Zero-Shot Learning: Towards the Effortless Classification of Mystical Creatures.

Student Name: Max Woolterton
Supervisor Name: Yang Long
Submitted as part of the degree of Msci Natural Sciences to the
Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—

Context: With the evolving variety of applications of Computer Vision, and the ever-increasing number of classes within, there is a growing need to be able to transfer the prior-knowledge of a pre-trained model from classes seen during its training to novel classes introduced at test-time. Zero-shot learning does exactly this, via the use of auxiliary/side data-sources, such as manually-annotated class-attributes, or class-relationships extracted from Knowledge Graphs. This semantic-information is then leveraged to transfer prior-knowledge from the seen classes to the unseen classes, allowing new classes to be detected without any need for re-training.

Method: Zero-shot learning methods that use Knowledge Graphs as auxiliary data are less explored in literature, and are hence the focus of our studies. The pre-existing ZSL-KG framework is explored, ablated and improved within this paper. Most notably, we experiment with the use of EfficientNets as backbone ConvNet models for Zero-shot learning, and provide a comparison with the more popular use of ResNets. A comparison of Numberbatch vs. GloVe word-embeddings in various settings is also provided, alongside extensive ablation studies of our final model.

Results: The result of our works is a new variant of ZSL-KG, which we name ZSL-KG+ and evaluate on the AWA2 benchmark. This model achieves a new **state-of-the-art performance** in the **ZSL setting of 82.1%** Top-1 accuracy (an **improvement of 4.0%** accuracy over the previous state-of-the-art model), while maintaining a competitive performance of 70.8% harmonic mean accuracy in the GZSL setting. Even more impressively, this is all achieved alongside a **reduction of 58% in the size of the model** and a **decrease of 35% in prediction-time**, making ZSL-KG+ ideal for **low-budget time-critical real-world deployments**. Experiments on a small custom Pegasus dataset are also run, showing that this approach can be limited by the density of the Knowledge Graph in niche areas. Finally, a **front-end interactive ZSL demo framework** aimed at improving the interpretability the model is delivered.

Index Terms—Generalized Zero-shot Learning, EfficientNet, Common Sense Knowledge Graphs, ConceptNet, Numberbatch, TrGCN

1 Introduction

Imagine a world in which you have never heard of a Pegasus. You do not know what the name refers to, and, most importantly, you have never seen any (pictures of) Pegasi. However, you have encountered and observed many horses and birds throughout your past. Now imagine someone describes to you what the mythical creature known as a Pegasus looks like. They tell you that it is a flying horse, with large plumed wings. From this, you could extract some attributes that a Pegasus should possess. Namely, it looks like a horse, can fly, has a large wing-span, and has feathered wings. Subsequently, you are asked to look at Figure 1 and point out which images are birds, which are horses, and which are Pegasi. You, as a human, would easily be able to transfer your prior knowledge about horses and birds to point out that to the top three images are the Pegasi, based on the semantic description of the mythical creature provided to you. This is the concept that Zero-shot Learning (Lampert et al., 2009) is based upon and attempts to mirror.

To achieve this task, traditional Machine Learning image-classification techniques would have to be retrained on an entirely new data set of images of Pegasi. Furthermore, this dataset would have to contain as many Pegasi as any other class, to avoid data-imbalance problems. However, what if images of this new class are hard to come by, such as in the case of a mythical creature. The harvesting of this new dataset could be problematic and expensive. Therefore, the benefits of being able to transfer a model's prior-knowledge to this task, with no need to retrain, and no need to harvest new images, become obvious.

Zero-shot Learning (ZSL) is a sub-field within Transfer Learning (Pan and Yang, 2009). In ZSL, some form of auxiliary/side information is used to transfer knowledge from the classes trained upon (the "seen" classes), to the brand-new classes introduced at test time (the "unseen" classes). Specifically, ZSL is a form of heterogeneous Transfer Learning (Wang et al., 2019), as the source and target feature-spaces are the same, but the label-spaces are distinct. The auxiliary information is usually a form of meaningful semantic-embedding such as textual descriptions (as provided in our imaginary Pegasus scenario), class-attributes vectors (as extracted from our textual description in our scenario), or word2vec word-embeddings (Mikolov et al., 2013, 2014) of the class name. Additionally, the semantic relations between (the seen and unseen) classes can also be used, often in the form of a Knowledge Graph (KG), as in this paper. However, approaches of this form are less commonly explored in the literature (see §2.5). In our scenario, this corresponds to relating the Pegasus to both a horse and a bird, allowing the model to transfer knowledge from those two classes. For an excellent description of the formal mathematics behind ZSL, see Wang et al. (2019), or for a slightly less formal approach, see Romera-Paredes and

Fig. 1: Top row (left to right): three images of Pegasi taken from our custom Pegasi dataset §4.5, one above the ocean (black), one in a field (black), and one above the clouds (white). Bottom row (left to right): An image of a white flying bird, a black horse and different white flying bird.

Torr (2015).

Furthermore, let us revert to our imaginary scenario. You are now asked to point out which Pegasi are in the sky and which are above the sea, or which are white and which are black. You would be able to perform these tasks with ease, provided you have seen these characteristics before (equivalent to the model being trained on a ZSL dataset containing these attributes). In traditional image-classification, you would have to re-partition your dataset into these individual categories (provided these categories are even contained in your dataset), and then re-train your model. Additionally, say you want to recognize ten different variants of Pegasi images, e.g. black, white, flying, grounded, sea backdrop, sky backdrop... You have now divided the size of each class by 10, once again introducing a data-imbalanced problem within the classes, while drastically reducing the amount and variance of training samples for each class (introducing over-fitting). However, in ZSL it would be possible to supply these new attribute-vectors to the model and produce 10 new classifiers for these different fine-grained classes without any re-training necessary.

There are additional problems within machine learning that ZSL helps to alleviate, most of which centre around the difficulty of collecting sufficient labelled instances for desired classes (Wang et al., 2019). For example, although human beings can recognize upwards of 30000 classes (Biederman, 1987), existing data-sets only cover a small subset of these classes, leaving many of the rarer classes with no labelled instances to train classifiers upon. Additionally, to obtain sufficient labelled instances of new, changing, or rare classes can be prohibitively expensive and time-consuming. These problems are all worsened in more niche applications with sparser and smaller datasets, for example, in pose-estimation (Wang et al., 2019). Furthermore, the use of ZSL allows real-time applications to deal with new classes appearing after the learning-stage, during deployment, a feat that would otherwise be impossible.

In the real-world, ZSL has been shown to have a variety of applications (see the survey of Wang et al. (2019) for individual works, alongside popular datasets in the computer vision-based tasks). For example, in computer vision, ZSL has been applied to image-recognition, image-generation and image-segmentation of unseen classes, pose-estimation for novel poses, person re-identification from unseen camera angles and domain adaptation for recognizing objects from unseen domains. There are also applications outside of computer vision, for example, in language-tasks, Nayak and Bach (2021) applied ZSL to intent-classification and fine-grained entity typing on language excerpts.

Many ZSL approaches focus on the use of class-attribute's as auxiliary information. However, class-attributes are not available for many datasets, and annotating these datasets is expensive and can take thousands of hours of expert manual labour (Zhao et al., 2019). Furthermore, it might be difficult and/or sub-optimal to describe classes via manual-annotation in some applications.

One of the more recently proposed (Wang et al., 2018), and less commonly explored approaches is to utilize KGs to obtain semantic relations between classes, while using word2vec word-embeddings to obtain a semantic description of each individual class. With this being a novel new direction in ZSL, in this project, after having conducted a thorough literature review, we selected one of the recent state-of-the-art expansions of this original idea and attempted to expand their works. Namely, we chose Nayak and Bach's (2021) ZSL-KG framework, due to its increased flexibility over other surveyed works. For example, ZSL-KG sets a new state-of-the-art performance in a variety of language datasets, whereas other works focus exclusively on image-classification.

Therefore, the aims of this project were set up as follows:

1) To explore and develop a deep understanding of the field of ZSL;
2) Re-create the works of Nayak and Bach (2021) and have a working implementation of ZSL-KG;
3) Implement a simpler baseline model to compare with (ES-ZSL; Romera-Paredes and Torr, (2015));
4) Improve upon the results of ZSL-KG in the Animals with Attributes 2 (AWA2; Xian et al., (2018)) benchmark by proposing a new variant of the framework, namely ZSL-KG+;
5) Develop a novel method of encompassing class-attribute data into the ZSL-KG+ framework;
6) Ablate the performance of ZSL-KG+ to develop and show an understanding of which constituent parts and auxiliary data-sources are important for the model and ZSL more generally;
7) Develop a live front-end graphical user interface to act as an interactive demo and learning resource for newcomers to the field and experts alike.

During the project, all deliverables bar **5)** were fully delivered upon.

We will now provide an overview of the project's achievements, quoting the related deliverable number in bold at the end of each paragraph.

Our exploration and understanding of ZSL is demonstrated in §2. **(1)**

ZSL-KG was **successfully re-implemented** and **expanded** to create ZSL-KG+. The notable changes are as follows: an **EfficientNet-B4** (Tan and Le, 2019) is used instead of a ResNet101 (He et al., 2016); to reduce over-fitting, the GNN is trained for 500 rather than 1000 epochs; the backbone is fine-tuned for zero rather than 25 epochs; ZSL-KG+ is evaluated using **two choices of word-embedding source**. **(2)**

Furthermore, informal experimentation was carried out into changing the architectural design of TrGCN and its hyper-parameters. These experiments showed that the hyper-parameters and architectural design choices from Nayak and Bach (2021) were **close to optimal**, as no statistically significant improvement could be found.

Overall, ZSL-KG+ sets a **new state-of-the-art performance of 82.1%** Top-1 accuracy in the ZSL setting on the AWA2 benchmark (vs. ZSL-KG's previous state-of-the-art 78.1%), **surpassing all other methods** surveyed within the field by a **considerable 4.0%**. ZSL-KG+ maintains competitive performance of a harmonic mean accuracy of 70.8% in the GZSL setting (7th highest of all methods surveyed, see Table 1 for more details). **(4)**

Although a novel method of encompassing class-attribute data into the model was not implemented, two such methods were proposed in §4.6. These methods were not implemented as they were deemed infeasible with the current dataset, thus the project explored more meaningful directions. **(5)**

Ablation studies breaking down the performance of ZSL-KG+ are presented in §4.4. Specifically, the contributions of the KG, the contributions of word-embeddings, the choice of word-embedding source, and the quality and parametric-size of ConvNet back-bone were all evaluated. **Such evaluations one not found in any of our surveyed works**. Additionally, in order to evaluate the contributions of the KG, a framework using **a novel Neural Network was proposed** to learn from word-embeddings only. **(6)**

Furthermore, the baseline model was implemented and then improved in the GZSL setting by applying Calibrated Stacking (Chao et al., 2016). This improved ES-ZSL's **harmonic mean accuracy** from **8.4% to 32.5%**. **(3)**

The interactive demo was delivered in the form of a website, implemented in the Django Python framework, with **live instances** of ZSL-KG+ and ES-ZSL **connected through the back-end** (see §4.7, Figure 8). This website uses a variety of metrics, statistics and tools to **improve the explainability of the model**, help beginners understand attribute-based ZSL and KG-based ZSL, and allow experts to analyse the models' performance. Notably, a **GradCAM** (Selvaraju et al., 2017) **heatmap is overlaid** onto the images to aid all three objectives (see Figure 9). **(7)**

To summarize, we propose a new framework named ZSL-KG+, which surpasses all other methods surveyed by a considerable **4.0% accuracy** in the ZSL setting, while achieving a **58% reduction** in **model-size** and a **35% reduction** in **prediction-time**, allowing for cheaper, faster and more accurate real-world deployments. We then extensively ablate this model in **ways not found in literature**. Finally, we deliver live-demo framework designed to increase the interpretability of models.

2 Related Work

In this section we try to show the reader a journey of the evolution of ZSL from some of its earlier roots up to some of the cutting-edge algorithms which have been proposed in recent years, many of which are shown in Table 1. As a thematic choice, any papers for which algorithm names are provided in this section are also shown and compared in terms of their performance in Table 1. For works which explore some unusual use cases of ZSL beyond standard zero-shot image recognition, see the appropriate paragraph in the introduction, §1. It is of note that most methods surveyed in this section could also be applied/adapted to such use cases, and are merely evaluated on the most popular benchmarks, which are in zero-shot image recognition.

2.1 Early attribute-based ZSL

In ZSL, some form of auxiliary information is required to transfer and generalize from the seen-class knowledge learned in training onto the unseen-classes during testing. These classes will often be varied and distinct from those classes trained upon, and, after all, the model cannot classify something it knows nothing about.

