UNIVERSITY OF LIVERPOOL

Learning Generalized Metrics in Zero-Shot Classification

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Guanyu Yang

Department of Electrical Engineering and Electronics
School of Electrical Engineering and Electronics and Computer Science
University of Liverpool

4th, April, 2022

Abstract

To overcome the practical constraint that the test data should be in the same feature space and follow the same distribution as the training data, transfer learning is proposed to achieve the specific task on a target domain by transferring the task-relevant knowledge from a different source domain. Zero-shot learning, as a sub-field of transfer learning, aims to achieve the classification of target classes without corresponding labelled training samples. It was proposed to imitate the efficient human learning ability that constructs concepts of unknown classes based on relevant descriptions and learned categorical knowledge. To solve the challenging task where samples corresponding to target classes are invalid during training, researchers proposed approaches rely on two main ideas, embedding and generation, respectively. Since the generative methods synthesize pseudo samples for unseen classes based on the corresponding semantic attributes, the followed training process for the classifier might be regarded as breaking the strict target unknown principle. In the inductive scenario where only seen classes are available during training, embedding methods draw more focus as they could learn a target space only depending on the training classes to achieve classification via settled or learned metrics.

With the methods steadily improved, different problem settings, and diverse experimental setups have emerged, the effectiveness of the proposed methods could be inappropriately evaluated. Thereby, in this dissertation, we first provide a comprehensive survey on zero-shot image classification to provide a thorough introduction to this field. Particularly, we have examined three implementation details that can boost the performance of zero-shot learning, i.e. whether the backbone structure has been modified, whether fine-tuning has been conducted, and whether additional knowledge has been used. By annotating these experimental details, we have collected a more careful comparison between various zero-shot methodologies.

The rest part of the dissertation summarizes our work which focuses on improving the metric for the embedding methods under the inductive zero-shot learning scenario. Due to the absence of the labelled target samples in the training stage, the learned embedding space or metrics is easily over-fitted for those seen classes thus leading to the model incorrectly predicting the unseen class as one of those in training when the test label space covered both the seen and unseen classes. To alleviate such an over-fitting problem, we proposed a self-focus mechanism for a ridge regression based method. The proposed mechanism takes the embedded semantic attribute vector as input to produce focus ratios for the dimensions in the embedding space. When these ratios are used for constructing the optimization loss, the correlations between the location and the importance of each dimension are considered. Thus the learned embedding space will be more generalized for classification. However, this mechanism can not be flexibly applied to the methods with learnable metrics. We then, proposed two adversarial frameworks on the sample and parameter spaces, respectively, for the relation network based methods. The designed frameworks help train a robust model on seen data and enhance the sensitivity of unseen classes through adversarial perturbations. As a result, the learned model returns high responses to unseen classes while not affecting the recognition of seen classes due to the robustness.

Key Words: Zero-shot learning, generalized zero-shot learning, embedding methods, knowledge transform, image classification

Contents

List of Figures

List of Tables

Acknowledgment

My research and education would not have been possible without the help and support of supervisors, colleagues, and family. First, I am extremely grateful for the advice and support of my primary supervisor Professor Kaizhu Huang who have helped guide me on both academic and personal decisions in the past several years. His valuable suggestions, incisive comments and constructive criticism have contributed greatly to my research. Furthermore, I want to thank my co-supervisor Dr. John Y. Goulermas for his patience and support in supervising my study, especially in academic writing.

I am also greatly indebted to all the advisors and colleagues in the Pattern REcognition & Machine Intelligence Laboratory: Dr. Qiufeng Wang, Dr. Xi Yang, Dr. Yuyao Yan, Dr. Haochuan Jiang and Dr. Shufei Zhang. Interacting with them always brings me inspiration and positive attitude in my research. And the technical and equipment support brought by them has saved me a lot of trouble.

Last but not the least, I would give the deepest gratitude to my parents for their regretless support and love for me. None of this would have been possible without them.

Chapter 1

Introduction

Machine learning algorithms have been widely applied in real-life applications. However, the achievement of these algorithms is commonly limited by an assumption: the test data should be in the same feature space and follow the same distribution as the training data [1, 2]. In particular, those traditional deep learning methods based on deep neural networks (DNN) perform significantly and even surpass humans in some areas. Still, their performance relies on a large number of labelled training samples. Therefore, it is very time-consuming and impractical to collect a large number of labelled samples and retrain the model for each unique task. Semi-supervised learning can be considered as a solution where a large number of data with only a few of them labelled are employed during training. However, collecting sufficient unlabelled data might also be unrealistic in practice, thus resulting in such a technique being unsatisfactory. To overcome such difficulty, researchers proposed transfer learning, where differences between training and test process may occur in terms of the domains, tasks, and distributions. The main idea of transfer learning is to achieve the target task with the help of relevant labelled data or extracted knowledge from a source domain. Based on whether the sample spaces or label spaces of source and target domains are overlapped, transfer learning can be divided into two categories, homogenous and heterogeneous transfer learning, respectively.

The subject of this dissertation, zero-shot learning, is a subfield of heterogeneous transfer learning where the target labels are disjoint with those in the source data. It is an extremely constrained scenario derived from the few-shot learning. In the few-shot learning, though labelled samples corresponding to the target classes are not available during training, few of them are provided during the test called support set to represent each target class. Specially designed models can be used to extract knowledge or prototypes of the target category to support the corresponding task [3–5]. In contrast, there is no labelled sample corresponding to the target class in zero-shot learning, regardless of the stage of training or test. Specifically, here we call the classes in training and target classes as seen and unseen classes, respectively. Since it is impractical to construct a cognitive concept of an unseen class without any basis, the auxiliary information is introduced to support transferring the cognitive knowledge extracted from the seen classes to the unseen classes. Inspired by the efficient learning process of humans, textual description or manually defined attributes are two commonly adopted auxiliary information in zero-shot learning. In this dissertation, we mainly focus on methods employing the semantic attributes as auxiliary information since the semantic space is consistent during training and the test. Moreover, the descriptions for each class could simply be denoted as a vector holding indicated value for each attribute. It is easier to implement than using textual descriptions since the additional natural language processing model is not required for extracting semantic information.

Among the zero-shot learning ideas, the generative methods can usually achieve impressive results [6–10]. In this kind of method, the task is divided into three steps. First, a generator is trained on training data to generate pseudo samples with the given semantic attributes for specific classes. Then, pseudo samples corresponding to unseen classes could be synthesized with the trained generator and semantic attributes of the unseen classes. By expanding the original training data with the labelled pseudo data for unseen classes, the zero-shot classification is converted to a conventional classification task. Thereby, the last step is to train a common classifier on the expanded data. VAE [11] or GAN [12] based structures with several restrictions on distribution make these generative models perform more generalized to both seen and unseen classes. However, in the most rigorous scenario, namely inductive zero-shot learning, most of these designs become unimplementable since the unseen classes are strictly required to be unknown during the training process.

The embedding methods in zero-shot learning could avoid breaking the strict unknown principle. Depending on the semantic attributes for each class, a specific embedding space is learned where settled or learnable metrics could measure the probability or the similarity of a sample corresponding to a class. Due to the absence of the labelled target samples in the training stage, the learned embedding space or metrics is easily overfitted for those seen classes, thus leading to the model incorrectly predicting the unseen class as one of those in training when the test label space covered both the seen and unseen classes. While some methods construct additional restrictions to prevent such biased estimation of the model during training with the help of semantic information or unlabelled samples of the unseen classes, such designs also are not practicable under the inductive scenario as requiring the knowledge of unseen classes during training. Therefore, it is of great research interest to train a generalized model based purely on training data.

The relation between embedding based zero-shot learning and its hyper research field is shown in Fig. 1.1. In this dissertation, we focus on improving the metrics for those

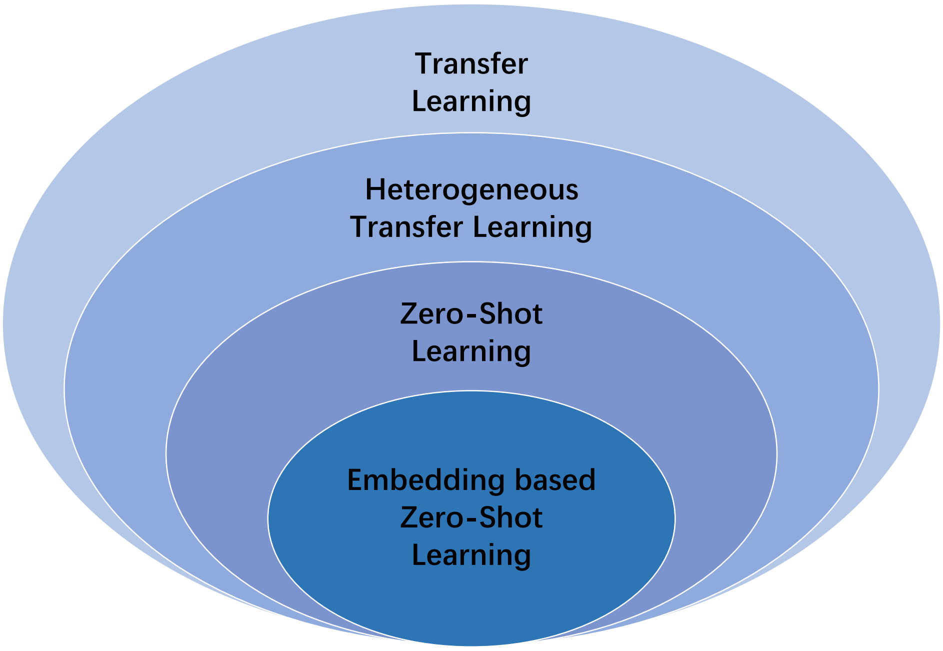

Transfer Learning
Heterogeneous Transfer Learning
Zero-Shot Learning
Embedding based Zero-Shot Learning

Fig. 1.1: The affiliations of embedding based zero-shot learning.

embedding methods aligning the visual and semantic spaces for the inductive zero-shot image classification. In specific, embedding functions for visual samples, semantic attributes, or both are trained to project information from different spaces into a hidden space. With this learned space, the corresponding class for each visual instance is searched as the one with the closest or most similar attributes, thus achieving classification via distance [13] or similarity [14] metrics. The improved metrics can be the ones constituting the objective function during training. Additional regularization or supervised information is considered in such modified metric, thus leading to a more generalized feature space. Alternatively, the improved metric can also be the one applied during the test. Under a specific learning strategy, the modified metric provides more balanced results between training and target labels. Our studies cover these two kinds of improvement where the learned generalized feature space or measurement prevents the prediction from being biased to those training labels.

1.1 Contributions of This Thesis

The research contributions in this dissertation are summarised in the following:

- We present a comprehensive survey on zero-shot image classification where an overview is presented for image classification in zero-shot learning, and the comparison results of various representative methods are collected on a number of benchmarks, aiming to provide a fair and objective reference for evaluating different methods.

