are located beyond the boundaries of the window in which this patch is located. This means that only $2401(49 \times 49)$ dot products will be calculated per patch. As the number of patches is fixed, the computational complexity becomes linear to image size. For a sequence of length N and window size M, the complexity of local attention would be $O(M \times N)$ while the complexity of global attention would be $O(N^2)$.

Separately, from our experiments using the machine A-6000 with 48GB GPU and 45GB RAM, we have found that the time taken to train Swin-GZSL is significantly less than that of ViT-ZSL. While ViT-ZSL needs to be trained for 80 epochs to be able to reach the state-of-the-art performance, Swin-GZSL needs only 15-20 epochs of training to be able to achieve similar levels of performance. To put this into perspective, for a total training that spans over epochs, ViT-ZSL takes approximately 6.5 hours on CUB and SUN datasets and 8.5 hours on AWA2 dataset. In contrast, Swin-ZSL takes < 15 minutes on CUB and SUN datasets and < 25 minutes on AWA2 dataset. This difference in the number of epochs makes a huge impact because their computational complexities are also different as explained above.

In [32], the authors perform a comparative analysis on the speed-accuracy trade-off of classification task of Vision Transformer and Swin Transformer, which is shown in Table III. All the backbones were pre-trained on ImageNet-22K dataset and performance was evaluated on ImageNet-1K. The Swin backbones perform better than their Vision Transformer while computing less floating-point operations per second. From this, we can infer the difference in computational requirements of both architectures. Both Base and Large variants of Swin outperform their counterparts in Vision Transformer. Another important difference between ViT-ZSL and Swin-GZSL is that ViT-ZSL completely abandons biases that make it translation invariant. They are not designed to look for translation invariance but instead to learn these inductive biases through the training process. But Swin-GZSL, on the other hand, uses a relative position bias when computing attention that preserves certain translation invariant features [32]. This is also why ViT-ZSL needs to be trained longer to start performing well as it has to learn on its own that these features are important.

TABLE II
QUANTITATIVE RESULTS: GZSL PERFORMANCE OF VARIOUS METHODS ON THE DATASETS AWA2, CUB AND SUN

Models	AWA2			CUB			SUN		
	S	U	H	S	U	H	S	U	H
ConSE [45]	90.60	0.5	0.99	72.20	1.60	3.13	39.90	6.80	11.60
SSE [35]	82.50	8.10	14.80	46.90	8.50	14.40	36.40	2.10	4.00
LATEM [39]	77.30	11.50	20.00	57.30	15.20	24.00	28.80	14.70	19.50
ALE [9]	81.80	14.00	23.91	62.80	23.70	34.41	33.10	21.80	26.30
GAZSL [44]	86.50	19.20	31.42	60.60	23.90	34.28	35.70	21.70	26.70
SELAR [36]	78.70	32.90	46.40	76.30	43.00	55.00	37.20	23.80	29.00
f-CLSWGAN [43]	64.40	57.90	59.60	57.70	43.70	49.73	36.60	42.60	39.37
IIR [42]	83.20	48.50	61.30	52.30	55.80	53.00	30.40	47.90	36.80
AREN [38]	79.10	54.70	64.68	63.20	69.00	65.97	40.30	32.30	35.90
f-VAEGAN-D2 [40]	76.10	57.10	65.24	75.60	63.20	68.85	50.10	37.80	43.10
APN [37]	78.00	56.50	65.50	69.30	65.30	67.20	34.00	41.10	37.60
DAZLE (Official) [2]	75.70	60.30	67.13	59.60	56.70	58.10	24.30	52.30	33.20
CADA-VAE [3]	75.00	55.80	63.99	53.50	51.60	52.53	35.70	47.20	40.65
ViT-ZSL [2]	90.00	51.90	65.84	75.20	67.30	71.03	55.30	44.50	49.32
Swin-GZSL (Our Proposed Method)	82.43	59.69	69.24	71.93	68.78	70.32	53.19	46.52	49.63

S and U denote the accuracy of the models on Seen classes and Unseen classes respectively. H denotes the harmonic mean of both Seen and Unseen classes (S and H)