One of the most popular choices of side-information, especially in early ZSL, is class-attributes. These attributes should be commonly observed and semantically meaningful visual properties of the objects within classes. For example, in AWA2, ”brown”, ”stripes”, ”furry”, ”longneck” and ”jungle” are all attributes. One of the most popular choices of side-information, especially in early ZSL, are class-attributes. These attributes should be commonly observed and semantically distinguishable visual properties of the objects within classes. For example, in AWA2,

"brown", "stripes", "furry", "longneck" and "jungle" are all attributes.

Early works of ZSL leverage attributes in a 2-stage approach to classify a given image (Kankuekul et al., 2012; Lampert et al., 2009, 2013). Firstly, the input image's attributes are predicted (by learning individual attribute classifiers), and then the nearest-neighbour in the semantic (attribute) space is selected as the predicted class. The two earliest and simplest approaches are called Direct Attribute Prediction (DAP) and Indirect Attribute Prediction (IAP) (Lampert et al., 2009, 2013). Unfortunately, these two-stage models suffer from domain shift (Fu et al., 2015), as the learned intermediary task is to learn attribute-classifiers, while the target task is to predict the class-label.

Therefore, later works began directly embedding either: (1) the visual features into the semantic space, (2) the semantic features into the visual space, or (3) both the visual and the semantic features into a third space, in each case for nearest-neighbour-based classification using end-to-end set-ups (Wang et al., 2019; Xian et al., 2018). For example, Li et al. (2017) investigated how to better align the visual and semantic spaces by creating an optimized semantic space. In all of these approaches, the intuition is that there exists a possible mapping such that the visual samples may be mapped to be distributed about their class's semantic counterpart in the target space.

Most of the above methods, once the visual and semantic spaces have been aligned, use an L^1 or L^2 based nearest-neighbour classifier to determine the most likely class-label. However, this approach treats all dimensions of the attribute/semantic space as equally important, which is likely sub-optimal, as attributes will have varying importance when discriminating between classes. Therefore, Romera-Paredes and Torr (2015), using the framework introduced by Akata et al. (2013), explores directly modelling the relationship between image-features, attributes and classes using a 2-layer linear model. This model is called ES-ZSL, and is discussed in §3.6.

Although class-attributes are a proven form of auxiliary information in ZSL which have been used since the conception of ZSL (Wang et al., 2019; Xian et al., 2018), they are often not available for the desired classes in real-world applications, and crafting them can take up to thousands of hours of expert manual labour (Zhao et al., 2019). To bypass to this problem, Rohrbach et al. (2010) explores mining these attributes from a variety of internet-sources. Alternatively, the field of attribute-learning could be explored to solve this problem (Russakovsky and Fei-Fei, 2010). Additionally, Rohrbach et al. (2010) also investigates directly mining the relationships between attributes and classes, and goes even further by directly mining the relationships between classes without ever touching on attributes (Rohrbach et al., 2013, 2010).

However, attributes are not the only auxiliary data-source that can be used for ZSL. Theoretically, any data-source that provides some form of semantic information about a class has something to offer within ZSL. For example, Frome et al. (2013) projects images into a word-embedding (see §3.4) space for comparison. Even further, Akata et al. (2015) evaluates combinations of four different auxiliary data-sources, namely, attributes, two word-embedding spaces and the WordNet (Miller, 1995) KG. Another novel data-source was used in Karessli et al. (2017), where the semantic-space was constructed from human's gaze tracks when observing the classes' images. Controversially, Mensink et al. (2012) does not use any side-information, and instead uses a 1-shot image to construct its semantic-space, however this is arguably closer to few-shot learning than it is zero-shot learning.

2.2 Generalized ZSL:

So far, all approaches discussed have been evaluated in the ZSL setting, where the possible labels in the prediction-stage may only belong to unseen classes. This setting has been criticized for being unrealistic for real-world scenarios (Scheirer et al., 2012) and thus steering the performance-driven research within the field of ZSL in the wrong direction. Therefore, Scheirer et al. (2012) proposed the Generalized Zero Shot Learning setting, where both seen and unseen classes can be predicted during testing.

Unfortunately, when new or pre-existing methods began to be evaluated in the more realistic GZSL setting, it became evident their (unseen) performance was considerably degraded by the inclusion of seen classes in the possible labels at the prediction stage (Xian et al., 2018). This is because, although the models are capable of competently distinguishing the unseen classes from each other, they tend to over-predict and over-fit to the seen classes (Chao et al., 2016). This results in relatively higher scores being outputted for the seen classes classifiers than for the unseen, overall resulting in frequent misclassifications due to the bias introduced by this over-fitting. Chao et al. (2016) studied this effect and introduced a method to counter it, which we employ in this paper, named Calibrated Stacking. Zhang et al. (2018) also presents an empirical analysis of this effect, naming it the class-level over-fitting problem (CO), wherein models generalize well to unseen samples of seen classes, but poorly to those from unseen classes. Zhang et al. (2018) also proposed possible mitigation for some classical ZSL algorithms.

With the proposal of GZSL starting to gain more interest, Xian et al. (2018), an excellent survey of ZSL up to 2017, re-evaluates many of the aforementioned models in the GZSL setting on a variety of benchmarks. Xian et al. (2018) simultaneously points out many of the flaws of prior ZSL papers/conventions that violate the zero-shot assumptions, such as using pre-trained ConvNets to extract image features that have been trained on a portion of the test classes. Xian et al. (2018) thus introduces a standardized evaluation protocol to avoid these many "Bad" and "Ugly" aspects of early ZSL. For example, they released standardized train, test and validation splits for all prior datasets to be used which avoid violating any zero-shot assumptions, and standardized the use of the harmonic mean as the main metric in the GZSL setting. Xian et al. (2018) also introduces the AWA2 dataset and benchmark (which is used in this paper), which contains only public copyright images, unlike AWA1 (Lampert et al., 2013), thereby allowing new feature extraction methods to be tested on an ongoing basis by the research community.

Wang et al. (2019) also provides an excellent survey of ZSL up to 2018, along with setting out the mathematics of

ZSL in great depth, although does not touch on GZSL in much detail.

From 2018 onwards, the vast majority of papers evaluate their works in both the ZSL and GZSL settings.

2.3 Transductive ZSL:

Another direction explored in ZSL is transductive ZSL. Wherein, unlike the methods mentioned so far, which are all inductive, unlabeled examples (visual and/or semantic) of test classes are assumed to be available at train time (Fu et al., 2015; Paul et al., 2019; Song et al., 2018; Xu et al., 2021, 2017). This helps reduce CO in GZSL, for example by improving image-feature-representation in a ConvNet backbone for test classes. However, the transductive setting has been argued to be unrealistic (Verma et al., 2020), as during training test data is often not available, or if it is, it is available in small quantities only. Additionally, Xian et al. (2018) claims that it is not fair to compare transductive with inductive methods, due to the observed benefit deriving from the relaxation of the zero-shot assumptions. Thus, we focus on inductive approaches only within this paper.

2.4 Generative ZSL:

One of the more cutting-edge methods of countering CO in GZSL is the application of generative models to generate synthetic examples of the unseen classes. These examples can either be generated at the image-level or the feature-level (as would be extracted by a ConvNet). These methods first learn a conditional generative model, which synthesizes visual features conditioned on a class's semantic descriptor. These synthetic visual features then act as pseudo-examples of the unseen classes, effectively reducing the ZSL task to a supervised classification problem. Overall, this strategy enables the learning of semantic-visual alignment on unseen classes implicitly, and thus improves the model's ability to classify unseen classes (Chou et al., 2020). The generative models are generally based on Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), for example Chen et al. (SP-AEN, 2018), Felix et al. (2018) and Paul et al. (SABR-I, 2019), or Variational Autoencoders (VAEs) (Kingma and Welling, 2013), for example Mishra et al. (CVAE-ZSL, 2018), Verma et al. (SE-ZSL, 2018) and Schonfeld et al. (CADA-VAE, 2019).

A prominent issue with such generative models is that, as they are trained only on seen samples, the quality of the synthesized unseen samples is highly dependent on the quality, quantity and variation within both seen classes and their samples. Unfortunately, within ZSL datasets or applications, there is often an insufficient amount of data to train a good generative model, overall hindering performance in the final of ZSL task. Additionally, GAN-based approaches are prone to the modal-collapse problem (Arjovsky et al., 2017), which greatly hinders their generalization ability. The result of the over-fitting caused by both of these problems is that often, some semantic features that are particularly unimportant/prominent within the training set are either discarded or over-relied upon within the generation of image features (Chou et al., 2020).

Therefore, several papers explore different ways of countering this, and regularizing the generative model. for example, Felix et al. (2018) enforces visual-semantic feature consistency by employing an additional reconstruction loss. This requires the model to be able to reconstruct the semantic features using the synthesized visual features alone. This ensures that all semantic features are accounted for in the synthesized images. Furthermore, Verma et al. (2020) (ZSML) investigates the use of a *conditional* Wasserstein GAN (Arjovsky et al., 2017), which helps reduce the modal-collapse problem in traditional GANs by employing the use of the Wasserstein loss (Arjovsky et al., 2017). Verma et al. (2020) also introduce a novel meta-learning (Finn et al., 2017) based episodic training strategy of splitting the training set into disjoint train and validation classes each episode. Overall, this set up incentives high generalization power and the ability to learn from very few samples per class (Verma et al., 2020).

However, even with these mitigations, the success of these approaches is still highly dependent upon whether there is enough data and variation in the seen classes to generate diverse visual features (Chou et al., 2020). Otherwise, the yielded features will be too close to those of the seen classes and thus may potentially hinder GZSL, instead of assisting.

This phenomenon inspired Chou et al. (2020), a work fusing both adaptive and generative learning into ZSL, to generate synthetic visual features for not only the given unseen classes but also for a whole range of ”virtual” (unseen) classes. These virtual classes are produced by sampling and interpolating over the semantic representations of all seen classes. This provides a wider spectrum of unseen scenarios to improve the generalization of the model. This effectively transforms the sparse space of classes into a continuous distribution over attribute combinations. Overall, the approach proposed by Chou et al. (2020), named AGZSL, is the highest performing model surveyed in the GZSL setting on AWA2.

In most datasets with class-attribute annotations, for example AWA2 and aPY, the annotations are generated by having experts annotate a variety of images from each class, and then taking the mean over all annotated images in a class (either leaving this as a float, or thresholding this value into a binary ”yes” or ”no”). From the observed range in attribute entries' values when represented as floats, it is clear that not all images in a class possess the same attributes. Furthermore, phenomena such as occlusion, view orientation and different backgrounds cause additional intra-class variations in image features (Chou et al., 2020).

Most methods only account for inter-class variations, modelling only the classes' ”mean” representations, and ignore these intra-class variations. However, Chou et al. (2020) also explores the possibility of image-adaptive class representations via the application of image-specific, per-class attention to the semantic features. The produced semantic-representation thus models intra-class variations. Additionally, they learn two classification models, one for seen classes, and the other for unseen classes. The seen class expert is trained with all data (and allowed to predict any label), whereas the unseen expert is trained only using virtual-classes' images (and allowed to predict only unseen labels). The latter process resembles meta-learning (Finn et al., 2017). This allows the unseen expert to specialize in its generalization capabilities, reducing any over-fitting to

the seen classes, while the seen expert acts somewhat as an anomaly detector (alongside categorizes the seen samples). Both experts project the image-adaptive semantic feature vectors to the visual space and use cosine similarity to predict the class label.

However, Han et al. (2021) argues that the visual space used for classification in many generative models lacks discriminative semantic information and is thus sub-optimal for GZSL classification. Thus, they propose the fusion of an embedding model on top of a generative model to create a hybrid GZSL framework, CE-GZSL. Furthermore, they propose a contrastive embedding, capable of offering both class and instance-wise supervision via the use of a contrastive loss (Khosla et al., 2020). Various other recent works explore the benefits of using a contrastive loss in ZSL (Anwaar et al., 2021; Wang and Jiang, 2021).

While most of the above generative works focus on applying variants of VAEs and GANs to synthesize their unseen samples, the recent work of Shen et al. (2020) is the first to investigate the incorporation of flow-based generative models in an invertible end-to-end set up. In their framework, the forward pass of the flow network learns factorized data embeddings of semantic and non-semantic factors, while the reverse pass generates the synthetic data samples.