  - For a rigorous and detailed review of the task in zero-shot learning, we review and explain some commonly used terms and notations, and define the zero-shot tasks under different learning scenarios.
  - A hierarchical classification for zero-shot image classification methods is proposed to introduce the representative methods for each family according to their based frameworks.
  - For constructing a fairer and more careful comparison between various zero-shot methodologies, three implementation details that can boost the performance of zero-shot learning are examined, i.e. whether the backbone structure has been modified, whether fine-tuning has been conducted, and whether additional knowledge has been used. Annotating these additional beneficial operations or knowledge applied in methods could help provide a more detailed and fair comparison reference for future researchers.

- We propose a self-focus mechanism for a mean squared error based deep embedding model to prevent the learned embedding function from over-fitting to the seen classes.

  - A self-focus module, constructed as a 1-layer fully connected neural network with the activation function sigmoid followed by softmax operation, is developed to generate the focus ratios measuring the importance of the dimensions in the embedding space during training.
  - During the optimization process, the proposed mechanism allows the correlations between the location and the importance of each dimension to be considered.
  - Through the proposed methods, the embedded semantic attributes for unseen classes equip larger gaps to those of seen classes which indicates an alleviation of the over-fitting.

- An adversarial framework is designed for an embedding method with learned similarity metrics. Gradient with respect to the input samples is adopted to make the recognizer more sensitive to the unseen classes and keep the prediction of seen classes roughly consistent.

- A regularization term as the sum of the l2-norm of input samples is adopted to achieve robust learning. This training process could be regarded as an adversarial defence which makes the learned classifier sufficiently robust to small perturbations in the sample space.
    - During the test, the perturbed instance inclined to lead a prediction as unseen is obtained in the neighborhood of the original sample through calculating an adversarial perturbation based on a designed classification loss.
    - The learned model equips the appealing feature that small sample re-adjustment can lead to high responses to unseen classes while not affecting the recognition of seen classes due to the robust adversarial training.

- Following the idea of the previous adversarial framework, instead of adjusting the samples, we adopted such robust defence and attack techniques on the learned parameter spaces.
    - With the same baseline, such an adversarial framework on parameters space achieves better comprehensive performance compared with the one on the sample space.
    - During the test, the perturbed instance inclined to lead a prediction as unseen is obtained in the neighbourhood of the original sample through calculating an adversarial perturbation based on a designed classification loss.
    - The perturbation can be calculated not only for a single instance but also for a group of samples, thus allowing the training samples to support calculating more generalized perturbation for the parameters of the learned recognizer.
    - During the test, if the target instances are allowed to be recognized in a batch way, a shared perturbation for the parameters will lead to more significant performance.

The correlations between the proposed methods are shown in Fig. 1.2. We first focused on the method with the settled metric (Euclidean distance) embedding method and developed a self-focus mechanism to prevent obtaining the over-fitted embedding function. Since this proposed mechanism can not be implied to the embedding methods with learned metrics (learned similarity). We further designed an adversarial framework for this kind of method to learn more generalized metrics for recognizing unseen classes. However, the designed framework can only consider a single instance at a time, which may lead to too extreme perturbations in the framework and thus affect the prediction of seen classes. Therefore, we improved the adversarial framework and developed it into the

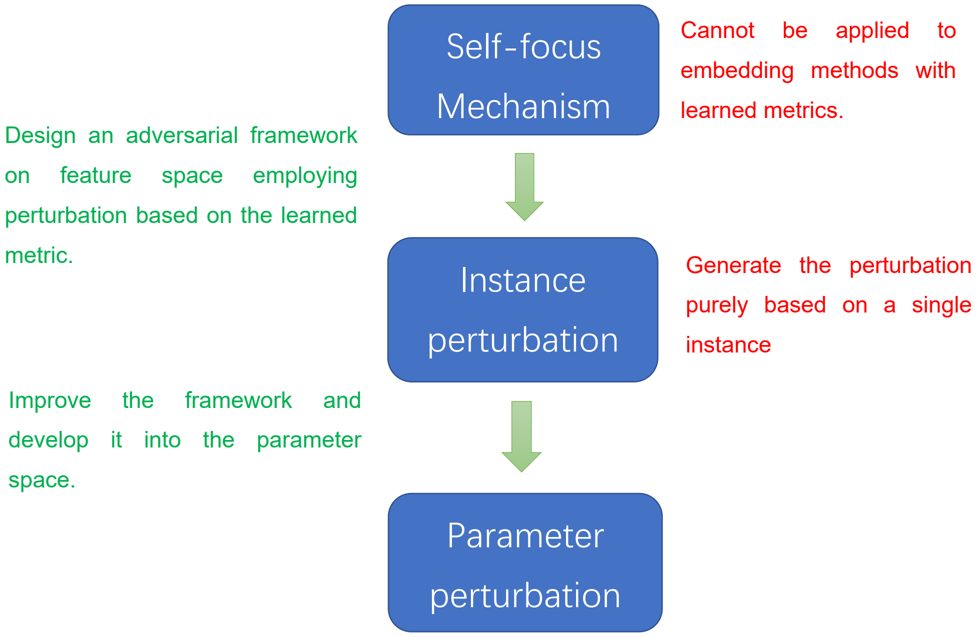

Self-focus Mechanism
Cannot be applied to embedding methods with learned metrics.
Design an adversarial framework on feature space employing perturbation based on the learned metric.
Instance perturbation
Generate the perturbation purely based on a single instance
Improve the framework and develop it into the parameter space.
Parameter perturbation

Fig. 1.2: The correlation between the proposed methods.

parameter space. Such design achieves obtaining perturbation based on multi-instance, thus avoiding the drawbacks associated with extreme perturbations.

1.2 Summary of Remaining Chapters

Chapter 2 Comprehensive Survey on Zero-Shot Image Classification In this chapter, we provide a comprehensive survey on zero-shot image classification. Backgrounds of the research field including the motivations and research target are introduced, optional auxiliary information such as semantic attributes and text are described with examples, the training and test scenarios and problem definitions are specifically explained, and representative methods for each family of methods are reviewed to provide a thorough understanding of the zero-shot learning. Furthermore, we summarize the reported performance of the representative methods with implementation details in terms of backbone modification, fine-tuning and additional knowledge. This chapter is based on a paper by Yang et al., which has been currently accepted by the journal Applied Computing and Intelligence.

Chapter 3 Efficient Self-Focus Mechanism for Coarse-Grained Generalized Zero-Shot Learning In this chapter, we introduce a self-focus mechanism based on a well-performed deep embedding method as the baseline. The baseline is designed to project semantic attributes into the settled visual feature space to avoid aggravating an introduced hubness problem in zero-shot learning. The specific designs for the baseline and the proposed self-focus module are presented. We indicate how the learned focus ratio improves the metric to employ the correlations between the location and the importance of each dimension, and also explain why such an improved metric is purely adopted during training. Furthermore, sufficient experiment results are demonstrated to verify that the proposed mechanism could effectively alleviate the class-level over-fitting problem. This chapter is based on the papers of Yang et al. [15, 16]

Chapter 4 Adversarial Relation Network for Generalized Zero-shot Learning In this chapter, we propose an adversarial framework based on an embedding baseline where similarities between semantic attributes and visual samples are learned by a DNN, namely relation network. The two main components in this framework, a robust training process and a sampler re-adjustment process, are introduced in detail. We proved how the training process with the l2 norm on inputs as a regularization term could improve the robustness of the model. We also explained the calculation process for the beneficial perturbation. The comprehensive performance evaluation and ablation study are demonstrated to verify the effectiveness of the entire framework and each component process, respectively. This chapter is based on a paper by Yang et al. [17]

Chapter 5 Instance-Specific Perturbation on Parameters for Relation Network based Generalized Zero-Shot Learning In this chapter, referring to the proposed adversarial framework on sample space, we develop an adversarial framework on the parameter space of the learn metrics for the relation network based embedding methods. We present the process of how the robustness of parameters is attained and how the instance-specific perturbations are calculated for both the single instance and group instances cases. The benefit of obtaining perturbations with group instances is explained, and the effectiveness of the proposed framework is evaluated on two baselines, thereby verifying its generalizability.

Chapter 6 Conclusion We will summarise this dissertation and conduct discussions on future works.

In order to make each of these chapters self-contained, some critical contents appearing in previous chapters may be briefly reiterated in several chapters, such as the model definitions or illustrative figures.

Chapter 2

Comprehensive Survey on Zero-Shot Image Classification

In this chapter, we present an overview of image classification in zero-shot learning including its relevant definitions, learning scenarios, and various methodologies. While we properly structure each part and summarize each family of methods with illustration, visualization, and tables, we put one main focus of this work on sorting out the implementation details, such as commonly used benchmarks, and diverse experiment settings so as to offer more practical guidance to researchers in the area. In the end, the comparison results of various representative methods are collected on a number of benchmarks, aiming to provide a fair and objective reference for evaluating different methods.

Compared to the recently-presented surveys [18, 19], our work shows three major differences. First, our work also introduces the most recently published important methods, as more seminar works and even breakthroughs emerged recently, thus reflecting a more timely and comprehensive review. Second, based on model components, training strategies, and learning objectives, we provide a more detailed hierarchical classification for zero-shot image classification methods. Third, we put one main focus of our survey on comparing different methods from the perspective of implementations, thus offering practical guidelines for applying zero-shot learning in real scenarios.

2.1 Backgrounds

In the field of computer vision, deep learning methods have made great achievements in both applied computing and machine intelligence. Remarkably, deep learning attains unprecedented success in image classification. Exploiting many powerful DNNs, machines can perform at a level close to or even beyond that of humans in many applications as long as sufficient labelled samples are provided [20–22]. However, the conventional DNN models rely on many important factors in order to achieve excellent performance.

Typically, DNNs require a huge number of labelled samples for training, whilst massive sample collection and labelling may unfortunately be difficult, time-consuming, or even impossible in many cases.

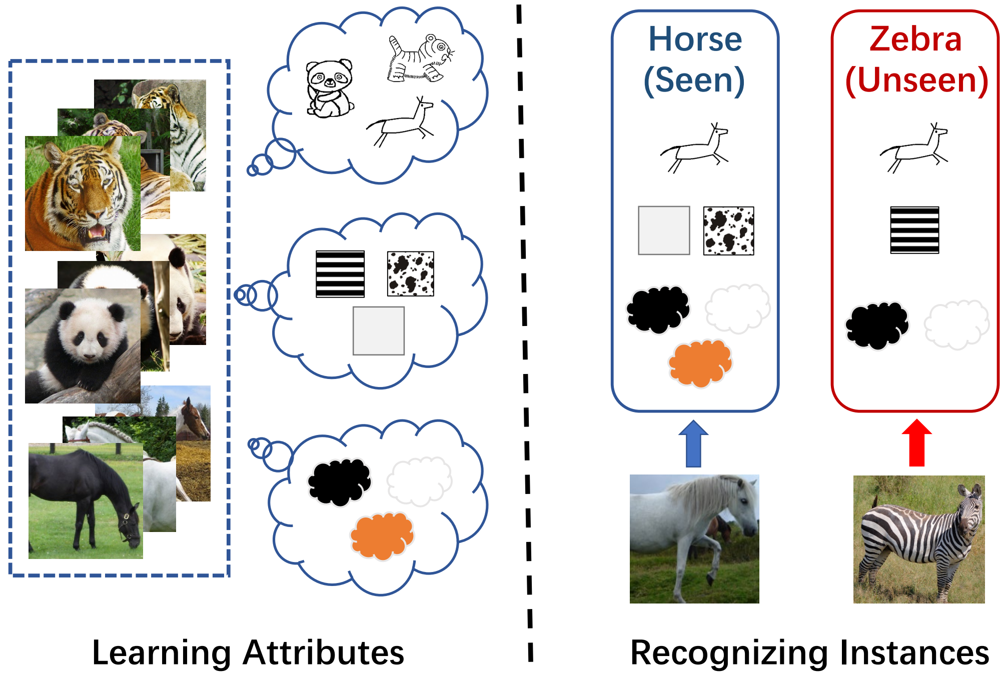


Fig. 2.1: Examples for Human learning processes.