VIII. CONCLUSION

In this paper, we proposed an approach - Shifted Window Attention via Swin Transformer - to solve the GZSL problem to identify and classify images. Our approach uses the shifted-window based attention module for relating visual and semantic attributes. For linking visual and semantic features, our unique technique utilizes a shifted-window-based attention module. Our results on the datasets AWA2 and SUN datasets show that we achieved the state-of-the-art in terms of harmonic mean and on AWA2 and SUN datasets. On CUB dataset, we achieved near-state-of-the-art accuracy. Additionally, Swin-GZSL also demonstrated reduced bias due to its superior capacity to generalise and perform better on unseen classes. Not only does our model perform better but it also is computationally less expensive.

Even though the model attained state-of-the-art results, the harmonic mean accuracy of the model on unseen classes of two datasets is just around 50%. Although the bias is reduced, it is far from being completely eliminated. Building ensemble models that leverage the capabilities of both generative and embedding learning is something that can be explored upon, to reduce bias. Attempts to overcome the issue using meta-learning techniques are also promising directions that can be focused upon.

REFERENCES

[1] B. Goertzel, "Artificial General Intelligence: Concept, State of the Art, and Future Prospects," Journal of Artificial General Intelligence, vol. 5, p. 1–48, 2014.
[2] D. Huynh and E. Elhamifar, "Fine-Grained Generalised Zero-Shot Learning via Dense Attribute-Based Attention," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[3] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell and Z. Akata, "Generalised Zero- and Few-Shot Learning via Aligned Variational Autoencoders," CoRR, vol. abs/1812.01784, 2018.
[4] F. Alamri and A. Dutta, "Multi-Head Self-Attention via Vision Transformer for Zero-Shot Learning," CoRR, vol. abs/2108.00045, 2021.
[5] L. Zhang, T. Xiang and S. Gong, "Learning a Deep Embedding Model for Zero-Shot Learning," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017.
[6] Dinu et al., "Improving zero-shot learning by mitigating the hubness problem", ICLRW, 2015.
[7] Radovanovic et al., "Hubs in Space: Popular Nearest Neighbors in´ High-Dimensional Data", JMLR, 2010.
[8] Paul et al., "Semantically Aligned Bias Reducing Zero Shot Learning", CVPR, 2019.
[9] Z. Akata, F. Perronnin, Z. Harchaoui and C. Schmid, "Label Embedding for Image Classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 7, p. 1425–1438, 2016.
[10] Z. Akata, S. E. Reed, D. Walter, H. Lee and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, 2015.
[11] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein and B. Schiele, "Latent Embeddings for Zero-Shot Classification," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016.
[12] R. Socher, M. Ganjoo, C. D. Manning and A. Y. Ng, "Zero-Shot Learning Through Cross-Modal Transfer," in Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013.
[13] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim and X.-Z. Wang, "A Review of Generalised Zero-Shot Learning Methods," CoRR, vol. abs/2011.08641, 2020.
[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model," in Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013.
[15] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado and J. Dean, "Zero-Shot Learning by Convex Combination of Semantic Embeddings," in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.
[16] G. Dinu and M. Baroni, "Improving zero-shot learning by mitigating the hubness problem," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, 2015.
[17] Y. Yu, Z. Ji, Y. Fu, J. Guo, Y. Pang and Z. Zhang, "Stacked Semantics-Guided Attention Model for Fine-Grained Zero-Shot Learning," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montreal, Canada, 2018.

[18] Y. Liu, J. Guo, D. Cai and X. He, "Attribute Attention for Semantic Disambiguation in Zero-Shot Learning," in 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, 2019.

[19] Y. Liu, L. Zhou, X. Bai, Y. Huang, L. Gu, J. Zhou and T. Harada, "Goal-Oriented Gaze Estimation for Zero-Shot Learning," CoRR, vol. abs/2103.03433, 2021.

[20] F. Alamri and A. Dutta, "Implicit and Explicit Attention for Zero-Shot Learning," in Pattern Recognition - 43rd DAGM German Conference, DAGM GCPR 2021, Bonn, Germany, September 28 - October 1, 2021, Proceedings, 2021.