2.5 Knowledge-graph based ZSL:

Most of the surveyed works thus far have focused their efforts on manipulating and learning from semantic embeddings of classes in the form of manually annotated class-attributes vectors. A notable shortcoming of these approaches, in addition to the associated cost of producing such annotations, is that they do not use (or model) any explicit relationships between classes. The use of these relationships allows the borrowing of statistical strength from related classes (Wang et al., 2018). With the introduction of Graph Convolutional Network (GCN) (Kipf and Welling, 2016), knowledge graphs (KGs; a data structure/source capable of representing such class relationships), have received much attention within ZSL in recent years.

Aside from Deng et al. (2014), who created exclusion rules relating both attributes and classes from a KG (such as a dog cannot be a cat), Wang et al. (2018) was the first paper to utilize knowledge graphs in ZSL. Wang et al. (2018) applied a 6-layer deep GCN to the KG to learn visual classifiers for unseen classes in a model-of-models approach (Larochelle et al., 2008), using a pre-trained ConvNet's classifier layer as the ground-truth. Their approach, named GCNZ, utilizes both categorical relationships between classes, extracted from the WordNet KG (Miller, 1995), and the semantic meanings/embeddings of each class, in the form of their name's word-embedding, extracted from GloVe (Pennington et al., 2014). This side data is used to train the GCN to produce the weights of the final linear classifier layer of a pre-trained ConvNet classifier trained on ImageNet, such as a ResNet (He et al., 2016). Hence, by supplying an unseen class as input to the GCN, a zero-shot classifier could be produced to be applied to the pre-trained image-features extracted by the ConvNet. It is of note that in this approach, the aggregated learning of the pre-trained ConvNet classifier is also an important side data-source.

However, GCNs were originally designed for classification, whereas within GCNZ the GCN is used for the arguably more complex task of regression (Kampffmeyer et al., 2019). Recently, Li et al. (2018) showed that GCNs perform a form of Laplacian smoothing, whereby as the network depth increases, all nodes' feature-representations converge. This is beneficial in the classification case. However, this smoothing is detrimental for the regression setting (Kampffmeyer et al., 2019), as the GCN is being used to pass information between nodes in the graph to create distinct classifier weights. Kampffmeyer et al. (2019) illustrated this phenomenon in practice, by creating a 1-layer-deep variant of GCNZ called SGCN, with improved results (see Table 1). It is notable that this is contrary to Wang et al.'s (2018) claims that an increased network depth is crucial in their task, with their 6-layer deep network outperforming their 2 and 4-layer deep networks. Unfortunately, limiting the depth of the GCN to 1 layer means that information cannot be propagated from distant nodes in the network, as the GCN can only "see" 2-hops away. Therefore, Kampffmeyer et al. (2019) proposed a dense connectivity scheme where additional connections are inserted into the graph between a node and all of its descendants and ancestors. Overall, this counters the smoothing problem, however it removes much of the structural information contained in the graph, by effectively flattening it. Therefore, these new connections are weighted based on their distance away from the original node. This led to the creation of a second model, named Dense Graph Propagation (DGP), which achieves the second-highest harmonic mean accuracy of all surveyed methods. Kampffmeyer et al. (2019) also introduced the process of freezing the linear weights and fine-tuning the backbone to improve image representation on unseen classes, as performed in our paper and in Nayak and Bach (2021). It is of note that the works of neither Wang et al. (2018) nor Kampffmeyer et al. (2019) require the use of any manual class-attribute annotations, and can thus be applied to any image data-set in the world, assuming their class names are contained within the WordNet hierarchy.

However, if some classes are missing from the graph, the interesting works of Zhao et al. (2017) apply ZSL to extend a KG to additional entities.

In a similar vein, in the case that there does not exist a knowledge graph (containing sufficient relations between classes), Liu et al. (2020) proposes an end-to-end framework to generate a full graph, given pre-defined semantic embeddings (i.e. class annotations). Simultaneously, after observing that the robustness of ZSL heavily relies on the quality of the attribute-space, Liu et al. (2020) investigates optimizing the semantic-space, similarly to Li and Wang (2017). Liu et al. (2020) does so by leveraging inter-class relations to create more powerful class-representations in a framework called Attribute Propagation Network (APNet). APNet comprises a graph propagation model that generates the custom attribute vectors, alongside a (meta-learning based) parameterized nearest-neighbour classifier, which projects images into this optimized semantic-space for classification.

While Liu et al. (2020) removes the need for a pre-

existing KG, Liu et al. (2021) goes even further by additionally removing the need for a pre-trained ConvNet. Unlike Wang et al. (2018) and Kampffmeyer et al. (2019), although Liu et al. (2021) requires neither a pre-existing KG nor a pre-trained ConvNet's weights to use as ground-truth, it does still require class annotations. However, the Isometric Propagation Network (IPN) framework they introduce could be adapted to remove this requirement, at the cost of performance. Additionally, instead of a GCN, a Graph Attention Network (GAT) (Velickovic et al., 2017) is used to propagate information across the graphs. IPN actually generates and uses two separate graphs, one to link the classes in the semantic space, as in Liu et al. (2020), and one to link them in the visual space. The visual embedding of a class is created via a linear projection of its mean visual-feature-representation. The two graphs are then created based on the similarity of classes within the respective spaces. Specifically, the attention scores between classes from the GAT are what are thresholded to create the respective KG-like structures. The final class-representation then becomes the concatenated visual and semantic class-representations produced after the final layer of propagation of the IPN. A nearest-neighbour-based classifier using a custom learnable distance-metric is then used on a projection of the images features to classify images.

Finally, Nayak and Bach (2021) takes the works of Kampffmeyer et al. (2019) and expands them even further, creating a framework they call ZSL-KG. These are the works we expand upon within this paper. Firstly, both Wang et al. (2018) and Kampffmeyer et al. (2019) require access to the whole graph during training, and must be re-trained when additional nodes are added. Conversely, ZSL-KG can be applied to and predict with new portions of the graph without any need to re-train. Additionally, the need to have access to the entire graph during training puts considerable memory constraints on these algorithms, meaning they can both only be applied to small graphs, such as ImageNet's subset (22k nodes) of WordNet (155k nodes). Whereas, ZSL-KG is trained on ConceptNet (Speer et al., 2017), which has over 8 million nodes. Nayak and Bach (2021) achieves this by defining local-neighbourhoods via simulating repeated random-walks across the graph (see §3.3 for details). This advantage is what allows Nayak and Bach (2021) to incorporate not only the relations between classes into their model, but also the relation between classes and any other concepts within the graph (see Figure 3). This allows the rich, high-level information contained in ConceptNet to be used as auxiliary information, alongside word-embeddings of all of these individual concepts. Furthermore, unlike Kampffmeyer et al. (2019), ZSL-KG is not limited to only acyclic directed graphs, as no parent-child relationships are required.

To construct their semantic space, the concepts connected to the classes' nodes are aggregated using their novel transformer-based (Vaswani et al., 2017) graph convolutional network, named TrGCN (see §3.5), designed to replace the original GCN. TrGCN makes extensive use of self-attention and multi-layer perceptrons to allow the model to learn non-linear permutation-invariant aggregations which capture the complex structure of the KG (in simpler words, to pull in knowledge from a large section of graph into a singular vector, see §3.5). Furthermore, Nayak and Bach (2021) shows an empirical performance improvement of TrGCNs over GATs (which are used in IPN).

Notably, Nayak and Bach (2021) is one of the only surveyed papers to show the flexibility of their proposed method outside image-based object-classification, by applying their framework on various language tasks, namely fine-grained entity typing and intent classification, defining new state-of-the-art results in both.

Our work builds on top of that of Nayak and Bach (2021), by successfully incorporating EfficientNets (Tan and Le, 2019) into ZSL and proceeding to empirically study and tweak many of the architectural and design choices of the original framework to produce a new state-of-art model.

3 Solution

In this section, we introduce our variant of ZSL-KG, which we have named ZSL-KG+. Firstly, we provide a high-level overview of how ZSL-KG+ operates and is trained in §3.1, followed by a layered introduction of the various components of ZSL-KG+ in the remainder of §3. Furthermore, we clarify many of the missing implementation details of the original paper (Nayak and Bach, 2021), while providing more explicit explanations of design choices and their benefits as the audience of this paper is not assumed to have expert ZSL knowledge.

3.1 Architecture and Training Overview:

1) We have a Graph Neural Network (GNN) (see §3.5) that processes a node's local neighbourhood (see §3.3) and outputs the weights of a fully-connected linear layer. These output weights are later used as classifiers on the pre-trained ConvNet backbone's (see §3.2) output features produced from an input image (by replacing the pre-existing weights of the fully-connected layer within the pre-trained backbone model during prediction, see §3.2 for details).
2) This GNN is trained by passing through class names as the input concepts and minimizing the L2 distance between the outputted linear (classifier) weights and the classes' respective fully-connected (classifier) weights in a backbone ConvNet classifier pre-trained on ImageNet (Deng et al., 2009).
3) This allows the GNN to learn the relationship between a class's local neighbourhood within the KG (which contains large amounts of rich, high-level information regarding how the class relates logically to 1000s of other concepts/things in the world, see §3.3 for details) and what physical features an image in the class tends to possess (features extracted by the backbone from the images using convolutions).
4) The GNN is trained on a randomly selected subset of 950 of the 1000 classes in the ILSVRC 2012 (Russakovsky et al., 2015), a labelled subset of ImageNet, with the remaining 50 classes used as the validation set (used to select the best GNN parameters found).
5) The intuition is that the variety and quantity of visual features (in images of ILSVRC) and related

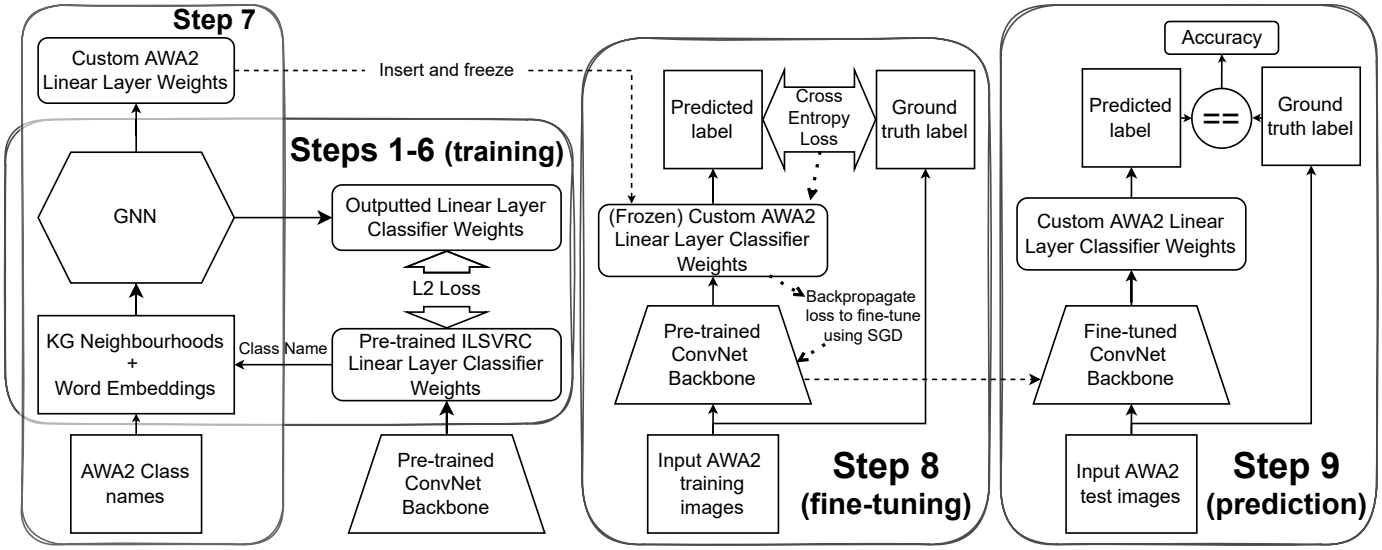


Fig. 2: A diagrammatic representation of the overall training process laid out in §3.1.

concepts (in the KG) of classes in the training set are varied enough such that this GNN can generalize to produce a competent classifier for any inputted class, even those the backbone model has never been trained on. This is exactly what is needed for zero-shot learning.

6) The above GNN is trained for 500 epochs, and the parameters which correspond to the lowest validation loss are saved. This constitutes the "training" phase.

7) All 50 class names from AWA2 are then passed through the saved GNN to generate and save classifier weights for each. Henceforth, these weights are frozen (and this is the last time the GNN is used).

8) Subsequently, the pre-trained backbone model is taken and fine-tuned on the training images within AWA2 (minus the validation split, which is used to tune γ (Chao et al., 2016)). This is called the "finetuning" stage. This stage is performed as only a subset of the seen classes in AWA2 (22 out of 40) are part of the ILSVRC 2012's classes, which the backbone is pre-trained on. Therefore, this process helps to improve image representation of the 18 training classes "unseen" by the backbone. We do so by minimizing cross-entropy loss on the predicted class logits using SGD a learning rate 0.0001 and momentum of 0.9. This is the same method employed by Nayak and Bach (2021).