In fact, not in line with DNNs' high demand for data, there are many scenarios which are commonly seen in practice:

- **Large target size**. Human beings could distinguish around 3,000 basic-level classes [23], and each basic class could be expanded as subordinate ones, such as dogs in different breed [24]. Such a huge number of categories makes it infeasible to construct a task where each category has a sufficient number of labelled samples.

- **Rare target classes**. Some tasks suffer from rare classes for which the corresponding samples are difficult to be obtained, such as fine-grained classification over flowers and birds [25, 26] or medical images corresponding to certain specific situation [27].

- **Growing target size**. The target set for some tasks changes rapidly, with candidate classes increasing over time, such as detection of new events in newly collected media data [28], recognizing the brand of a product [29] or learning some writing styles [30].

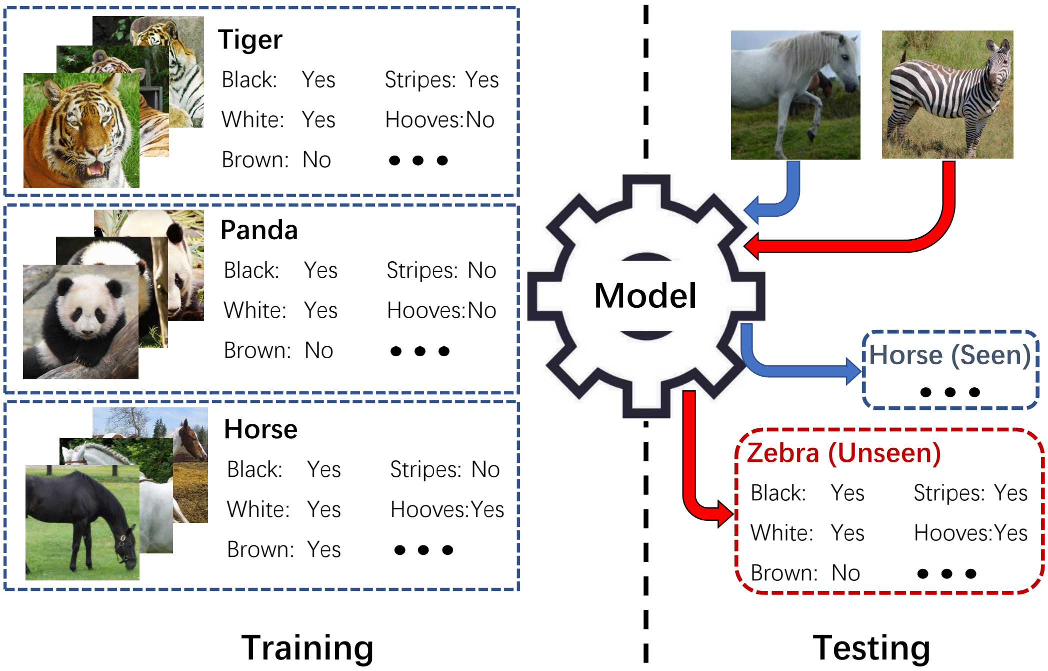


Fig. 2.2: Examples for zero-shot learning processes.

In those scenarios, re-training a DNN model over target classes appears not very feasible. Fine-tuning the trained model might be tractable only if some of the labelled target samples could be obtained. To overcome such restrictions, zero-shot learning, earlier called zero-data learning, is set up to simulate the learning capacity of human beings [31]. Fig. 2.1 demonstrates a schematic graph for the efficient human learning process. Assuming a child is equipped with knowledge including the shape of the horse, the concept of stripes, and colours of black and white, once being told that zebra looks like a horse covered in black and white stripes, the child has a good chance of recognizing a zebra even if seeing it for the first time [32]. Fig. 2.2 demonstrates a schematic graph for the zero-shot learning process that situations are also similar in zero-shot learning. Based on the auxiliary information used to describe each category and some corresponding samples, a model can be trained to construct the correlation between samples and the auxiliary information, thus enabling to extend the classification to unseen categories, based on their correlation as well as the auxiliary information.

2.2 Overview of Zero-Shot Learning

To describe the zero-shot classification task precisely, we will first review and explain some commonly used terms and notations in this section, then focus on introducing the

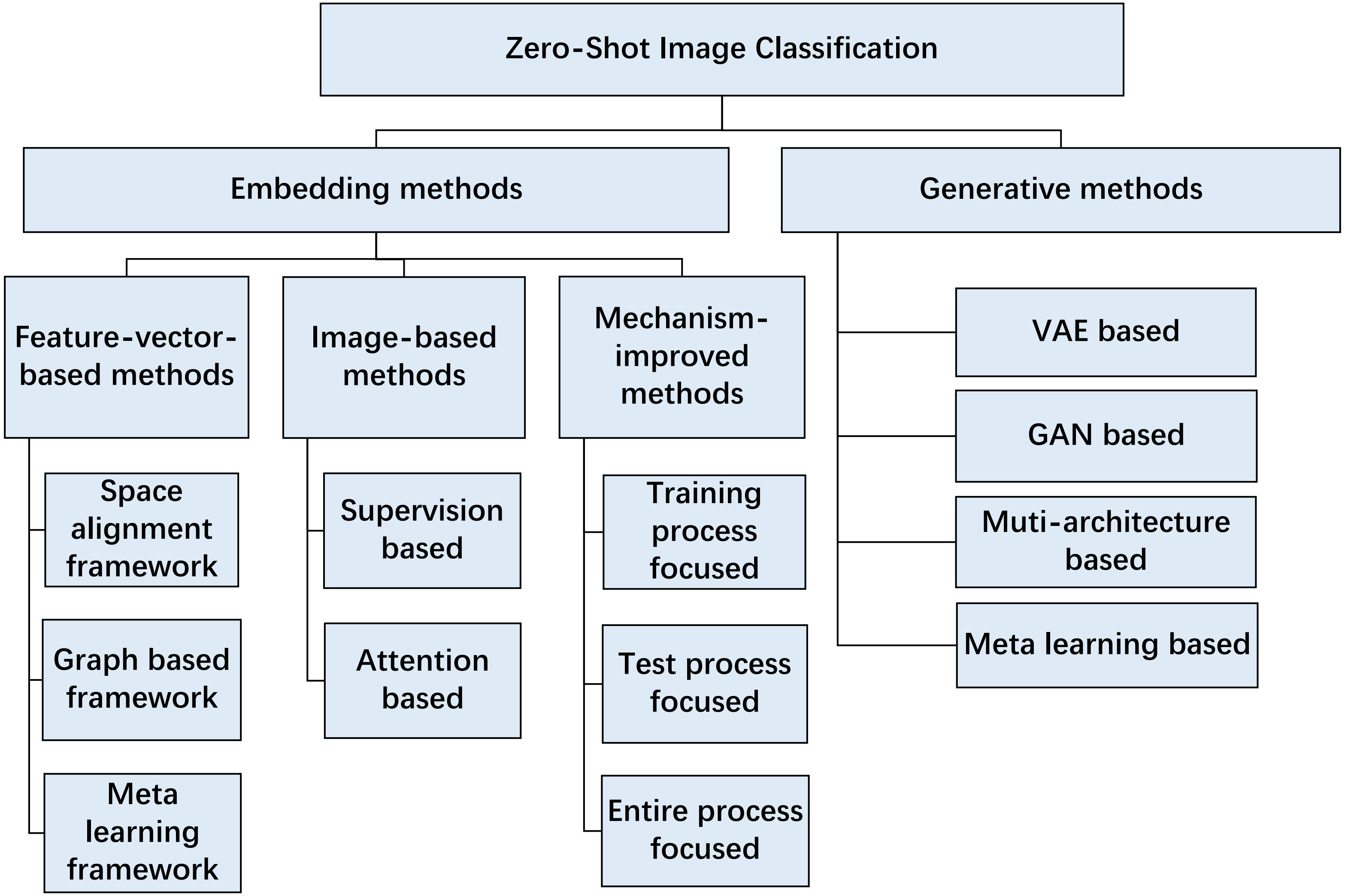


Fig. 2.3: The taxonomy structural diagram for Zero-Shot image classification methods.

zero-shot image classification methods which employ semantic descriptions as auxiliary information in the next two sections. Based on the design of the information extractor, we classify the current methods into two main categories: *embedding methods* and *generative methods*, and propose a taxonomy structure for these methods as shown in Fig. 2.3. For simplicity of expression, all subsequent references to zero-shot learning refer to the image classification task under this domain.

2.2.1 Auxiliary Information

In zero-shot learning, the target classes without corresponding training samples are named as unseen classes, whilst the classes with labelled samples during training are called seen classes. Due to the absence of training samples for unseen classes, the auxiliary information is essential for constructing the cognitive concepts of unseen categories. The space of such auxiliary information should contain enough information to distinguish all classes. In other words, for each class, corresponding auxiliary information should be unique and sufficiently representative to guarantee that an effective correlation between the auxiliary information and the samples can be learned for classification. Since zero-shot learning is inspired from the human efficient learning process, semantic information has become the commonly dominant auxiliary information [26, 33, 34]. Similar to the feature space for image processing, there is also a corresponding semantic space holding numeric values in zero-shot learning. To obtain such semantic space, two different kinds of semantic sources, attributes and textual descriptions, are mainly leveraged.

Attribute. Attribute is the earliest and most commonly used source of semantic space in zero-shot learning [19, 31, 35]. As a kind of human-annotated information, attribute contains precise classification knowledge though its collection might be time-consuming. Considering an attribute as a word or phrase introducing a property, one can build up a list of attributes. By combing these attributes, all the seen and unseen classes can be described. Moreover, these combined descriptions should be different for each class. Then the vectors, holding binary values 0 and 1 with sizes equal to the number of the attributes, form a semantic space where each value denotes whether the described class is equipped with the corresponding attributes or not. In other words, the attribute vectors for all the classes share the same size, and each dimension of the vector denotes a specific property in a settled order. For example, in animal recognition, one attribute could be *stripe*. Value 1 in the dimension of *stripe* of the attribute vector means that the described animal is with stripes [35]. Suppose there are only 3 attributes: *black*, *white*, and *stripes*, then the attribute vectors describing classes *panda*, *polar bear* and *zebra* should be something like [1, 1, 0], [0, 1, 0] and [1, 1, 1], respectively. However, since an attribute vector is designed to describe the entire class, it might be imprecise to use binary values only. The diversity of individuals within each class may lead to a mismatch between the sample and attributes. Taking the animal recognition again as an example, we can see horses might also be in pure black and pure white. If the attribute values of both *black* and *white* equal 1 for the class *horse*, then the black horse samples are contradictory to the attribute *white*, so are the white horses to the *black*. Therefore, instead of taking the binary value, it makes more sense to employ continuous values indicating the degree or confidence level for an attribute. It is shown in [36] that adopting the average value of the voting results or the proportion of the samples corresponding to an attribute leads to better classification performance. Additionally, the relative attribute measuring the degree of attribute among classes is also suggested [37].