[21] S. Chen, Z. Hong, Y. Liu, G.-S. Xie, B. Sun, H. Li, Q. Peng, K. Lu and X. You, "TransZero: Attribute-guided Transformer for Zero-Shot Learning," CoRR, vol. abs/2112.01683, 2021.

[22] V. K. Verma, K. J. Liang, N. Mehta and L. Carin, "Meta-Learned Attribute Self-Gating for Continual Generalised Zero-Shot Learning," CoRR, vol. abs/2102.11856, 2021.

[23] V. K. Verma, A. Mishra, A. Pandey, H. A. Murthy and P. Rai, "Towards Zero-Shot Learning with Fewer Seen Class Examples," in IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021, 2021.

[24] Y. Yu, Z. Ji, J. Han and Z. Zhang, "Episode-Based Prototype Generating Network for Zero-Shot Learning," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, 2020.

[25] N. V. Nayak and S. H. Bach, "Zero-Shot Learning with Common Sense Knowledge Graphs," CoRR, vol. abs/2006.10713, 2020.

[26] F. Li, Z. Zhu, X. Zhang, J. Cheng and Y. Zhao, "From Anchor Generation to Distribution Alignment: Learning a Discriminative Embedding Space for Zero-Shot Recognition," CoRR, vol. abs/2002.03554, 2020.

[27] G.-S. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin and L. Shao, "Region Graph Embedding Network for Zero-Shot Learning," in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV, 2020.

[28] Y. Annadani and S. Biswas, "Preserving Semantic Relations for ZeroShot Learning," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 1822, 2018, 2018.

[29] Y. Liu, Q. Gao, J. Li, J. Han and L. Shao, "Zero Shot Learning via Low-rank Embedded Semantic AutoEncoder," in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, 2018.

[30] G. Liu, J. Guan, M. Zhang, J. Zhang, Z. Wang and Z. Lu, "Joint Projection and Subspace Learning for Zero-Shot Recognition," in IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019, 2019.

[31] Z. Ji, H. Wang, Y. Pang and L. Shao, "Dual triplet network for image zero-shot learning," Neurocomputing, vol. 373, p. 90–97, 2020.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, 2021.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," CoRR, vol. abs/2010.11929, 2020.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," CoRR, vol. abs/1706.03762, 2017.

[35] Z. Zhang and V. Saligrama, "Zero-Shot Learning via Semantic Similarity Embedding," in 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015.

[36] S. Yang, K. Wang, L. Herranz and J. van de Weijer, "On Implicit Attribute Localization for Generalised Zero-Shot Learning," IEEE Signal Process. Lett., vol. 28, p. 872–876, 2021.

[37] W. Xu, Y. Xian, J. Wang, B. Schiele and Z. Akata, "Attribute Prototype Network for Zero-Shot Learning," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 612, 2020, virtual, 2020.

[38] G.-S. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao and L. Shao, "Attentive Region Embedding Network for Zero-Shot Learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019.

[39] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein and B. Schiele, "Latent Embeddings for Zero-Shot Classification," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016.

[40] Y. Xian, S. Sharma, B. Schiele and Z. Akata, "F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019.

[41] C. H. Lampert, H. Nickisch and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, 2009.

[42] Y. L. Cacheux, H. L. Borgne and M. Crucianu, "Modeling Inter and Intra-Class Relations in the Triplet Loss for Zero-Shot Learning," in 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, 2019.

[43] Y. Xian, T. Lorenz, B. Schiele and Z. Akata, "Feature Generating Networks for Zero-Shot Learning," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018.

[44] Y. Zhu, J. Xie, Z. Tang, X. Peng and A. Elgammal, "Semantic-Guided Multi-Attention Localization for Zero-Shot Learning," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019.

[45] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado and J. Dean, "Zero-Shot Learning by Convex Combination of Semantic Embeddings," in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[46] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[47] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[48] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zeroshot learning—a comprehensive evaluation of the good, the bad and the ugly," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.