9) For the prediction stage this fine-tuned model is used alongside the AWA2 test classes' fully-connected weights to produce an output logit for each test class (50 for GZSL or 10 for ZSL), using the original images as inputs. In the GZSL setting, γ is subtracted from the logits of seen test classes to account for the bias of over-predicting seen classes (Chao et al., 2016), as suggested by Xu et al. (2020). As per Nayak and Bach (2021), we calibrated γ on AWA2 validation splits. This is called Calibrated Stacking (Chao et al., 2016). Finally, a softmax is applied to calculate each class's probability, and the maximum output is taken as the predicted class.

See Figure 2 for a diagrammatic representation of steps 1-9.

3.2 The Backbone - ResNets and EfficientNets:

A ResNet (residual neural network) (He et al., 2016), inspired by the human brain, is a type of Artificial Neural Network (NN) that utilizes skip-connections to stabilize and improve the training of very deep NNs. Specifically, skip-connections are used to counter the vanishing gradient and accuracy saturation problems experienced when adding more sequential layers to a NN (He et al., 2016). This has allowed ResNets to be used to achieve state-of-the-art performance on a variety of computer vision tasks, most obviously in image-classification (as by default, a ResNet's final layer is a classifier). A ResNet is made up of many repeated blocks of skip-connections and convolutions of slowly increasing width, and decreasing resolution (until the final layer which is a fully-connected linear layer). However, the rate at which the width and resolution are varied with respect to layer depth was chosen somewhat arbitrarily within He et al. (2016) (and prior works), as they focus solely on optimizing the network depth.

Therefore, Tan and Le (2019) systematically studied the simultaneous scaling and balancing of depth, width and resolution to obtain optimal performance. Using neural architecture search, Tan and Le (2019) proposed a new family of scalable models, called EfficientNets. An EfficientNet can be scaled to achieve increased performance based on the resources available using a compound coefficient (Tan and Le, 2019), which leads to the creation of one of 8 models: EfficientNet-B0, 1, 2, 3, 4, 5, 6 and 7. The largest of which, EfficientNet-B7 achieves new SOTA performance of 84.3% Top-1 accuracy on ImageNet using 8.4x less parameters (66M vs 557M) than the previous SOTA model GPipe (Huang et al., 2019), and is 6.1x faster. Additionally,

to provide context with respect to ResNets, EfficientNet-B1 achieves superior performance (79.1% vs 77.8% accuracy) to the largest ResNet, ResNet-152, while using 7.6x less parameters (7.8M vs 60M, thus allowing us to scale up to theoretically a much higher performance on the same GPU by using a larger EfficientNet). Therefore, contrary to the majority of surveyed models within ZSL, which use a ResNet backbone (or pre-saved ResNet features), in this work the use of the new state-of-the-art EfficientNet architecture within ZSL is trialled (and compared to the prior standard of using ResNets).

Within both EfficientNets and ResNets, the final layer acts as a classifier, taking in the outputted feature map from the final convolutional block as the input features of the image. Therefore, these models can be thought of as being composed of two parts, a feature extractor (the convolutional backbone), and a classifier (the linear layer). In this paper, we initialize the backbone model with pre-trained weights from training on ImageNet, provided by PyTorch. In the training phase, we train a GNN to output custom classifier weights for the final layer to act as a classifier for any given input class. Then, in the fine-tuning stage, we fix the linear weights and fine-tune the feature-extracting backbone's convolutional weights (to improve image representation, as the pre-trained models haven't seen all of AWA2's classes before, see §3.1 for details). Finally, in the testing stage we use the derived custom backbone model (the fine-tuned backbone plus the custom linear classifier weights) as an end-to-end classifier on the unseen images of seen/unseen classes in both ZSL and GZSL settings.

3.3 The Knowledge Graph - ConceptNet:

ConceptNet (Speer et al., 2017) is a multilingual commonsense KG, or a semantic network, designed to represent the meaning of words used in human language and how they (logically) relate to each other. It is used to allow computers to process and understand the knowledge that humans possess about natural language. More specifically, ConceptNet is a directed graph. The nodes, called concepts, represent an individual concept/phrase in a particular language. These concepts are (predominantly) of the form "c/{language}/{concept_name}", for example the concept for cat in English would be $\boxed{\text{c/en/cat}}$. The edges in the graph represent the relations between different concepts, or equivalently, assertions of common-sense connecting two concepts. There are $\sim$ 40 types of relations in ConceptNet. These relations are of the form "r/{relationType}", for example $\xrightarrow{\text{r/Synonym}}$ means A and B have very similar meanings (with A and B possibly in different languages), $\xrightarrow{\text{r/Desires}}$ means A is a conscious entity that typically wants B and $\xrightarrow{\text{r/LocatedNear}}$ means A and B are typically found near each other. The edges in the graph also have weights which represent the strength of the relation. Typically, these are set to 1.0, but they can be higher or lower. For example, $\boxed{\text{c/en/dog}} \xrightarrow{\text{r/UsedFor}} \boxed{\text{c/en/companionship}}$ has weight 6.32, while $\boxed{\text{c/en/dog}} \xrightarrow{\text{r/UsedFor}} \boxed{\text{c/en/biting\_postman}}$ has weight 1.0. Assimilating the above into natural English conveys that a dog is used to provide companionship more often than it is used to bite a postman. By exploring dog's local neighbourhood, you could then combine this knowledge with $\boxed{\text{c/en/dog}} \xrightarrow{\text{r/IsA}} \boxed{\text{c/en/loyal\_friend}}$ (weight 6.63) to infer that a loyal friend is also very likely to provide companionship, despite there being no direct link in the KG between these two concepts. See Figure 3 for an extract of the 1-hop neighbourhood of one of the test classes' concepts in AWA2.

For a machine learning model to understand everything about a concept, we would ideally supply it with the node's entire "neighbourhood". This "neighbourhood" would comprise the concept itself, all the nodes the concept is connected to, alongside all further nodes those nodes are connected to and so on... However, for most nodes this comprises the entire KG. Considering ConceptNet contains over 8 million concepts and 20 million relations, it quickly becomes obvious that passing the entire graph as the "neighbourhood" directly to any model would be infeasible. Therefore, we must come up with a sampling method to determine the most useful or significant concepts linked to a given concept. Define the k-hop neighbourhood of a node i as $\mathcal{N}_{k_{hop}}(i)$ (all nodes/relations pairs that are k hops away from i). As per the studies of Nayak and Bach (2021), (after following some graph pre-processing steps set out in Appendix B of Nayak and Bach (2021)) we chose to generate a concept i's local neighbourhood $\mathcal{N}_N(i)$ as follows:

- Simulate $n = 20$ random walks of length $k = 40$ on the graph $\mathcal{N}_{2_{hop}}(i)$, starting at i, recording the visit

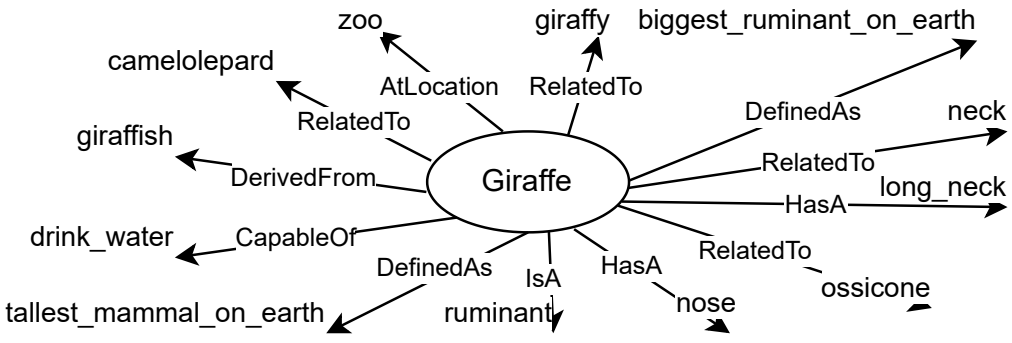


Fig. 3: An extract of the local neighbourhood of the concept $\boxed{\text{c/en/giraffe}}$, one of the test classes in AWA2.

count v_j of each node $j \in \mathcal{N}_{2_{hop}}(i)$;
- As smoothing, add 1 to the visit counts v_j for $j \in \mathcal{N}_{1_{hop}}(i)$ (helps counter the difficulties of high node degree within ConceptNet);
- Calculate hitting probabilities for each node $j \in \mathcal{N}_{2_{hop}}(i)$ as $p_j = \frac{v_j}{\sum_{m \in \mathcal{N}_{2_{hop}}(i)} v_m}$;
- The local neighbourhood $\mathcal{N}_N(i)$ is selected as the N nodes with the highest hitting probabilities p_j.

3.4 The Word Embeddings - Numberbatch and GloVe:

In addition to representing the logical links between concepts, the ConceptNet project also provides machine interpretable semantic vectors (or word embeddings) representing the meaning of the concepts themselves. This data set of word embeddings is called ConceptNet Numberbatch (Speer et al., 2017). Numberbatch was built from the graph-based data of ConceptNet itself, OpenSubtitles 2016 (Lison and Tiedemann, 2016) and two of the most famous word embedding methods/datasets that preceded Numberbatch (both of which it has been shown to outperform on a variety of tasks (Speer et al., 2017)): GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013, 2014). Numberbatch's semantic vectors can be used as an excellent starting point for further machine learning. Contrary to all prior works surveyed (Kampffmeyer et al., 2019; Nayak and Bach, 2021; Wang et al., 2018), who used GloVe, we instead decided to trial the use Numberbatch as our node's starting features, as we believed this could lead to superior generalization and performance. This intuition was drawn based on the increased variety in data sources used to train Numberbatch, alongside the promising results presented in Speer et al. (2017). Both sets of feature vectors are of length 300. We performed pre-processing to prepare the concepts' feature vectors in the same way as Wang et al. (2018), by taking the average of constituent words contained within Numberbatch.

3.5 The Graph Neural Network - TrGCN:

Graphs are a powerful, versatile, and human-interpretable format/structure in which to represent knowledge and store information. However, a notable drawback is that graphs cannot be easily supplied as an input to a NN, as they are not directly machine-interpretable. We require a method to harvest/aggregate the rich information stored in the structure of a graph and represent it as a simple, machine-interpretable vector. This is the idea behind Graph Neural Networks (Hamilton et al., 2017), and this process can be carried out on the graph-level, edge-level, or, as in the case of this paper, node-level, to create graph, edge, or node-embeddings. These embeddings can then be used in downstream ML inference tasks. The embedding/aggregating process can be thought of as synonymous to the feature extraction stage of a ConvNet classifier e.g. ResNet.

There are many possible choices for the design of the GNN, with various recent works (Hamilton et al., 2017; Murphy et al., 2018) focusing on the use of LSTMs (Hochreiter and Schmidhuber, 1997), a form of Recurrent Neural Network (RNN). A prominent issue with LSTMs encountered by Hamilton et al. (2017) is that they are not permutation-invariant to the ordering of the concept's neighbourhood. Thus, the resultant GNN is capable of producing various outputs for the same input concept, some of them producing inconsistent downstream predictions ($\sim$16% of the time in studies conducted by Nayak and Bach (2021) on AWA2). Murphy et al. (2018) investigated leveraging this permutation-invariance by taking an average over the resultant distribution of outputs. This fixed the problem while providing a minor increase in accuracy, at the cost of greatly increased computational cost.

Various previous works used linear aggregations of neighbourhood features, which, although permutation-invariant, fail to capture or model the complex information that can be derived from the KG, alongside having their own individual downfalls.

Therefore, for the aggregator in our GNN, we opted to use the non-linear transformer-based TrGCN (see Figure 4) introduced by Nayak and Bach (2021) which is naturally permutation-invariant and is shown to outperform all previous works on a variety of ZSL subtasks (Nayak and Bach, 2021). In Figure 4, we clear up many of the missing implementation details of TrGCN that were left out in the original paper, such as the plethora of dropout layers present.