Text. Instead of using human-annotated attributes, descriptions of a class such as the name or definition could also be considered as the source to construct a semantic space. However, it is not straightforward to transform the unstructured textual information into representative real values. When the class name is exploited as the semantic source without any external knowledge, the contained information might be far from enough for achieving a good classification on images. In this case, pre-trained word embedding models borrowed from natural language processing could embed the class names to some representative word vectors and form a meaningful semantic space. Specifically, the semantic similarity of two vocabularies can be approximately measured by the distance between the two corresponding embedded vectors, thus the similarity knowledge contained in the training text corpora (for constructing the word embedding models) could be adopted for classification. In the existing methods, Word2Vec [38–41] and GloVe [38, 41, 42] pre-trained on English language Wikipedia [43] are two commonly used embedding models for class name sources. Such semantic similarity measure space can also be constructed via the knowledge in terms of ontology. An example is to adopt the hierarchical embedding from a large-scale hierarchical database WordNet [38]. The keyword is another optional semantic source. The descriptions of classes are collected through databases or search engines to extract keywords. Consequently, the binary occurrence indicator [44] or frequencies [38] in Bag-of-Words, or transformed term frequency–inverse document frequency features [25, 26, 45] can construct such semantic vectors. The description in the form of paragraph could also be used as a semantic source. For example, visual descriptions in the form of ten single sentences are collected for images in [46]. After that, the text encoder model is utilized to return the required semantic vectors. This kind of semantic source contains more information as well as more noises.

Other auxiliary information. In addition to the semantic source, other types of supporting information also exist. That kind of information is often employed simultaneously with semantic information to assist the model in extracting more effective classification knowledge. For instance, hierarchical labels in taxonomy are introduced to provide additional supervision of classification [47, 48]; the human defined correlation between attributes [49] capturing the gaze point of each sample is adopted as the attention supervision to improve the attention module producing more representative feature maps [42]; Some of these information may not provide sufficient knowledge to accomplish the entire classification task. However, they can be regarded as the supplementary of semantic information which may better construct cognitive concepts of unknown categories.

2.2.2 Learning Scenarios

In conventional image classification tasks, due to the differences in the distribution of instances between the training and test sets, the trained model does not perform as well during the test as it does on the training set. This phenomenon is also present in zero-shot learning, and is even more severe owing to the disjoint property of seen and unseen classes. Such differences in the distribution between seen and unseen classes are called domain shift [50]. Moreover, the poor model performance is termed as class-level overfitting [51].

To address this challenge, by effectively employing classification knowledge from samples and auxiliary information, researchers have proposed various methods of introducing knowledge at different stages (including training and testing). As a result, the implementation scenarios become diverse. Both sample space and auxiliary information space can be defined in zero-shot learning, according to which we can divide the scenarios accordingly. In general, from the perspective of the training stage, the task can be divided into three scenarios, namely inductive, semantic transductive, and transductive, which are defined as follows:

- **Inductive zero-shot learning**. Only labelled training samples and auxiliary information of seen classes are available during training.

- **Semantic transductive zero-shot learning**. Labelled training samples and auxiliary information of all classes are available during training.

- **Transductive zero-shot learning**. Labelled training samples, unlabelled test samples, and auxiliary information of all classes are available during training.

From the definition, the inductive zero-shot learning represents the most severe learning scenario because both the target classes and instances are unknown. Models trained in this scenario are more likely to suffer from class-level over-fitting. In comparison, models trained in the rest two transductive scenarios share a clear learning objective since the classification knowledge is guided by the unseen information. However, these trained models will not generalize to new unseen classes as well as the models trained in the inductive scenario [19].

2.3 Overview of Zero-Shot Learning

When the zero-shot problem was first proposed in the early stage, researchers focused only on achieving good classification on unseen classes, which is known as conventional Zero-Shot Learning. Later, it was found that the classification of the unseen classes would suffer from a devastating blow once the seen categories were also included as candidates for classification. In other words, the early proposed models could not distinguish well between seen and unseen categories and thus failed to construct the cognition concepts of new classes. Consequently, a more challenging task called Generalized Zero-Shot Learning attracts much attention, which requires classifying both seen and unseen classes [52]. The original intention of zero-shot learning is to simulate the human process of constructing the cognition concept of classes from learned knowledge and supporting information in the absence of samples. Since the constructed cognitive concepts can be evaluated accurately only if the unseen and seen classes can also be correctly distinguished, the focus of current works has shifted to the generalized one. Fig. 2.4 shows the schematic of different scenarios in training and test, where the combination of the different scenarios forms six common settings.

Problem definitions In zero-shot learning, each sample is originally designed as an image containing certain specific objects in a tensor form holding value for each pixel. To ensure more convenient implementation, the visual features extracted by a pre-trained DNN are commonly regarded as the samples instead of using the image. For a rigorous presentation, here we take the entire image as the input sample in our article. Assuming there are totally N samples from K classes, we denote $\boldsymbol{X} = \boldsymbol{X}^S \cup \boldsymbol{X}^U$ as the set of all the image samples from both seen and unseen classes, and $\mathbf{F}(\cdot)$ as a feature extractor for obtaining the feature $\mathbf{F}(x_i)$ of the image x_i. Similarly the corresponding label set could be denoted as $\boldsymbol{Y} = \boldsymbol{Y}^S \cup \boldsymbol{Y}^U$, and $y_i = k$ indicates that sample x_i belongs to the kth-class. The set of the auxiliary information is denoted as $\boldsymbol{A} = \boldsymbol{A}^S \cup \boldsymbol{A}^U$ which contains K vectors where each vector a_k stands for the auxiliary information of the kth-class. Here let K^S and K^U indicate the number of seen and unseen classes, respectively, and the first K^S classes represented in $\boldsymbol{A}$ are assumed as the seen ones for convenience. Note that the seen

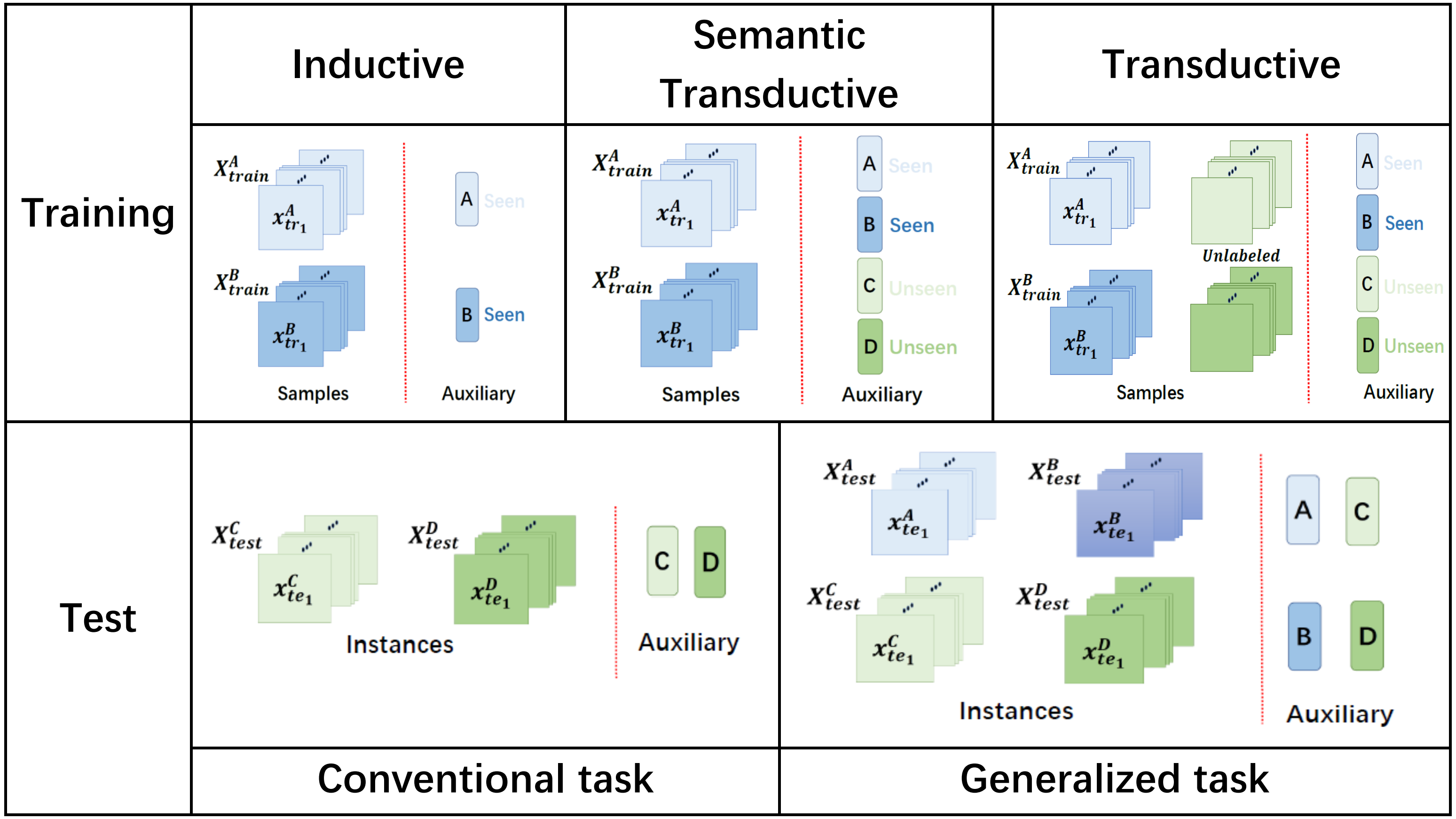


Fig. 2.4: Schematic diagrams of utilizing data for different scenarios in training and test.