Specifically, we use the ZSL-KG framework introduced by Nayak and Bach (2021), training the GNN to produce the classifier weights in a 2-stage iterative aggregation process (see Figure 5). We start with the Numberbatch word embeddings of the class u's name as our initial graph features, $h_u^{(0)}$. To obtain the classifier weights $h_u^{(2)}$, we firstly sample the concept u's local neighbourhood of size 50 (with a self-loop, so size 51 overall; see §3.3 for details) $\mathcal{N}_{50}(u)$. We then calculate and pass a TrGCN module the set $\{h_v^{(1)} \; \forall v \in \mathcal{N}_{50}(u)\}$ to output $h_u^{(2)}$. The $h_v^{(1)}$ are respectively calculated by passing a different TrGCN module the local neighbourhood $\mathcal{N}_{100}(v)$'s respective word embeddings $h_u^{(0)}$, or equivalently the set $\{h_k^{(0)} \; \forall k \in \mathcal{N}_{100}(v)\}$. Note, $\mathcal{N}_{100}(v)$ also includes a self-loop.

However, in order to ablate and evaluate the individual contributions of the KG and word embeddings we also designed a novel NN (see Figure 6) that only uses word embeddings to learn to produce the classifiers. This model directly replaces the GNN in our training setup, and allows us to estimate the performance benefit the KG offers. To evaluate the performance contributions of the word embeddings, we also trained our GNN on (fixed) random concept embeddings $h_u^{(0)}$.

3.6 The Baseline Model - ES-ZSL:

During this project, so as to compare the performance of our KG and class-name word-embedding based model to a class attribute-based model, a baseline model was also implemented. This being the model described in "An embarrassingly simple approach to zero-shot learning" (Romera-Paredes and Torr, 2015), which we refer to as ES-ZSL. However, this is not a model designed for GZSL (see §2). Therefore, as part of this project we adapted ES-ZSL to create a generalized version. In doing so, we noticed that the model suffered from a severe over-prediction bias for

the seen classes, and achieved 4.4% accuracy for unseen test classes, but 92.4% accuracy for seen classes (with a resultant 8.4% harmonic mean accuracy). Therefore, to tackle this, inspired by the countermeasure in Xu et al. (2020), we undertook Calibrated Stacking (Chao et al., 2016) on the validation split (see §3.1 for details). This increased the harmonic mean accuracy to 32.5% (see Table 1)

4 Results and Evaluation

4.1 Experimental Setup:

All experiments were run on either an NVIDIA TITAN Xp or an NVIDIA GeForce RTX 2080 Ti, with all implementation performed in Python 3.8.11 using PyTorch 1.11.0 and allennlp 0.9.0 alongside NVIDIA's CUDA 11.0. All data collection was performed using TensorBoard to allow for real-time analysis and comparison of experiments. Many thanks go to Durham University for the use of their NVIDIA CUDA Centre (NCC) GPU cluster for the testing, tuning and training of all models, clocking approximately 1000 GPU hours over a period of many months. Without the use of the same none of the research contained herein would have been possible.

The data for each experiment was gathered using a sequential-sampling strategy derived using insights from Decision Theory, wherein each experiment was first run on five fixed random seeds, and then, if the noise/disparity between these runs was sufficiently high, data from ten additional (fixed) seeds would be sampled. The majority of experiments showed very similar convergence across different seeds ($\pm 2\%$ harmonic mean accuracy) and thus were not rerun, overall showing a robustness of the model to the choice of random seed. However, the experiments without a GNN showed far larger ranges of convergence ($\pm 6\%$ harmonic mean accuracy), and were thus rerun. At the end of this process the mean overall results across different seeds is taken and analysed using plots aggregated using the Python module tensorboard-reducer. The models are trained as set out in §3.1 & Figure 2, and are evaluated by calculating their class-balanced Top-1 accuracies on the

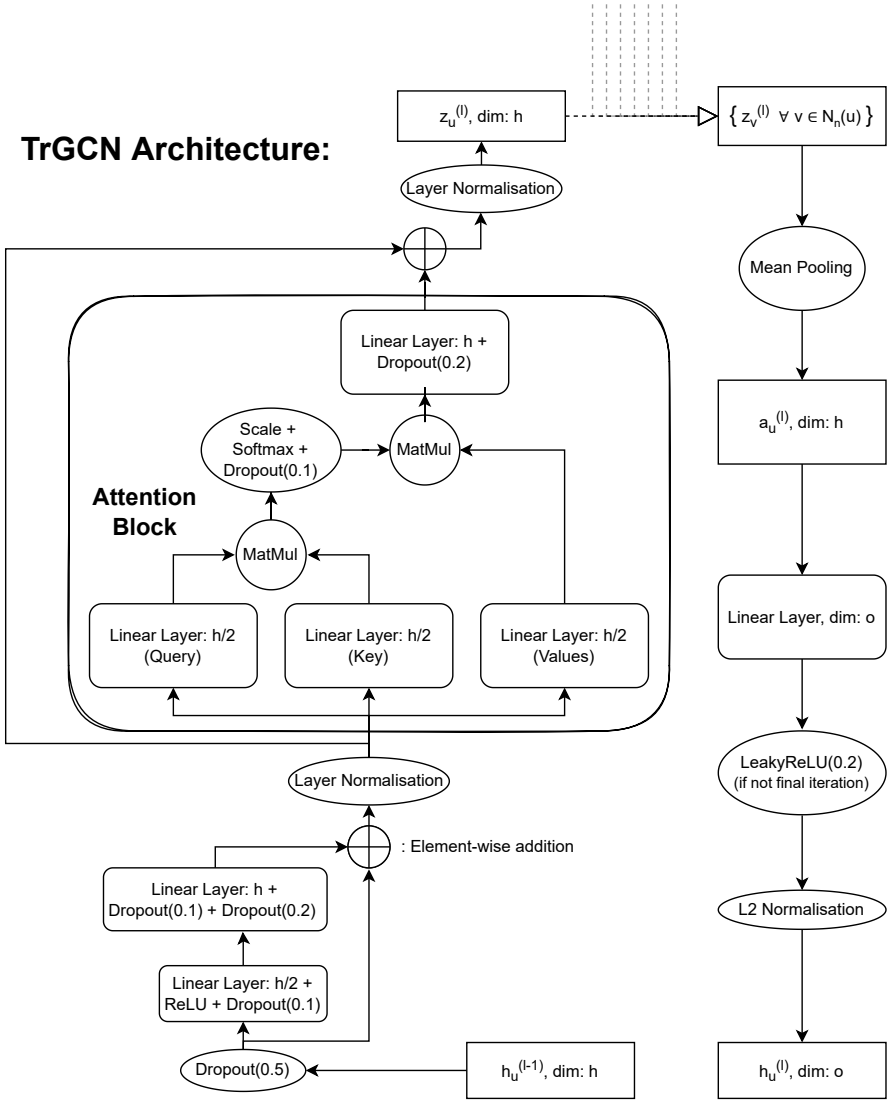


Fig. 4: The structure of a TrGCN (Nayak and Bach, 2021) module for a stage l, with input dimension h, output dimension o and neighbourhood size n. Layer Normalisation layer is taken from Ba et al. (2016).

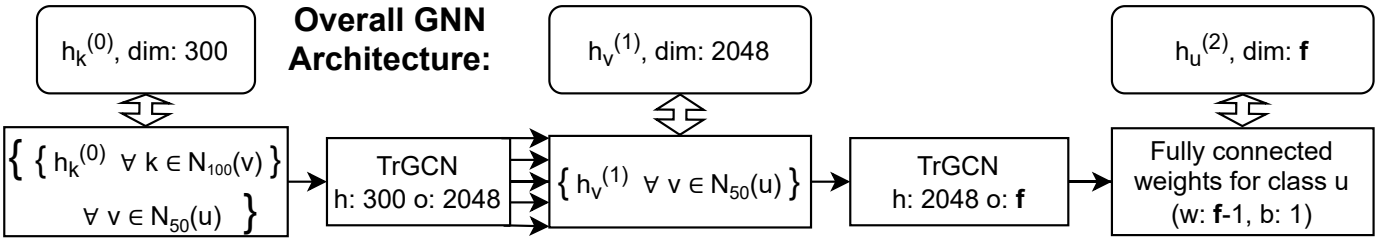


Fig. 5: The ZSL-KG based (Nayak and Bach, 2021) architecture of our GNN used in this paper. This diagram presents a high-level overview of how the GNN is trained to produce the fully connected weights of dimension **f** (dependent on backbone model) to classify a class u. The full details of the TrGCN module can be seen in Figure 4.

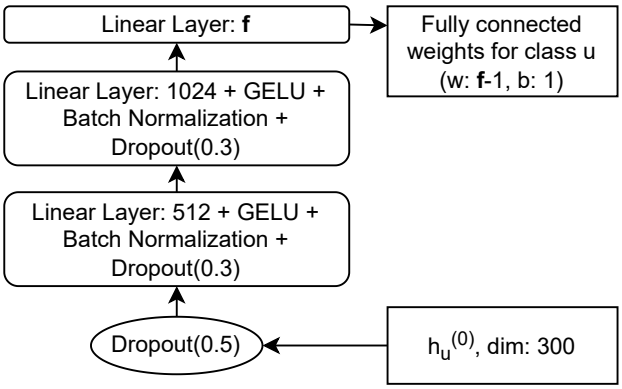


Fig. 6: The novel NN architecture used as a replacement for our GNN shown in Figure 5.

test splits of AWA2. Therefore, in the ZSL setting, only the overall accuracy ($T1$) of unseen classes alone is evaluated. Whereas, in the GZSL setting, the seen (S), unseen (U) and harmonic mean ($H = \frac{2 \cdot S \cdot U}{S+U}$) (Xian et al., 2018) accuracies are evaluated. In the GZSL setting, the most important metric for comparison of models/performance is H, as this encompasses the performance on both seen and unseen classes.

4.2 Implementation Issues / Notes:

Unfortunately, contrary to the claims of Nayak and Bach (2021) (although they publish no data to back these claims up), during experimentation it was noted that the best performance of all models came from epoch 0 of **fine-tuning** (before the model had been fine-tuned). Much experimentation was led into this direction, however no definitive answer was arrived at (the authors were also contacted, but with no reply as of the time of writing). Therefore, within our results section, all experiments' results are quoted at epoch 0. It was noted that the seen accuracy increases, and the unseen accuracy decreases during fine-tuning. Additionally, when γ is tuned on an epoch-by-epoch basis, it is seen to increase, and the disparity between seen and unseen accuracies decreases (however the harmonic mean accuracy still plateaus after an initial decrease). This shows that much of the seen accuracy's apparent amelioration comes from the increased over-prediction of seen classes (which is countered by our use of Calibrated Stacking). Furthermore, the GNN was found to overfit to the validation set when trained for 1000 epochs (as in Nayak and Bach (2021)), which is possible through its evaluation on the validation set every epoch (see §3.1), and superior results were consistently obtained on the test set through training for 500 epochs. Thus, all models were run for 500 epochs, bar the EfficientNet-B6 and EfficientNet-B7 experiments, which were run for 1000 and 2000 epochs respectively. These exceptions were made as their validation losses had yet to converge at 500 epochs, presumably due to the increased width of classifier layers when scaling an EfficientNet.

Additionally, the **calibration of** γ on the validation splits performed by Nayak and Bach (2021), $\gamma = 3.0$ was found to be far from the optimal tuning when using ResNet as the backbone, which, in our studies, was calculated as $\gamma = 2.0$. For EfficientNet-B4, $\gamma = 0.4$ was found to be optimal. This shows that γ is model dependent. It was also noticed that the optimal γ varied throughout fine-tuning epochs and across random seeds, showing that it is also weight-dependent. See Table 2 for the calibrated values of γ. Additionally, the choice and calibration of γ is seen to have a large effect on the final harmonic mean accuracy (see Figure 7), and could be argued to be one of the main weaknesses of ZSL-KG+, and could be the subject of future research.

4.3 Comparison to state-of-the-art methods:

Table 1 compares our best performing (EfficientNet) model, a version of ZSL-KG+ using the customary ResNet101 backbone, and our baseline model ES-ZSL, with various of the best-performing models taken from literature (with a section dedicated to KG-based models). These results show that our best model sets a **new state-of-the-art performance** of a Top-1 accuracy of 82.1% on the ZSL benchmark for AWA2, **beating all other models surveyed within the field**. Furthermore, the table demonstrates that considerable performance gain is obtained through the use of EfficientNet-B4 vs Nayak and Bach's original choice of ResNet101 in ZSL-KG. However, ZSL-KG+ does not reach the same performance in harmonic mean accuracy as the original paper. This is believed to be due to their different experience in applying fine-tuning to the model (evidenced by their considerably higher seen accuracy than unseen in comparison to our higher unseen than seen on the same architecture; see earlier paragraph), and/or that γ was possibly tuned on the test-set (see analysis of Figure 7 for reasoning). However, two **considerable pieces of evidence** suggest that, were the same outcome of fine-tuning to be replicated on ZSL-KG+, ZSL-KG+ **should theoretically outperform** ZSL-KG's

TABLE 1: Comparison of our model with the literature on results for ZSL and GZSL image classification on the AWA2 dataset. We report the class-balanced top 1 accuracy ($T1$) for ZSL and the seen (S), unseen (U) and harmonic mean (H) accuracies for GZSL. The top subsection contains general ZSL methods, the middle KG-based methods, and the bottom methods presented within this paper. Bold-faced entries represent the best models in the table for each metric, whereas underlined entries represent the best models within each sub-category. For a greater range of comparisons, see the Appendix of Nayak and Bach (2021).

Model Name	U	S	H	$T1$
SP-AEN (Chen et al., 2018)	23.3	90.9	37.1	58.5
CADA-VAE (Schonfeld et al., 2019)	55.8	75.0	63.9	-
SABR-I (Paul et al., 2019)	30.3	**93.9**	46.9	65.2
CVAE-ZSL (Mishra et al., 2018)	-	-	51.2	65.8
SE-ZSL (Verma et al., 2018)	58.3	68.1	62.8	69.2
APNet (Liu et al., 2020)	54.8	83.9	66.4	-
CE-GZSL (Han et al., 2021)	63.1	78.6	70.0	70.4
TF-VAEGAN (Narayan et al., 2020)	55.5	83.6	66.7	73.4
IZF-Softmax (Shen et al., 2020)	60.6	77.5	68.0	74.5
ZSML (Verma et al., 2020)	58.9	74.6	65.8	<u>77.5</u>
IPN (Liu et al., 2021)	67.5	79.2	72.9	-
AGZSL (Chou et al., 2020)	<u>69.0</u>	86.5	**76.8**	76.4
GCNZ (Wang et al., 2018)	66.6	81.6	73.3	77.0
SGCN (Kampffmeyer et al., 2019)	67.5	81.2	73.7	77.1
DGP (Kampffmeyer et al., 2019)	<u>71.3</u>	79.4	<u>75.1</u>	77.1
ZSL-KG (Nayak and Bach, 2021)	66.8	<u>84.4</u>	74.6	<u>78.1</u>
ZSL-KG+ (EfficientNet-B4 backbone)	**71.8**	69.9	<u>70.8</u>	**82.1**
ZSL-KG+ (ResNet101 backbone)	71.3	61.8	65.1	78.6
Baseline Model (ES-ZSL)	20.9	<u>73.3</u>	32.5	52.8

harmonic mean accuracy of 74.6% and conceivably surpass the current state-of-the-art Chou et al.'s, 2021 76.8%. These pieces of evidence are that, firstly, ZSL-KG+ outperforms ZSL-KG on the ZSL by benchmark by 4.0% (achieving state-of-the-art performance in the field). Secondly, in our studies, replacing ResNet101 with EfficientNet-B4 alone yielded and increase in seen accuracy of 8.1% and unseen accuracy of 0.5%, resulting in an overall increase in harmonic mean accuracy of 5.7%. We claim these pieces of evidence are strong enough to merit further research, as they suggest that ZSL-KG+, having already broken the state of the art for ZSL, has the **potential to surpass the best performing model** in the **GZSL setting** as well.

It is also of note that, out of the six benchmarks ZSL-KG (Nayak and Bach, 2021) was evaluated on, AWA2 is one of the only two against which it did not achieve new state-of-the-art performance in the GZSL setting.

Furthermore, as is discussed further below, we achieve a considerable **reduction of 58%** in the number of parameters (45M to 19M) and **35% in prediction-time** vs Nayak and Bach (2021) in prediction stage.

4.4 Ablative studies:

In an attempt to attribute the model's performance to that of its constituent parts and/or data sources, ablation experiments are run against various variants of the model and/or training setup mentioned throughout §3.

Namely, the importance of the **quality of the backbone model** is assessed by comparing EfficientNet-B0, 1, 2, 3, 4, 5, 6 and 7 (Tan and Le, 2019) and ResNet50, 101 and 152 (He et al., 2016). ResNet101 is the model used in the original ZSL-KG framework (Nayak and Bach, 2021). During these experiments, to reduce the number of independent variables and provide a fairer comparison, γ is tuned on the test-set at test time (as the tuning of γ is shown to be significant to performance). Thus, these results are technically not ZSL, and serve only to compare different backbones, while also providing figures for the **maximum possible performance of a model**, were a method to tune γ perfectly be developed (the second reason these tests were run in such a setup). The results of this experiment can be seen in Figure 7. Overall, these results show that EfficientNets, when their coefficient is chosen correctly (via *evaluation* on the training images in AWA2), allows ZSL-KG+ to outperform its ResNet-based counterparts while using considerably less parameters. This allows cheaper deployment on smaller devices, and run-times in real-time applications to be cut down considerably. The inconsistency of EfficientNet-B3's performance is not understood, and comparison with other datasets would be needed to form any meaningful deductions. Overall, this plot shows that the **quality and complexity of image features** provided from the backbone model is of **great importance to the final performance** of ZSL-KG+. More specifically, ignoring the outlier of EfficientNet-B3, this plot suggests that there is a hot-spot for the optimal complexity of image features, and that using a model with increased performance on ImageNet does not necessarily lead to increased performance in ZSL. The best backbone model is EfficientNet-B4, which, when γ is tuned on the test-set (which notably varied across random seeds), achieves a H of

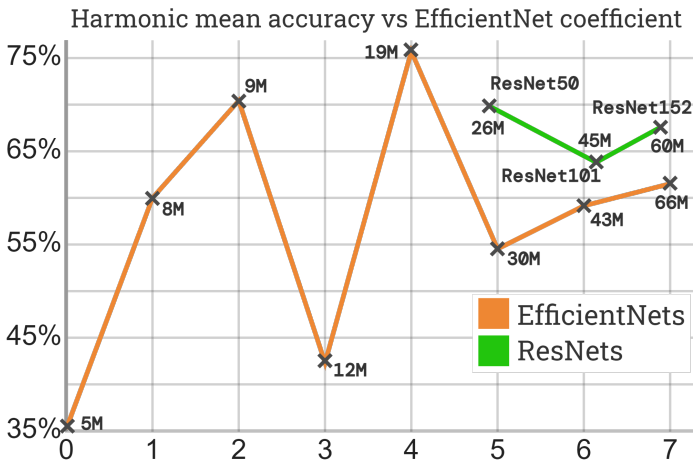


Fig. 7: A plot of ZSL-KG+'s harmonic mean accuracy averaged over 10 seeds for a variety of backbones, with γ tuned per-seed on the test-set. EfficientNet backbones are shown in orange and are plotted against their coefficient, and are labelled with their parameter count in millions. ResNets are plotted relative to their nearest EfficientNet with respect to parameter count.

75.8% (with $U = 78.9\%, S = 72.9\%$). This harmonic mean accuracy is very similar to the published results of ZSL-KG (which we could not replicate), suggesting perhaps γ was tuned on the test-set in Nayak and Bach (2021) (awaiting clarification from authors).

Furthermore, the **contributions of the KG** (and GNN) are assessed via swapping out the GNN with the novel NN design shown in Figure 6 which takes word-embeddings only as input ($h_u^{(0)}$). The results of this experiment can be seen in Table 2, and suggest that **considerable benefit is gained via the inclusion of the KG** as an additional side data-source with which to conduct transfer-learning, as is evidenced by a significant boost in all metrics.

Conversely, both the contributions and the importance of **the choice of word-embeddings** / initial graph features are evaluated via comparing the GNN's ability to learn when $h_u^{(0)}$ are sourced from Numberbatch, GloVe and a random distribution on the interval $[0, 1)$. The results of this experiment can be seen in Table 2. These results show that the **choice and presence of quality word-embedding data is vital** to the GNN's learning capabilities. This is evidenced by the fact that when random word-embeddings are used the model fails to improve beyond random guessing and by the significant disparity between all metrics when two different well-researched word-embedding sources are used. Interestingly it is of note that, contrary to the claims of Speer et al. (2017), the use of **Numberbatch over GloVe did not offer a significant boost** to the performance of the model, and in fact proved detrimental, with lower results in both ZSL and GZSL. We believe that this is due to two factors. We believe that much of the performance gains shown in Speer et al. (2017) come from the inclusion of the graph-based structural data, contained in ConceptNet, into the training data for Numberbatch (see §3.3 for details). Therefore, we hypothesize that firstly, as our GNN is capable of lifting this performance benefit directly (and perhaps better) from ConceptNet instead, the inclusion of ConceptNet graph-based data in Numberbatch is diluting the purity and usefulness of language-based-elements of the word-embedding data. Secondly, as the GNN is connected in an end-to-end fashion to the knowledge graph and the specific downstream task (unlike in Numberbatch which is trained for a general downstream task), it is better able to represent and exploit the data within ConceptNet. Overall, this experiment suggests that the model **would benefit from any further improvements in word-embedding data** in the future (either in general or with respect to this task).

To verify these hypotheses, further experimentation was carried out by comparing the **performance of the novel NN** word-embedding-only model when supplied with **GloVe vs Numberbatch** input features (see Table 2). These experiments **confirm the hypotheses**, as the use of Numberbatch over GloVe considerably lessons the performance gap in the GZSL setting (with an improvement of 9.0% in harmonic mean accuracy) between the word-embedding only model (which has no access to ConceptNet) and the full GNN-based model. However, it is not understood why the use of GloVe is slightly more beneficial in the ZSL setting (by 1.3% Top-1 accuracy).

Informal experiments were also run into changing a variety of TrGCN's parameters, such as the number of attention heads, the number of linear projections, dropout rates, alongside the depth of the GNN and number of nodes in the iterative neighbourhoods. All of these experiments showed that the hyper-parameters and architectural design choices from Nayak and Bach (2021) were **close to optimal**, as no statistically significant improvement could be found.

4.5 Mystical Creature Experiments:

Furthermore, in an attempt to demonstrate the mystical creature based scenario depicted in the introduction, §1, a small custom data set of 20 public-copyright Pegasi images was produced. The "Pegasus" was then added to the test-set classes of AWA2, and custom Pegasi classifier weights were retrieved from the GNN.

Unfortunately, it was noted that there are very few meaningful connections in ConceptNet for `c/en/pegasus`, e.g. no connection to `c/en/horse`, `c/en/horselike`, `c/en/wings` or `c/en/winged_creature` (or any related nodes). Therefore, unsurprisingly, during evaluation the Pegasi classifier performed very poorly (0% accuracy), generally mistaking them for various types of whales. This provides evidence that ConceptNet's knowledge is still far behind that of humans in many of its rarer, less commonly accessed areas. Therefore, we theorize that the expansion of ConceptNet and KGs in general **could offer much benefit** to any future KG-based machine learning applications, especially those exploring more niche concepts. Additional empirical evidence supporting this theory could be gathered in future works by observing any performance detriment of the model when randomly removing a proportion of connections within the KG (e.g. 20%). To conclude, it would be very interesting to run this and/or similar experiments on a more connected KG, such as CSKG (Ilievski et al., 2021).

TABLE 2: Results for ablation studies of ZSL-KG+ on ZSL and GZSL image classification on the AWA2 dataset. We report the class-balanced top 1 accuracy ($T1$) for ZSL and the seen (S), unseen (U) and harmonic mean (H) accuracies for GZSL. Bold-faced entries represent the best model in the table, whereas underlined entries represent the best model within each sub-category (separated by double spacing). This type-facing is performed to aid comparison for the reader.

Backbone	Word Embeddings	Uses GNN + KG	γ	U	S	H	$T1$
ResNet101	GloVe	Yes	2.0	71.3	61.8	65.1	78.6
EfficientNet-B4	GloVe	Yes	0.4	**71.8**	**69.9**	**70.8**	**82.1**
EfficientNet-B4	Numberbatch	Yes	0.35	65.8	69.2	67.4	77.7
EfficientNet-B4	Random	Yes	0.0	2.1	2.1	2.1	10.6
EfficientNet-B4	GloVe	No	0.15	<u>49.9</u>	42.3	44.6	<u>65.2</u>
EfficientNet-B4	Numberbatch	No	0.2	45.2	<u>66.2</u>	<u>53.6</u>	63.9

4.6 Integrating class-attribute data:

One of the initial project deliverables was to investigate the possibility of integrating the class-attribute data provided as part of the AWA2 dataset into the KG-based model in an attempt to achieve superior performance by integrating an additional data source. It is of note that this would however remove one of the main benefits of the KG-based model, namely, that the dataset in question need not be time-consumingly annotated by an expert, making ZSL-KG+ applicable to any labelled image data set in the world.