and unseen classes are disjoint, which means $\boldsymbol{X}^S \cap \boldsymbol{X}^U = \boldsymbol{Y}^S \cap \boldsymbol{Y}^U = \boldsymbol{A}^S \cap \boldsymbol{A}^U = \emptyset$. As partial of seen class samples are adopted as test instances which should not participate in the training process, the seen sets of the samples and labels are further consistently divided into training and test sets as $\boldsymbol{X}^S = \boldsymbol{X}^S_{tr} \cup \boldsymbol{X}^S_{te}$ and $\boldsymbol{Y}^S = \boldsymbol{Y}^S_{tr} \cup \boldsymbol{Y}^S_{te}$. Specifically, both of the train and test seen sets should cover all the K^S seen classes. Since there are three scenarios for the training process, the training set $\boldsymbol{D}_{tr} = \{\boldsymbol{X}_{tr}, \boldsymbol{Y}_{tr}, \boldsymbol{A}_{tr}\}$ can be respectively defined for the inductive, semantic transductive, and transductive scenarios in the three forms as $\boldsymbol{D}^I_{tr} = \{\boldsymbol{X}^S_{tr}, \boldsymbol{Y}^S_{tr}, \boldsymbol{A}^S\}$, $\boldsymbol{D}^{ST}_{tr} = \{\boldsymbol{X}^S_{tr}, \boldsymbol{Y}^S_{tr}, \boldsymbol{A}\}$ and $\boldsymbol{D}^T_{tr} = \{\boldsymbol{X}^S_{tr} \cup \boldsymbol{X}^U, \boldsymbol{Y}^S_{tr}, \boldsymbol{A}\}$. For the test set $\boldsymbol{D}_{te} = \{\boldsymbol{X}_{te}, \boldsymbol{Y}_{te}, \boldsymbol{A}_{te}\}$, it can also be defined in two forms as $\boldsymbol{D}^C_{te} = \{\boldsymbol{X}^U, \boldsymbol{Y}^U, \boldsymbol{A}^U\}$ for conventional task and $\boldsymbol{D}^G_{te} = \{\boldsymbol{X}^U \cup \mathbf{X}^S_{te}, \boldsymbol{Y}^U \cup \boldsymbol{Y}^S_{te}, \boldsymbol{A}\}$ for generalized task, respectively. With these definitions, the target of zero-shot learning can be represented to train an information extractor $\mathbf{M}$ (containing the feature extractor $\mathbf{F}(\cdot)$) with a settled or a learnable classifier $\mathbf{C}$ on the training set $\boldsymbol{D}_{tr}$ to achieve classification on $\boldsymbol{X}_{te}$.

2.4 Embedding Methods

In the embedding methods, the information extractor $\mathbf{M} = \{\theta(\cdot),\ \phi(\cdot)\}$ is designed as a union of embedding functions $\theta(\cdot)$ and $\phi(\cdot)$. The aim of these extractors is to find the proper embedding spaces for both visual samples and auxiliary information so that the trainable or settled classifier $\mathbf{C}$ can achieve class recognition on the target space. From the perspective of the learning objective, we further classify the existing embedding methods as: (1) *feature-vector-based*, (2) *image-based*, and (3) *mechanism-improved* methods.

2.4.1 Feature-Vector-Based Methods

Considering the limitation of the sample size and the latent distribution differences between the samples of the unseen and seen classes, the most easily associated and appropriate visual feature space is the learned space in large-scale conventional image classification tasks. Fair data splits and extracted features for several benchmarks are discussed and evaluated in [53]. The feature vector space learned by the deep residual network called ResNet101 [54] over a benchmark dataset ImageNet [55] is commonly selected in the implementations. Based on the fixed feature extractor $\mathbf{F} = \mathbf{F}_f$, feature vectors $\mathbf{F}_f(\boldsymbol{X})$ are regarded as the visual samples and the insight of the feature-vector-based methods is to design embedding functions or classifiers trying to improve the performance where the classifier $\mathbf{C}\left(x_i, \boldsymbol{A}, \mathbf{M}\right)$ is commonly constructed as a function taking the embedded features and attributes to return the predicted confidence scores of all the classes represented in $\boldsymbol{A}$. We will review this family of methods according to their mainly relied frameworks.

Space alignment framework. These encoding based methods often have a specific embedding target space, which can be a commonly-used visual feature space, an manually defined semantic description space, or an unknown hidden space for detecting certain correlations. This idea is the first as well as one most common solution to zero-shot learning.

The classifier can be designed based on a fixed distance metric $\mathbf{d}(\cdot,\cdot)$ such as Euclidean distance or Cosine distance. Thereby, the predicted label for each visual feature $\mathbf{F}_f(x_i)$ is obtained as

$$\hat{y_i} = \arg\min_k(\mathbf{d}\left(\theta\left(\mathbf{F}_f\left(x_i\right)\right), \phi\left(a_k\right)\right), \quad s.t. \quad a_k \in \boldsymbol{A}_{te}. \tag{2.1}$$

In the following, we briefly review some representative work in the space alignment framework. In [13], semantic-to-visual mapping is learned to align semantic and visual features from the same class. Specifically, this method utilizes a multi-layer neural network as the embedding function implying that the visual feature space is more appropriate as a target space to avoid aggravating the hubness problem. More studies adopt the reconstruction or bi-direction mapping (a relaxed form of reconstruction) process to align the information from different spaces. Linear embedding functions are applied for both visual-to-semantic and semantic-to-visual projections in [56], and a rank minimization technique is additionally adopted for optimizing the linear transformation matrices. In [57], the encoding processes of the reconstruction are designed in both visual and semantic spaces, and achieve the joint embedding by minimizing the maximum mean discrepancy in the hidden layer. Then as a more strict case, the embeddings for the visual feature and semantic attributes from the same class are enforced to be equal in [58], and a two-alternate-steps algorithm is proposed in [59] to solve transformation matrices in the joint embedding with reconstruction supervision in two alternate steps. Similar classes for each class are selected via a threshold among cosine similarity in [60], then a semantic-to-visual-to-semantic reconstruction process is proposed, where the inter-class distances are pushed and the intra-class distances are reduced on the visual space. A projecting codebook is learned in [61] with an additional center loss in [62] and a reconstruction loss in [56] to embedded visual features and semantic attributes to a hidden orthogonal semantic space. The label space is selected as the embedding target space in [63], where the embedding of the unseen semantic attributes to the label space can be achieved by learning the projecting function from both the semantic and visual spaces to the label space. Such embedding is equivalent to linearly representing the labels of unseen classes by those of seen classes, thus improving the generalization of the model in the label space.

The classifier can also be designed learnable such as a bilinear function $\boldsymbol{W}$, which predicts the confidence scores as

$$\mathbf{C}\left(x_i, \boldsymbol{A}, \mathbf{M}, \boldsymbol{W}\right) = \theta\left(\mathbf{F}_f\left(x_i\right)\right)^T \boldsymbol{W}\, \phi\left(A\right). \tag{2.2}$$

The semantic attributes of both the seen and unseen classes are purely represented by those of seen in [64] to train the bilinear function which thus associates unseen classes with seen classes. Norms of the embedded semantic attributes and embedded visual feature is constrained in [65] for fair comparisons over classes and bounding the variation in semantic space, respectively. In [51], the bilinear function is decomposed into two transformation matrices, and it is proved that minimizing the mean squared error between similarity matrices and the predicted scores for all samples is equivalent to restricting those transformation matrices to be orthogonal. A pairwise ranking loss function similar to the one in [66] is proposed in [67] as

$$\sum_{j}^{K^S} \left[margin\ I(j = y_i) + \mathbf{C}\left(x_i, a_j, \mathbf{M}, \boldsymbol{W}\right) - \mathbf{C}\left(x_i, a_{y_i}, \mathbf{M}, \boldsymbol{W}\right)\right]_+ . \tag{2.3}$$

Instead of the sum of all these pairwise terms, the ranking loss is modified by focusing on the pair. This leads to the maximum value in [68] and results in a weighted approximate one in [38] inspired by the unregularized ranking support vector machine [69]. It can also be redesigned with a triplet mining strategy to construct the triplet loss with the most negative samples and the most negative attributes as proposed in [70].

Moreover, the classifier can be defined in other forms. The instances from each class are assumed to follow an exponential family distribution in [71] where the parameters are learned from the semantic attributes. The method in [41] develops the ranking loss into a non-linear classifier case by learning multi-bilinear classifiers where each time this model chooses the one with the highest confidence score to be optimized. In [72] the attributes of unseen classes are utilized to reconstruct those of seen classes by the sparse coding approach. The solved coefficients are regarded as the similarity between classes. Then a neural network is designed to learn the similarity between the embedded attributes and visual features under the supervision of the labels and the similarities.

Graph based framework. A graph containing correlations between classes can be additionally constructed to enhance the generalization of the trained model. In [73], two relation graphs of the features in the hidden space are constructed based on the k-nearest neighbors among samples and the class labels which contribute to reducing distances between highly relevant features. This design is improved in [74] where two separated latent spaces are learned for embedding the visual samples and semantic attributes, and the k-nearest neighbor is replaced by the Cosine similarity to imply the relations between samples. Based on the two embedding spaces and the weighted sum of relations between samples and class labels an asymmetric graph structure with orthogonal projection is introduced to improve the learned latent space. By fixing the number of super-classes in different class layers, clusters obtained through the clustering algorithm for the attributes are taken to represent the super-class in [75], thereby a hierarchical graph over classes can be constructed to overcome the domain gap between seen and unseen classes. In [76], the relations between the classes are captured by augmenting the original label matrix in a dependency propagation process with the support of the low-rank constraint.

The graphic convolutional neural network (GCN) is a neural network that directly approximates localized spectral filters on graphs to learn hidden layer representations more relevant to the target task [77]. GCN is applied on the word embeddings of all the classes in [78] to learn the classifier parameters for each class. Then a dense graph propagation module is proposed in [79] where the connections from nodes to their ancestors and descendants are considered. In addition to the graph of word embeddings, in [80], the graph constructed through the k-nearest neighbor in the attribute space is also employed to learn the classifier parameters. The outputs of the GCN based on two graphs are weighted summed to learn the final parameters.

Meta learning framework. Meta learning process proposed in the few-shot learning aims to train models with high knowledge transfer ability [81]. In zero-shot learning, models trained on seen class data tend to overfit and perform poor on unseen classes. Therefore, the methods with similar meta learning strategies are developed to train more generalized models.

Relation network (RN) [14] is designed to learn a similarity measure based on the neural network architecture. The visual feature and embedded semantic attributes will be concatenated and used as the input to the measure model to return the similarity. The whole model is trained under a meta learning process where each time the loss function is designed based on a meta learning task sampled from the training set. Specifically, each time a small group of the samples are selected to construct the meta classification task where the number of the included classes is not settled. By training over several meta tasks, the trained model would be more adaptive for different tasks. Therefore, the model would be more generalized.

As an improvement of RN, CRnet [82] follows the same training process with the meta tasks. Additionally, an unsupervised K-means clustering algorithm is implemented to find the similar class groups and the corresponding group centers. Instead of training one embedding function for the semantic attributes, multi-attributes embedding functions are trained based on the group centers where the inputs are the differences between these centers and the semantic attributes. Then the sum of these embedded attributes is utilized for learning the similarity in the same way as RN.

A similar process is adopted in a correction network [83]. Based on the sampled meta tasks, an additional correction module is trained to modify the predicted value of the original model to become more precise. Then the learned correction module would be generalized since it is adapted to different meta tasks. As such, the correction will contribute to better performance.