Unfortunately, it quickly became apparent that due to the training setup of performing transfer-learning from a model trained on the ILSVRC to the AWA2 domain, this was infeasible. This is because one requires the target dataset and the ILSVRC to have the same class annotations supplied in training (this could be a subset of the total available annotations, i.e. the intersection of both domain's annotations). It was discovered that ImageNet does provide annotations of 25 attributes for a subset of 400 of its 21841 synsets. However, only a small portion of 91 of these 400 synsets are contained within the 1000 from ILSVRC. This already limits the available data by over 10x. Furthermore, out of the 80 attributes for AWA2, only 15 have a synonym and or close match. This limits the attributes by more than 5x, thus limiting the "overall data" to 1.7% of its original "size". Thus, it was deemed that, for this particular application, the over-fitting and loss of generalization power (which is essential for transfer-learning) caused by such a drastic reduction in training data would far outweigh any potential benefits of including the attribute data, and so the project continued to explore other more meaningful directions. Additionally, a portion of this attribute data is already included in the KG itself, as demonstrated by the giraffe class's local neighbourhood shown in §3.3 (the giraffe class was randomly sampled from the 50 classes in AWA2 to act as an unbiased example of whether or not attribute data is included in the neighbourhoods of classes). One could also argue that if the attributes themselves are not contained within the local neighbourhoods, then they are either (deemed) not useful enough, or, if they are, the KG itself or the sampling method should be improved, rather than forcibly supplying these (unhelpful) annotations.

Regardless, two methods were formulated in which attributes could be supplied to the model, which could be further explored in future works (ideally on a data set with far more "overlap" to ILSVRC). The first proposed solution is to augment the local neighbourhoods of classes with the positive class annotations' concepts. As the neighbourhoods supplied to the GNN must be fixed size, this would entail replacing the pre-existing concepts in the neighbourhood (the ones with the lowest hitting probability). As the number of positive annotations is class-dependent, and can vary from 0 to 80 (which is higher than the current neighbourhood size), the variation in the ratio of KG-structure-based concepts and attribute concepts could be problematic for learning.

Additionally, this method reduces the quantity of KG-based information supplied to the GNN. However, this could be mitigated by increasing the local neighbourhood size by the mean of the classes' positive annotation count (although this might also require increasing the network size of the GNN due to the increase in supplied information). One of the benefits of this approach is that the meaning of the annotation is also supplied to the model in the form of its word-embedding, which could provide great performance and generalization benefits.

The second proposed solution would be to supply the class annotations as input features to the final linear layer of the GNN (potentially having been projected into another latent-space beforehand via a set of linear layers similar to that in Figure 6), shown before the L2 Normalization in Figure 5. In this method, although the model is never shown the annotation's meaning, if additional linear projections are used, then, for fixed annotations, the model would be able to learn their meaning via training on the inputted images.

4.7 Interactive ZSL-KG+ demo:

An additional deliverable of this project was to deliver an interactive learning resource, in the form of a live website demo for the developed ZSL models (see Figure 8 for a snapshot of the developed website). This demo was to act as a tool for newcomers to the field to explore, experiment with and understand ZSL and the developed models (with many of the design features targeted at improving the explainability of the models). Additionally, this tool was to be used by experts in the field to derive insight into the models' strengths and weaknesses.

This website was developed using the Django framework, and can be retrieved and hosted locally from the project's GitHub repository. This website allows a random image to be sampled from the seen/unseen classes, this image is then passed through both ZSL-KG+ and the baseline model ES-ZSL to retrieve the top 5 predicted class labels

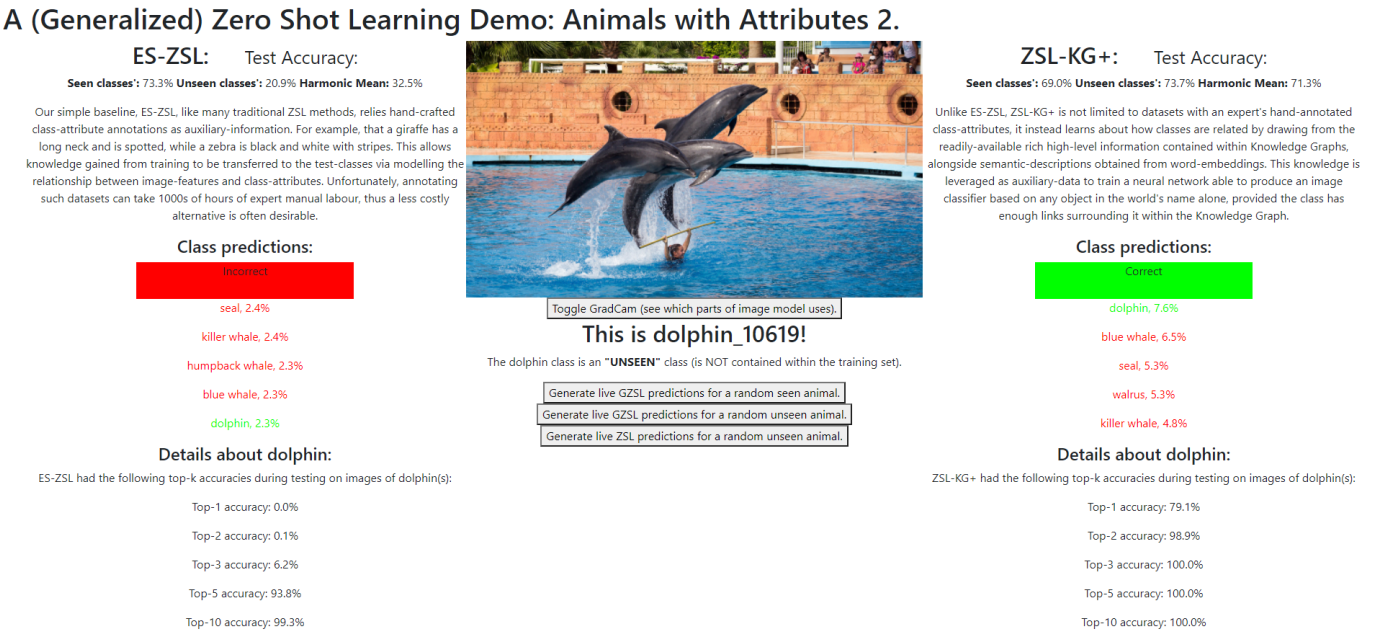


Fig. 8: A snapshot of our demo website developed as an interactive learning resource. ZSL-KG+, with its correct prediction, is shown on the right, while ES-ZSL, with its incorrect prediction, is shown on the left.

alongside their confidence levels. The image, predictions, a high-level description of the workings of both ZSL algorithms, alongside additional statistics related to the class itself (e.g. the models' top n accuracies) and the models overall (e.g. the seen/unseen/harmonic mean accuracies) are then displayed on the webpage. Providing high-level descriptions aids newcomers in understanding both the purpose of ZSL and how the two different approaches differ from each other (attribute-based and KG & word-embedding based). Additionally, the provision of the top five predicted classes enables the analysis of incorrect predictions (especially when combined with the GradCAM feature, see below and Figure 9). Furthermore, the accompanying confidence levels reveal surprising insight when contrasting the deductive power of the two models (ZSL-KG+'s confidence of a positive prediction is often $10\%+$, whereas the baseline model's is rarely above 3%, meaning it is only slightly more certain that the positive class is correct over any given other of the 50 classes). Finally, the displayed top n accuracies for the given class also prove insightful, as often when a class has a low top 1 accuracy, it is because it is consistently mistaken for another similar class, thus its top 2 accuracy is often considerably higher. This phenomena can be observed in Figure 9. The website also has a toggle for switching between ZSL and GZSL.

In an effort to improve the interpretability of the model, GradCAM (Selvaraju et al., 2017) was applied onto the convolutional backbone. This method allows a heatmap (shown in Figure 9) to be overlaid on to the input image that shows the individual sections/features of the input image that were used to predict the outputted class (which proves both useful for the analysis of correct and incorrect predictions). This heatmap overlay can be toggled on and off using the website.

4.8 Qualitative Study:

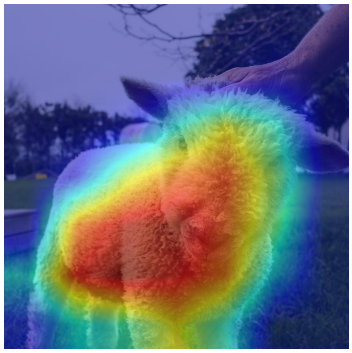
Finally, to demonstrate the power of our interactive demo, a qualitative study was conducted, using the various statistics and GradCAM heatmaps to analyse a variety of correct/incorrect, seen/unseen class predictions. An extract of this study is presented and the corresponding insights discussed in Figure 9.

5 Conclusions

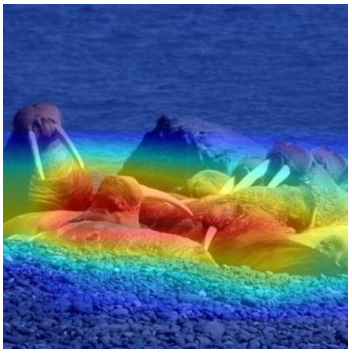
Overall, we propose a new variant to ZSL-KG, which we name ZSL-KG+, resulting in a new state-of-the-art performance in the ZSL setting of 82.1% Top-1 accuracy. This is a significant improvement of 4 points over the previous state-of-the-art model. ZSL-KG+ maintains competitive performance in the GZSL setting, with a harmonic mean accuracy of 70.8%, the seventh highest of any model surveyed. This performance is notably lower than ZSL-KG's 74.6%. However, this is believed to be due to a difference in the application of fine-tuning and/or the zero-shot assumptions when tuning γ (the original authors have been contacted regarding both). Overall, evidence strongly suggests that without these discrepancies, ZSL-KG+ would be able to outperform ZSL-KG in the GZSL setting too, perhaps breaking Chou et al.'s (2020) current state-of-the-art results of 76.8% harmonic mean accuracy.

The most notable change was the incorporation of the innovative works of Tan and Le (2019), by using a far more efficiently designed backbone-model (an EfficientNet-B4). This allows ZSL-KG+ to achieve a higher performance while

(a) Correctly predicted horse.

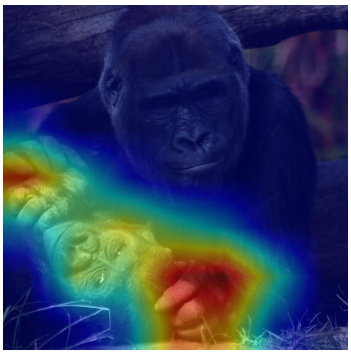


(b) Correctly predicted sheep.

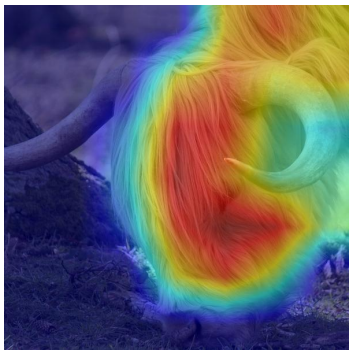


(c) A walrus mistaken for a seal due to their similar shape, surroundings and tendency to group together.

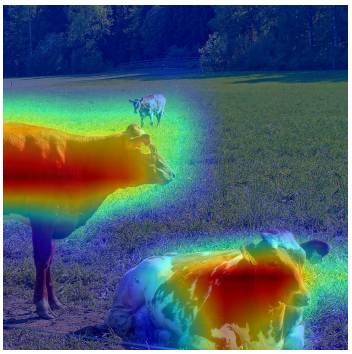
(d) A rat mistaken for a hamster due to its atypical texture and colour for a rat.



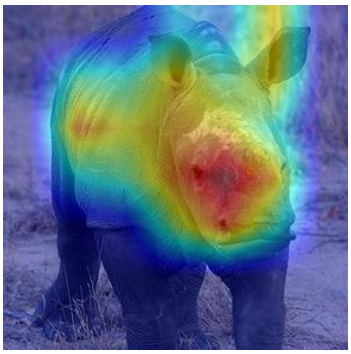
(e) Correctly predicted gorilla, with GradCAM only showing attention on one gorilla.



(f) Correctly predicted ox.



(g) A cow mistaken for an ox/horse due to similar shape, colour and surroundings.



(h) A rhino mistaken with a giraffe / hippo due to similar ears / similar shape, size and colour.

Fig. 9: GradCAM heatmaps taken from our website demo. The top row shows unseen-classes, while the bottom row shows seen-classes. Furthermore, the left-hand-side are correct predictions, while the right-hand-side are incorrect.

maintaining a considerably smaller run-time and memory-requirement, due to the 58% reduction in parameters and 35% reduction in prediction-time vs ZSL-KG. This makes ZSL-KG+ ideal for low-cost time-critical real-world deployments. Extensive ablation experiments were also carried out, comparing the contributions of the KG, the quality and size of the backbone model and the use of two different sources of word-embedding. Finally, a Django-based front-end interactive ZSL website demo was iteratively developed based on user-feedback and delivered.