2.4.2 Image-Based Methods

In the image-based methods, it is the original images $\boldsymbol{X}$ instead of the extracted feature vectors $\mathbf{F}_f(\boldsymbol{X})$ that are regarded as samples. Moreover, the well-designed backbone architecture with pre-trained parameters from the image classification task is partially or entirely borrowed as a learnable one $\mathbf{F} = \mathbf{F}_l$. The insight underlying these methods is to optimize the feature extractor $\mathbf{F}_l$ simultaneously with the specific designed embedding function and classifier. Sometimes an additional module accompanying the backbone is designed to obtain a more adaptable feature space, thus improving the performance.

Supervision based methods. By providing additional constraints or regularizations in the loss function for training, the feature extractor can be pushed to capture more relevant information, which results in a more representative feature space. Rather than training an embedding model with a bilinear classifier purely on the information from seen classes, unlabelled data are also employed in quasi-fully supervised learning [84]. Without supervised information, the predicted scores of the unseen classes for those unlabelled data are constrained to be large by constructing the sum of negative log values of them as a regularization term during optimization. Then training the whole model under this quasi-fully supervised setting with the designed loss will also improve the features extracted by the backbone. This can alleviate the bias towards seen classes.

A discriminative feature learning process is introduced in [85]. A zoomed coordinate is learned based on the feature maps to reconstruct a zoomed image sample with the same size as the original one, where visual features are extracted from both of the zoomed and original image samples. Since the semantic attributes are not discriminative enough, only a partial list of learned embedded features is adopted for learning the bilinear classifier with the attributes. Additionally, a triplet loss based on the squared Euclidean distance is constructed among the rest of the embedded features to improve the learned feature space.

Domain-aware visual bias eliminating [86] adopts a margin second-order embedding based on bilinear pooling [87] and a softmax loss function with a kind of temperature during training. As a result, the learned feature space constrained to be more discriminative leads to a low entropy for the instances from seen class. Then the instances from unseen class during the test would be distinguished with a relatively high entropy.

Attention based methods. As the attention mechanism has achieved significant performance in the image classification tasks [88], several attention relevant modules are also designed in zero-shot learning for capturing more representative features corresponding to the semantic information. In most of these methods, the attention module is utilized to obtain local features corresponding to certain specific semantic property. To produce more adequate supervision on the attention based feature space, a second-order operation [87] is applied on the learned features and semantics [89]. In the region graph embedding network [90], a transformation matrix is solved to represent the similarity between the attributes of the seen and the unseen classes. According to these similarities, a cross-entropy loss is then designed to ensure that the classifier also outputs a higher score for similar unseen classes when classifying samples from seen classes. As a result, the feature extractor is pushed to learn the feature space capturing more correlation information between seen and unseen classes. In [91], a triplet loss is designed to push the inter-class distances and reduce intra-class distances between features corresponding to both local and entire images. This model thus improves the learned feature space more conducive to the classification task.

Instead of purely training the attention module through the loss function defined on the feature space, additional explicit human annotated labels for attention can also be provided to supply the training. For example, in [42], captured gaze points are employed to generate the ground truth of the attention maps for constructing the binary cross-entropy loss across all the pixels. In addition to capturing local features, the attention learned from several feature maps is combined to guide the learning of the bilinear classifier [92].

2.4.3 Mechanism-Improved Methods

The insight of the mechanism-improved methods is to propose a generalized mechanism without changing or slightly changing the structure of the original method. The proposed mechanism can be an improvement of the training process, an optimization of a specific loss function, or a redesign prediction process. Commonly, this family of methods are designed for those zero-shot models sharing certain commonalities.

Training process focused. A theoretical explanation to normalization on attributes is presented in [93]. Then a more efficient normalization scheme is proposed standardizing the embedded attributes to alleviate the irregular loss surface.

During the feature extracting process, a fine-tuned backbone is proposed in the attribute prototype network (APN) [94]. In this work, assume the size of attributes is D_a. The prototype for each attribute $P = \{p_{d_a} \in \mathbb{R}^C\}_{d_a=1}^{D_a}$ is learned to generate similarity map $M^{d_a} = \{m_{i,j}^{d_a}\}^{h \times w}$ with height h and width w through multiplication of these prototypes and the corresponding feature maps. During the fine-tuning, the commonly used linear embedding classification loss is optimized with several regularization terms. An attribute decorrelation term is defined as the sum of l_2-norm of each dimension of the prototypes in the same disjoint attribute groups. This thus helps decorrelate unrelated attributes via enforcing prototypes in the same group sharing the value. Another similarity map compactness term can enforce the similarity maps concentrating on the peak region [95], which is given as

$$\mathcal{L}_{CPT} = \sum_{d_a=1}^{D_a} \sum_{i=1}^{h} \sum_{j=1}^{w} m_{i,j}^{d_a} [(i - \tilde{i})^2 + (j - \tilde{j})^2], \tag{2.4}$$

where $(\tilde{i}, \tilde{j})$ is the coordinate of the maximum value in M^{d_a}. This element-wise multiplication between the similarity map and the distance between coordinates constrains the similarity map to focus on a small number of local features. Thereby, each similarity map M^{d_a} can be regarded as the attention map corresponding to d_a-th attribute. The comparison result in this work shows that the fine-tuned backbone in APN outperforms the ones in some other methods [96, 97], even when fine-tuning is also implemented. In this sense, it can be regarded as a general improved one for feature extracting.

Isometric propagation network (IPN) [98] is proposed to guarantee the relation between classes in a propagation process based on a specific similarity measure. By defining the average of samples from the same class as the initialized visual class prototype, in the propagation, each time the prototype is re-represented by the weighted sum of the prototypes of similar classes. The similar classes are detected through a threshold and a similarity measure which is the softmax with temperature on the cosine similarity for each prototype. The similarity is also utilized as the weight for the re-representation. Such a propagation process can also be implemented on the semantic prototypes learned based on the trained semantic embedding module in other methods such as that used in [82]. During the test, the unseen prototypes could be obtained using the weighted sum of the propagated prototypes of seen classes according to the similarity measure, which contributes to significant performance improvement with the commonly used linear classification model.

The image is divided into different regions for extracting more precise features with the attention module in [99–101]. Moreover, an additional seen-unseen trade-off loss can be adopted to balance the predicted scores for seen and unseen classes. For example, a self-calibration loss term as a biased cross-entropy loss for the predicted unseen scores of samples from seen classes is designed in [99], and a soft cross-entropy loss based on the similarity between seen and unseen classes is utilized in [101]. Training the models with these additional constraints increases the prediction scores for unseen classes, thereby promoting the sensitivity of unseen class recognition.

A meta learning process with constructed meta training tasks is adopted in [4, 81] for few-shot learning. Instead of employing a loss function associated with the original classification task on the whole training set, several semi-tasks of the original task, namely meta tasks, are constructed with the meta training data sampled from the original training set. Adopting this meta learning process in zero-shot learning improves the generalization and restrains over-fitting [14, 82, 98]. Fig. 2.5 demonstrates an example of the meta zero-shot task in [14].

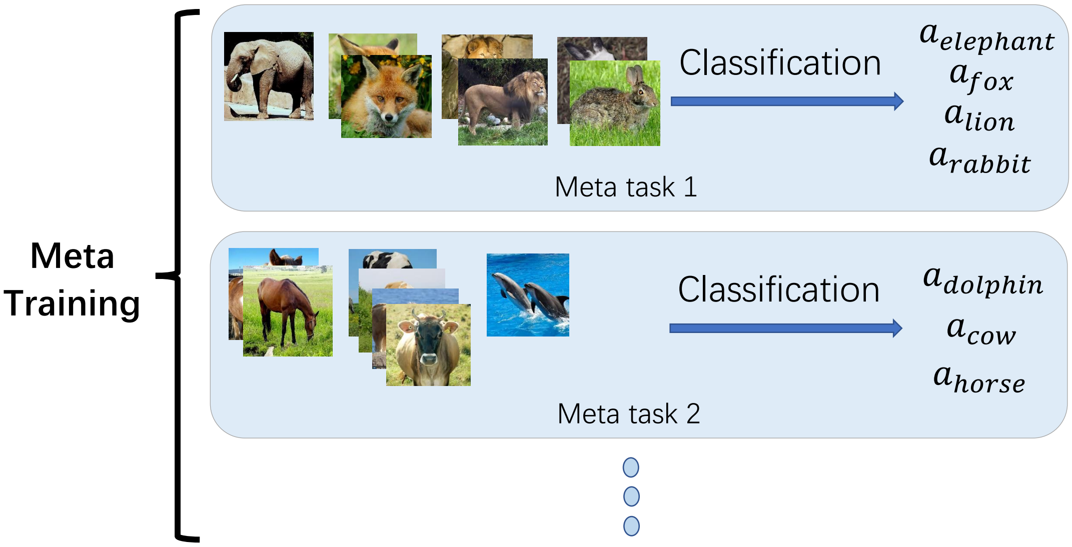


Fig. 2.5: One illustrative example of meta tasks in meta learning process adopted in training relation network.

Test process focused. Since most of the methods suffer the class-lever over-fitting in generalized zero-shot tasks, a mechanism named Calibrated stacking is proposed in [52] to adjust the predicted confidence score for each class. With a trained classifier **C** and corresponding information extractor **M**, the predicted confidence score in the regular test process can be obtained as $\mathbf{C}(x_i, \boldsymbol{A}_{te}, \mathbf{M})$. Then the prediction based on the calibrated stacking is defined as

$$\hat{y}_i = \arg\max_k \mathbf{C}(x_i, \boldsymbol{A}_{te}, \mathbf{M}) - \gamma I(k \leq K^S), \quad s.t. \quad a_k \in \boldsymbol{A}_{te}, \tag{2.5}$$

where $I()$ is the indicator function judging whether the k-th class belongs to a seen class and γ is the hyper-parameter controlling the scale of the adjustment. This calibrated stacking mechanism is simply subtracting a certain value for all the predicted seen confidence

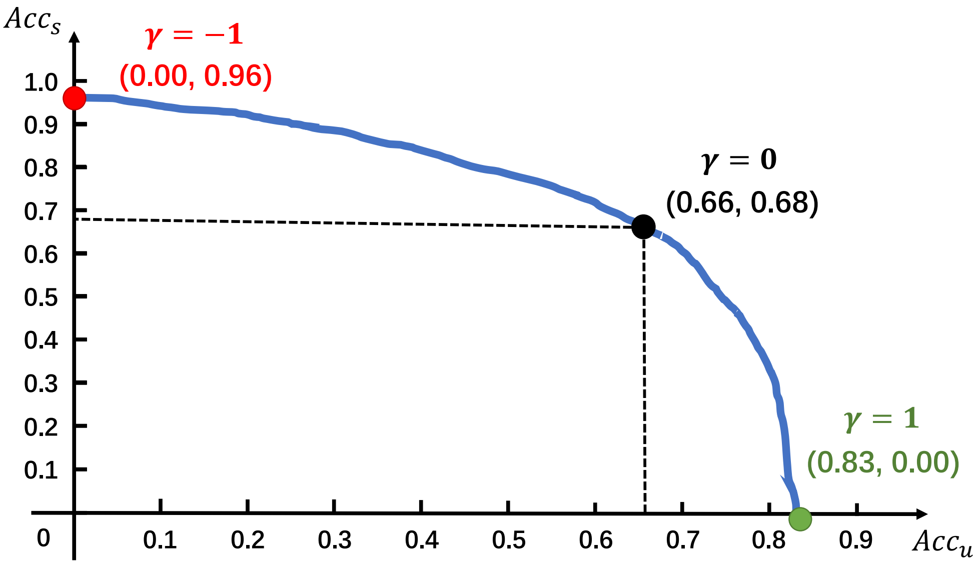


Fig. 2.6: Schematic of the seen-unseen accuracy curve. The black point with $\gamma = 0$ denotes the original performance of the model, the red point with $\gamma = -1$ and the green point with $\gamma = 1$ represent the adjusted results where the predicted scores are fully biased towards the seen and unseen classes, respectively.

scores. Specifically, assume all the confidence scores are scaled in the range (0,1). Setting $\gamma = 1$ will lead that all the predicted labels belong to unseen classes, and conversely $\gamma = -1$ will cause all the predicted labels as seen classes. In other words, setting $\gamma = -1$ and 1 lead to zero accuracies for the unseen classes and seen classes, respectively. By adjusting γ from -1 to 1 with a tiny step size, one can obtain the adjusted accuracies for both the seen and unseen classes. Then a seen versus unseen accuracy curve can be plotted. In this case, the area under seen-unseen accuracy curve (AUSUC) is proposed as one optimal criterion measuring the overall performance of the models in generalized zero-shot learning tasks. A schematic is shown in Fig. 2.6.

Entire process focused. In [102], a self-learning process is proposed where each time hard unseen classes are selected based on the frequencies of the prediction during the test. Then an expanded training set with additional sampled instances from those hard unseen classes is constructed to re-train the model. The modified training set could enhance the sensitivity of model for those hard classes thus can boost the performance of the model under the transductive scenario.

2.5 Generative Methods

The core component of generative methods is the generator that takes semantic information as input and outputs corresponding pseudo samples. Such a generator can be constructed based on variational autoencoder (VAE) [11] or generative adversarial network (GAN) [12] architecture. It can be also trained with the labelled samples with corresponded semantics. Then, by employing the unseen semantics, pseudo samples of unseen classes could be generated where the zero-shot learning task is converted to common classification. In this case, the information extractor **M** denotes a training process and the output is a trained generator **G** which takes $\boldsymbol{A}$ (sometimes combined with $\boldsymbol{X}_{tr}$) as inputs and outputs synthesized samples for corresponding classes. With the synthesized samples of unseen classes to support the training, the classifier can be designed as a common image classifier $C(\cdot)$ which takes samples as input and outputs the confidence score for each class. Here we will review those representative generative methods in different frameworks.

2.5.1 VAE Based Methods

Variational autoencoder is designed to derive a recognition model in the form $q_\phi(z|x)$ to approximate the real intractable posterior $p_{theta}(z|x)$ with the objective function:

$$\mathcal{L}(\theta, \phi, x_i) = -D_{KL}\left(q_\phi\left(z|x_i\right)||p_\theta\left(z\right)\right) + \mathbb{E}_{q_\phi(z|x_i)}\left[\log p_\theta\left(x_i|z\right)\right], \tag{2.6}$$

where D_{KL} denotes the Kullback-Leibler distance, $q_\phi(z|x)$ is regarded as a probabilistic encoder, and $p_\theta(x|z)$ is regarded as a probabilistic decoder. As the most straightforward form of VAE, conditional VAE [103] is applied to zero-shot learning in [104] as shown in Fig. 2.7, where the sample is concatenated with the corresponding attributes to learn the distribution parameters; the sampled random variables based on the learned parameters are again concatenated with the corresponding attributes to reconstruct the sample. The objective function can be simply redesigned as

$$\mathcal{L}(\theta, \phi, x_i, a_{y_i}) = -D_{KL}\left(q_\phi\left(z|x_i, a_{y_i}\right)||p_\theta\left(z|a_{y_i}\right)\right) + \mathbb{E}_{q_\phi(z|x_i)}\left[\log p_\theta\left(x_i|z, a_{y_i}\right)\right]. \tag{2.7}$$

In [105], Kullback-Leibler distance relevant to the synthesized samples and regression error of the semantic attributes from the corresponding synthesized samples are proposed as two additional regularization terms. A dual VAE architecture is designed in [8] where two VAE frameworks are trained respectively on the visual features and semantic attributes. The correlation between these two frameworks is constructed via minimizing the cross reconstruction errors and the Wasserstein distances between the latent Gaussian

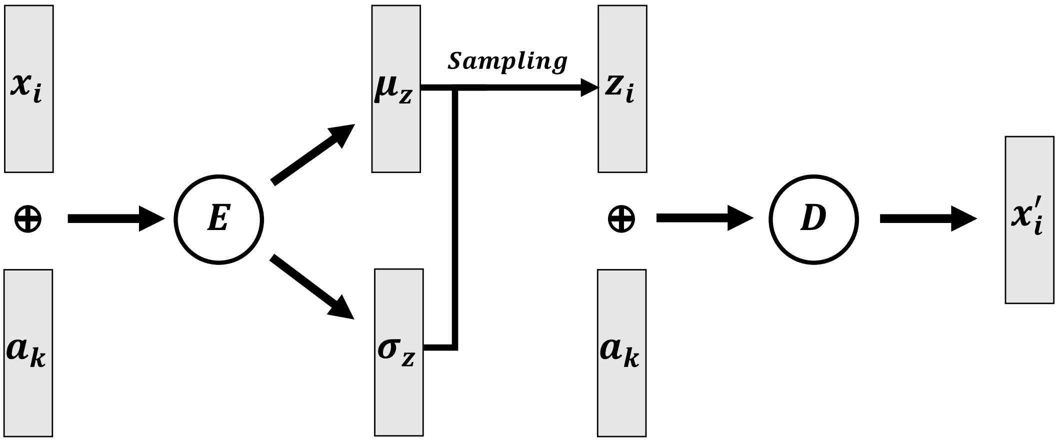


Fig. 2.7: Schematic diagram of conditional VAE, where $\oplus$ denotes concatenation, E denotes the encoder, and D denotes the decoder.

distribution for those sample-attributes pairs coming from the same class. The dual VAE is improved in [9], where a deep embedding network achieving the regression task from the semantic attribute to visual features is additionally designed. Then the hidden layer of this network is utilized as the input of the semantic VAE framework. The designed regression forces the hidden layer to become representative for both visual features and semantic attributes, thus benefiting the entire VAE framework. A disentangled dual VAE is designed in [106]. Different from the original dual VAE, each VAE framework learns two distributions, thereby sampling two random variables z_m^p and z_m^t. Notice that m denotes the modality which could be s and v representing semantic space and visual space, respectively. For a group of pairs of training data, $\{z_{m,i}^p\}$ is shuffled as $\{\tilde{z}_{m,i}^p\}$ and then added up with $\{z_{m,i}^t\}$. Optimizing the model with this additional classification loss disentangles category-distilling factors and category-dispersing factors from both of the visual and semantic features. The multimodal VAE proposed in [107] builds one VAE framework for the concatenation of the visual feature and the embedded semantic attributes from the same class to capture the correlations between modalities. In identifiable VAE designed in [108], three VAE frameworks sharing the decoder for sample reconstruction are built taking the sample, the attribute, and both of them as inputs, respectively. With an additional regularization term [109] encouraging disentanglement during inference, the learned latent space captures more significant information for generating discriminative samples.

2.5.2 GAN Based Methods

In generative adversarial networks, a generator G and a discriminator D are designed to be trained against each other iteratively with the loss function:

$$\min_{G}\max_{D}\mathcal{L}(D,G)=\mathbb{E}_{x\sim X_{tr}}[\log D(x)]+\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \tag{2.8}$$

Here, $p_z(z)$ denotes a prior on input noise variables z, the discriminator is trained to distinguish the generated pseudo samples from the samples in the original dataset and the target of generator is to synthesize pseudo samples as similar as the real samples so that the learned discriminator cannot recognize them. Then following the WGAN proposed in [110] where Wasserstein distance is leveraged, the loss of the conditional WGAN in zero-shot learning can be developed as

$$\begin{aligned}\min_{G}\max_{D}\mathcal{L}_{f-WGAN}\left(D,G\right)=&\mathbb{E}_{x\sim X_{tr}}\left[\log D\left(x,a_y\right)\right]-\mathbb{E}_{z\sim p_z(z)}\left[D\left(G\left(z,a_y\right),a_y\right)\right]\\&-\lambda\mathbb{E}_{z\sim p_z(z)}\left[\left(\left|\left|\nabla_{G(z,a_y)}D\left(G\left(z,a_y\right),a_y\right)\right|\right|^2-1\right)^2\right].\end{aligned} \tag{2.9}$$

In [6], a classifier for seen classes is pre-trained on the training set, then adopted to supply a classification supervision for the samples generated from a WGAN framework. Guided by this additional supervision, the generator will learn to synthesize more discriminate samples which benefits the training of the final classifier. Inspired by the prototypical networks in few-shot learning [5], multiple prototypes of each seen class are calculated in [111]. Samples of each class are grouped into several clusters, then the average of samples in each group is regarded as one prototype for the corresponding class. Similarly, the prototypes of the synthesized samples could also be obtained based on the clusters. By minimizing the distances from the synthesized samples to their closest corresponding prototypes and distances from the synthesized prototypes to their closest real prototypes, the synthesized samples are constrained to be highly related to the attributes and real samples. Instead of adopting the classification supervision, a gradient guidance from a pre-trained classifier is proposed in [7]. In this model, classifier parameters at different spots during training are employed for calculating the optimization gradients based on the real sample and synthesized sample, respectively. Expectations of the Cosine distances between the gradients are calculated from the real and synthesized samples, which are then utilized as an additional loss term to promote synthesizing samples as representative as real ones. In [10], conditional GAN is adopted with the designed instance-level and class-level contrastive embedding, where two classification problems are constructed in the embedded feature space to encourage the features to capture strong discriminative information. By employing additional taxonomy knowledge, hierarchical labels are obtained to calculate multi-prototypes for each class in [48]. Constraining the synthesized samples close to all their corresponding prototypes will encourage the synthesized samples to capture the hierarchical correlations. Inspired by space-aligned embedding, semantic rectifying GAN is proposed [112], in which a semantic rectifying loss is designed to enhance the discriminativeness of semantics under the guidance of visual relationships and two pre- and post-reconstructions (used to keep the consistency between synthesized visual and semantic features). Considering that the original semantics might not be discriminative enough, disentangling class representation generative adversarial network [113] is proposed to search automatically discriminative representations by a multi-modal triplet lossthat utilizes multi-modal information.

2.5.3 Muti-Architecture Based Methods

Since GAN based methods tend to over-fit and VAE based methods tend to under-fit, some works adopt both the frameworks in their methods. CVAE is trained with a regressor against a discriminator in [114]. The framework proposed in [96] shares the decoder in conditional VAE as the generator for a conditional WGAN. This framework is also applicable for the transductive scenario by training another discriminator for unseen samples. In this model, a pre-trained classifier on the training set is adopted as classification supervision contributing to more discriminating synthesized samples. The dual VAE is trained with two additional discriminators in [115] based on the sum of the dual VAE loss and the conditional WGAN loss to avoid blurry synthesized samples.