As in most GZSL algorithms, one of the most prominent issues with ZSL-KG+ revolves around the CO problem of over-predicting seen classes, and our resulting application and calibration of Calibrated Stacking. Therefore, investigating a hybrid approach of applying a novel generative ZSL method on top of ZSL-KG+ could be of interest in the future. As an alternative, an optimized calibration of γ could be further investigated. Furthermore, to consolidate the statistical strength of the claims within this paper regarding the observed improvement of ZSL-KG, re-running experiments on additional benchmarks would be beneficial.

One of the benefits in the approach of ZSL-KG over DGP and GCNZ is that ZSL-KG models the relationship between classes and any other concepts in the graph. However, this means that no explicit relationships between classes are modelled. Therefore, hybridizing both of these approaches into a singular model is another future expansion of this project. This could be performed in a similar fashion to how Liu et al. (2021) integrates their visual and semantic prototypes and graphs. Additionally, integrating visual-embeddings as a separate graph, as in IPN, would also be of interest. Furthermore, it is of note that this approach does not take advantage of the relation types provided within ConceptNet (e.g. $\xrightarrow{\text{r/LocatedNear}}$ or $\xrightarrow{\text{r/IsA}}$), only the presence of a connection, regardless of its type. Therefore, this presents a considerable avenue for further research.

Another direction to take this project in would be to further investigate the integration of the class-attributes into the framework, as explored and set out in §4.6. However, for any results to be statistically meaningful, this would have to be performed on a data-set with a larger attribute overlap with ILSVRC.

Finally, this paper also provided evidence that there are areas of ConceptNet that are poorly connected, such as that surrounding the concept `c/en/pegasus`. This suggests that the application and/or creation of denser knowledge graphs, such as that explored in Ilievski et al. (2021), could benefit future ZSL.

References

Akata, Z., Perronnin, F., Harchaoui, Z. and Schmid, C. (2013), Label-embedding for attribute-based classification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 819–826.

Akata, Z., Reed, S., Walter, D., Lee, H. and Schiele, B. (2015), Evaluation of output embeddings for fine-grained image classification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2927–2936.

Anwaar, M. U., Khan, R. A., Pan, Z. and Kleinsteuber, M. (2021), A contrastive learning approach for compositional zero-shot learning, *in* 'Proceedings of the 2021 International Conference on Multimodal Interaction', pp. 34–42.

Arjovsky, M., Chintala, S. and Bottou, L. (2017), Wasserstein generative adversarial networks, *in* 'International conference on machine learning', PMLR, pp. 214–223.

Ba, J. L., Kiros, J. R. and Hinton, G. E. (2016), 'Layer normalization', *arXiv preprint arXiv:1607.06450* .

Biederman, I. (1987), 'Recognition-by-components: a theory of human image understanding.', *Psychological review* **94**(2), 115.

Chao, W.-L., Changpinyo, S., Gong, B. and Sha, F. (2016), An empirical study and analysis of generalized zero-shot learning for object recognition in the wild, *in* 'European conference on computer vision', Springer, pp. 52–68.

Chen, L., Zhang, H., Xiao, J., Liu, W. and Chang, S.-F. (2018), Zero-shot visual recognition using semantics-preserving adversarial embedding networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1043–1052.

Chou, Y.-Y., Lin, H.-T. and Liu, T.-L. (2020), Adaptive and generative zero-shot learning, *in* 'International Conference on Learning Representations'.

Deng, J. et al. (2009), Imagenet: A large-scale hierarchical image database, *in* '2009 IEEE conference on computer vision and pattern recognition', Ieee, pp. 248–255.

Deng, J. et al. (2014), Large-scale object classification using label relation graphs, *in* 'European conference on computer vision', Springer, pp. 48–64.

Felix, R., Reid, I., Carneiro, G. et al. (2018), Multi-modal cycle-consistent generalized zero-shot learning, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 21–37.

Finn, C., Abbeel, P. and Levine, S. (2017), Model-agnostic meta-learning for fast adaptation of deep networks, *in* 'International conference on machine learning', PMLR, pp. 1126–1135.

Frome, A. et al. (2013), 'Devise: A deep visual-semantic embedding model', *Advances in neural information processing systems* **26**.

Fu, Y., Hospedales, T. M., Xiang, T. and Gong, S. (2015), 'Transductive multi-view zero-shot learning', *IEEE transactions on pattern analysis and machine intelligence* **37**(11), 2332–2345.

Goodfellow, I. et al. (2014), 'Generative adversarial nets', *Advances in neural information processing systems* **27**.

Hamilton, W. L., Ying, R. and Leskovec, J. (2017), 'Representation learning on graphs: Methods and applications', *arXiv preprint arXiv:1709.05584* .

Han, Z., Fu, Z., Chen, S. and Yang, J. (2021), Contrastive embedding for generalized zero-shot learning, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 2371–2381.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Huang, Y. et al. (2019), 'Gpipe: Efficient training of giant neural networks using pipeline parallelism', *Advances in neural information processing systems* **32**.

Ilievski, F., Szekely, P. and Zhang, B. (2021), Cskg: The commonsense knowledge graph, *in* 'European Semantic Web Conference', Springer, pp. 680–696.

Kampffmeyer, M. et al. (2019), Rethinking knowledge graph propagation for zero-shot learning, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 11487–11496.

Kankuekul, P., Kawewong, A., Tangruamsub, S. and Hasegawa, O. (2012), Online incremental attribute-based zero-shot learning, *in* '2012 IEEE conference on computer vision and pattern recognition', IEEE, pp. 3657–3664.

Karessli, N., Akata, Z., Schiele, B. and Bulling, A. (2017), Gaze embeddings for zero-shot image classification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4525–4534.

Khosla, P. et al. (2020), 'Supervised contrastive learning', *Advances in Neural Information Processing Systems* **33**, 18661–18673.

Kingma, D. P. and Welling, M. (2013), 'Auto-encoding variational bayes', *arXiv preprint arXiv:1312.6114* .

Kipf, T. N. and Welling, M. (2016), 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907* .

Lampert, C. H., Nickisch, H. and Harmeling, S. (2009), Learning to detect unseen object classes by between-class attribute transfer, *in* '2009 IEEE conference on computer vision and pattern recognition', IEEE, pp. 951–958.

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

Larochelle, H., Erhan, D. and Bengio, Y. (2008), Zero-data learning of new tasks., *in* 'AAAI', Vol. 1, p. 3.

Li, Q., Han, Z. and Wu, X.-M. (2018), Deeper insights into graph convolutional networks for semi-supervised learning, *in* 'Thirty-Second AAAI conference on artificial intelligence'.

Li, Y. and Wang, D. (2017), 'Zero-shot learning with generative latent prototype model', *arXiv preprint arXiv:1705.09474* .

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

Lison, P. and Tiedemann, J. (2016), Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles, *in* 'LREC'.

Liu, L., Zhou, T., Long, G., Jiang, J. and Zhang, C. (2020), Attribute propagation network for graph zero-shot learning, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 34, pp. 4868–4875.

Liu, L. et al. (2021), 'Isometric propagation network for generalized zero-shot learning', *arXiv preprint arXiv:2102.02038* .

Mensink, T., Verbeek, J., Perronnin, F. and Csurka, G. (2012), Metric learning for large scale image classification: Generalizing to new classes at near-zero cost, *in* 'European Conference on Computer Vision', Springer, pp. 488–501.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), 'Efficient estimation of word representations in vector space'.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2014), 'Distributed representations of words and phrases and their compositionality', *Advances in neural information processing systems* **26**.

Miller, G. A. (1995), 'Wordnet: a lexical database for english', *Communications of the ACM* **38**(11), 39–41.

Mishra, A., Krishna Reddy, S., Mittal, A. and Murthy, H. A. (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', pp. 2188–2196.

Murphy, R. L., Srinivasan, B., Rao, V. and Ribeiro, B. (2018), 'Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs', *arXiv preprint arXiv:1811.01900* .

Narayan, S., Gupta, A., Khan, F. S., Snoek, C. G. and Shao, L. (2020), Latent embedding feedback and discriminative features for zero-shot classification, *in* 'European Conference on Computer Vision', Springer, pp. 479–495.

Nayak, N. V. and Bach, S. H. (2021), 'Zero-shot learning with common sense knowledge graphs'.

Pan, S. J. and Yang, Q. (2009), 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering* **22**(10), 1345–1359.

Paul, A., Krishnan, N. C. and Munjal, P. (2019), Semantically aligned bias reducing zero shot learning, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 7056–7065.

Pennington, J., Socher, R. and Manning, C. D. (2014), GloVe: Global vectors for word representation, *in* 'Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)', pp. 1532–1543.

Rohrbach, M., Ebert, S. and Schiele, B. (2013), 'Transfer learning in a transductive setting', *Advances in neural information processing systems* **26**.

Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I. and Schiele, B. (2010), What helps where–and why? semantic relatedness for knowledge transfer, *in* '2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition', IEEE, pp. 910–917.

Romera-Paredes, B. and Torr, P. (2015), An embarrassingly simple approach to zero-shot learning, *in* 'International conference on machine learning', PMLR, pp. 2152–2161.

Russakovsky, O. and Fei-Fei, L. (2010), Attribute learning in large-scale datasets, *in* 'European Conference on Computer Vision', Springer, pp. 1–14.

Russakovsky, O. et al. (2015), 'Imagenet large scale visual recognition challenge', *International journal of computer vision* **115**(3), 211–252.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A. and Boult, T. E. (2012), 'Toward open set recognition', *IEEE transactions on pattern analysis and machine intelligence* **35**(7), 1757–1772.

Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T. and Akata, Z. (2019), Generalized zero-and few-shot learning via aligned variational autoencoders, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 8247–8255.

Selvaraju, R. R. et al. (2017), Grad-cam: Visual explanations from deep networks via gradient-based localization, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 618–626.

Shen, Y. et al. (2020), Invertible zero-shot recognition flows, *in* 'European Conference on Computer Vision', Springer, pp. 614–631.

Song, J., Shen, C., Yang, Y., Liu, Y. and Song, M. (2018), Transductive unbiased embedding for zero-shot learning, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1024–1033.

Speer, R., Chin, J. and Havasi, C. (2017), Conceptnet 5.5: An open multilingual graph of general knowledge, *in* 'Thirty-first AAAI conference on artificial intelligence'.

Tan, M. and Le, Q. (2019), Efficientnet: Rethinking model scaling for convolutional neural networks, *in* 'International conference on machine learning', PMLR, pp. 6105–6114.

Vaswani, A. et al. (2017), 'Attention is all you need', *Advances in neural information processing systems* **30**.

Velickovic, P. et al. (2017), 'Graph attention networks', *stat* **1050**, 20.

Verma, V. K., Arora, G., Mishra, A. and Rai, P. (2018), Generalized zero-shot learning via synthesized examples, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4281–4289.

Verma, V. K., Brahma, D. and Rai, P. (2020), Meta-learning for generalized zero-shot learning, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 34, pp. 6062–6069.

Wang, J. and Jiang, B. (2021), Zero-shot learning via contrastive learning on dual knowledge graphs, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision', pp. 885–892.

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

Wang, X., Ye, Y. and Gupta, A. (2018), Zero-shot recognition via semantic embeddings and knowledge graphs, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 6857–6866.

Xian, Y., Lampert, C. H., Schiele, B. and Akata, Z. (2018), '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.

Xu, G., Kordjamshidi, P. and Chai, J. Y. (2021), 'Zero-shot compositional concept learning', *arXiv preprint arXiv:2107.05176* .

Xu, W., Xian, Y., Wang, J., Schiele, B. and Akata, Z. (2020), Attribute prototype network for zero-shot learning, *in* H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, eds, 'Advances in Neural Information Processing Systems', Vol. 33, Curran Associates, Inc., pp. 21969–21980.

Xu, X., Hospedales, T. and Gong, S. (2017), 'Transductive zero-shot action recognition by word-vector embedding', *International Journal of Computer Vision* **123**(3), 309–333.

Zhang, H., Long, Y., Guan, Y. and Shao, L. (2018), 'Triple verification network for generalized zero-shot learning', *IEEE Transactions on Image Processing* **28**(1), 506–517.

Zhao, B. et al. (2019), A large-scale attribute dataset for zero-shot learning, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops', pp. 0–0.

Zhao, Y., Gao, S., Gallinari, P. and Guo, J. (2017), 'Zero-shot embedding for unseen entities in knowledge graph', *IEICE Transactions on Information and Systems* **100**(7), 1440–1447.