2.5.4 Meta Learning Based Methods

As a meta learning process proposed in [116], Model-Agnostic Meta-Learning is referred to in zero-shot learning to train generative models. First, each meta task contains meta training and meta validation set which are sampled from the training set. The model optimized over each meta task can become more generalized due to the divergence of the meta tasks. Moreover, the optimization process for parameters is also conducted in a meta way. Rather than learning parameters performing the best over tasks, the target here is to learn the most adaptive ones for all the meta tasks. In other words, the learned parameters may not achieve the best performance in the current training meta task, but may attain significant performance in different tasks with few-step training on them.

A conditional WGAN with a pre-trained classifier is optimized under this meta learning strategy in [117]. In [118], Model-Agnostic Meta-Learning is applied to the complex framework where the conditional VAE shares the decoder as the generator for a conditional WGAN. The parameters of the encoder, decoder (generator), and discriminator are optimized under such strategy to generate high-fidelity samples only relying on a few number of training examples from seen classes. Pseudo labels for the different meta task distribution is utilized for a task discriminator in [119]. During the training, once the task discriminator is defeated, the encoder is able to align multiple diverse tasks into a unified distribution. With the aligned embedded features, a conditional GAN which generates the pseudo embedded features from Gaussian noises and attributes with a learnable classifier can be trained under the meta learning strategy.

2.6 Implementation Details

2.6.1 Benchmarks and Evaluation Criteria

Benchmarks. To avoid overlapping between unseen classes and training classes used for the pre-trained feature extractor, specific data splits for five commonly used benchmarks are proposed with extracted features in [53]. This work has greatly facilitated the evaluation of models for subsequent studies. Here, we will focus on four of them to set up a summary of the comparisons between the most representative methods.

Table 2.1: Statistics for AwA1, AwA2, aPY, CUB and SUN in terms of granularity, class size, sample size and sample divergence.

Dataset	Size	Granularity	Semantic type	Size of semantics	Class size		Sample size		
					train(seen)	unseen	train	test$_{seen}$	test$_{unseen}$
AwA1	medium	coarse	Attributes	85	40	10	19832	4958	5685
AwA2	medium	coarse	Attributes	85	40	10	23527	5882	7913
CUB	medium	fine	Attributes	312	150	50	7057	1764	2967
aPY	small	coarse	Attributes/text	64	20	12	5932	1483	7924
SUN	medium	fine	Attributes	102	645	72	10320	2580	1440

Animals with Attributes (AwA2) [53] contains 30,475 images from public web sources for 50 highly descriptive animal classes with at least 92 labelled examples per class. For example, the attributes include *stripes*, *brown*, *eats fish* and so on. Caltech-UCSD-Birds-200-2011 datasets (CUB)) [34] is a fine-grained dataset with a large number of classes and attributes, containing 11,788 images from 200 different types of birds annotated with 312 attributes. SUN Attribute (SUN) [120] is a fine-grained dataset, medium-scale in class number, containing 14,340 scene images annotated with 102 attributes, e.g. *sailing/boating*, *glass*, and *ocean*. The dataset Attribute Pascal and Yahoo (aPY) [121] is a small-scale dataset with 64 attributes and 32 object classes, including animals, vehicles, and buildings. Fig. 2.8 present an example of generalized zero-shot task on AwA2.

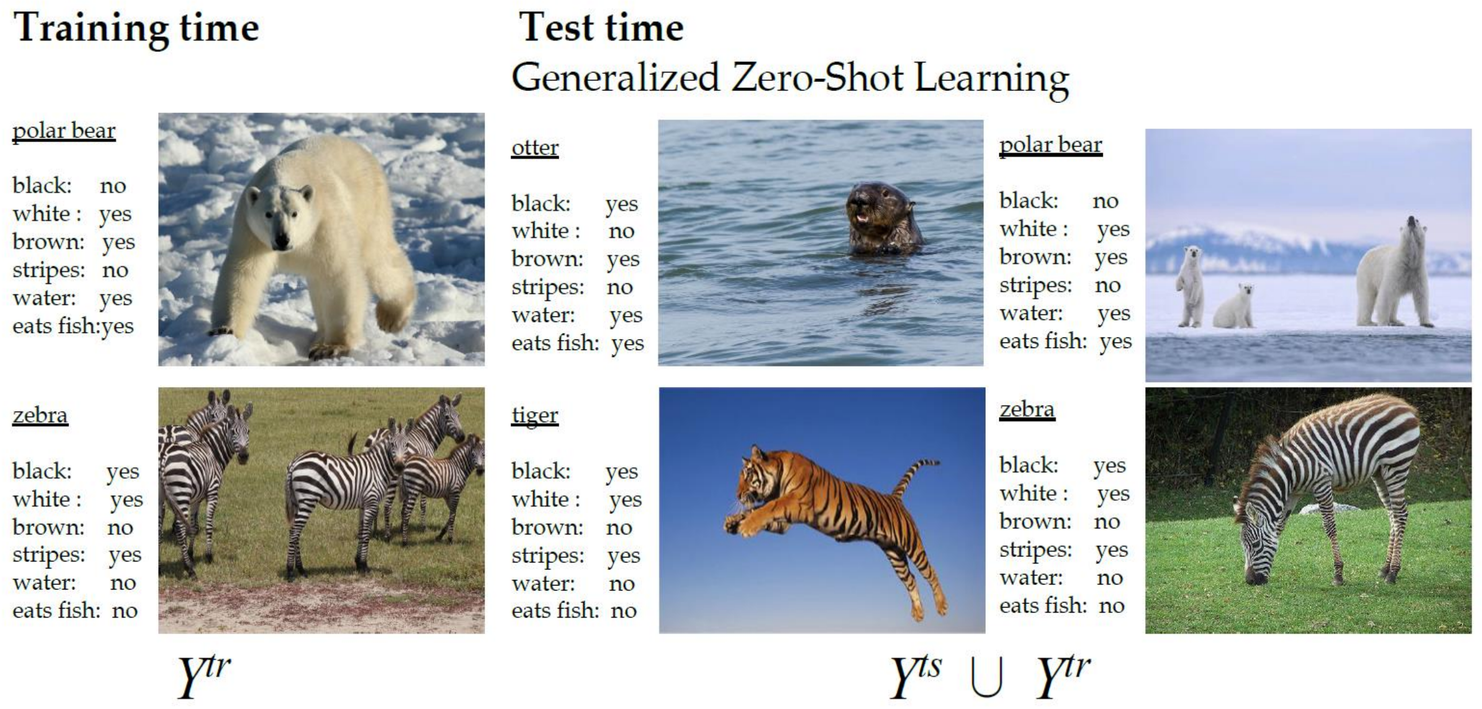


Fig. 2.8: Schematic diagram of generalized zero-shot learning on AwA2

We recommend the splitting strategy used in [53] for the datasets, since most of the current methods are evaluated following such protocol. More details can be found in Table 2.1. Notice that Animals with Attributes (AwA1) [33] is not introduced here since it is not publicly available due to the copyright issue. It is worthy to mention that there are some other datasets adopted in zero-shot learning, e.g. the large scale dataset ImageNet-1K [55], the small scale fine-grained dataset Oxford Flower-102 (FLO) [122], and fMRI (functional Magnetic Resonance Images) [123]. Since they are not the most commonly used as the previous four benchmarks and some of the experimental settings on them are inconsistent in different studies, we will not go into details about them. Some evaluation protocols for them can be referred to in [25, 26, 67, 124, 125].

Evaluation criteria. Compared with the conventional zero-shot learning task, the generalized one can better evaluate the capability for constructing recognition conception of unseen classes, thus are selected for demonstrating the performances of the methods in this article. Since the model needs to discriminate between seen and unseen classes simultaneously ensuring correct classification, the performance of both seen and unseen classes needs to be measured. Following the most commonly used generalized task criteria defined in [53], we define ACC_S and ACC_U as two average per-class top-1 accuracies to measure the classification performances on seen and unseen classes as

$$ACC_S = \frac{1}{K^S} \sum_{k=1}^{K^S} \frac{TP_k}{N_k}, \tag{2.10}$$

$$ACC_U = \frac{1}{K^U} \sum_{k=1}^{K^U} \frac{TP_k}{N_k}, \tag{2.11}$$

where TP_k denotes the number of the true positive samples that is correctly predicted in kth-class and the N_k denotes the number of the instances in kth-class. In other words, the top-1 prediction accuracy for each class is considered equally independent of the sample size of that class. Specifically, the candidates for the predicted labels in such classification are all the classes but not singly those of seen or unseen. Then the comprehensive performance in generalized zero-shot learning task can be evaluated by the harmonic mean of these accuracies defined as follows:

$$H = \frac{2 \times ACC_S \times ACC_U}{ACC_S + ACC_U}. \tag{2.12}$$

2.6.2 Comparisons with Implementation Details

In this section, we will summarize the reported performance of the representative methods with implementation details. Tables 2.2, 2.3 & 2.4, and Tables 2.5 & 2.6 present the comparisons on the methods on AwA2, CUB, aPY, and SUN benchmarks in types of embedding methods and generative methods, respectively. Average ranking denotes the mean of the ranks of H values for the four datasets, "–" denotes the results were not reported, I, ST and T represent the inductive, semantic transductive, and transductive training scenarios, respectively. Superscript with number denotes the same methods corresponding to different implementation setups. The results are obtained from the corresponding published papers or the comparisons provided in [53] and all the H values are displayed in boldface. The listed methods are roughly sorted according to the published periods and performances for different scenarios. Here we regard the ResNet101 pre-trained on ImageNet 1K outputs features in 2,048 dimensions as the settled backbone for extracting visual features. In Table 2.2 and 2.3 the backbone is not changed and most of the embedding methods summarized in Table 2.4 adjust the backbone. The column Extra in the table contains several indicators about the implementation details that could boost the performance of the model, which are listed as follows.

- **Backbone modification.** Indicator $\mathbb{B}$ denotes that the architecture of the feature extractor is modified to improve the obtained visual feature space. Such modification includes designing additional attention modules accompanied with the backbone, repeatedly adopting the feature extractor to extract the divided image regions to obtain multiple features, employing the multi-channel feature map layer before pooling in the pre-trained ResNet, or constructing the backbone with other advanced neural network architectures.

- **Fine-tuning.** Indicator $\mathbb{F}$ specifies that the borrowed backbone is fine-tuned during training. As in most of the methods, the pre-trained backbone is frozen and the extracted visual features are directly employed as the training samples, their performances are evaluated under the same feature space. On the contrary, the methods fine-tuning the backbone with the proposed model lead to different feature spaces, thus the evaluation of them can not be considered strictly in the same setting as the methods without fine-tuning.

- **Additional knowledge.** Indicator $\mathbb{K}$ denotes that the information commonly not included in the benchmarks is leveraged to improve the performance of the model. Note that the pre-trained DNN is not counted as additional knowledge as this is somehow a common setting in zero-shot learning. Such additional knowledge includes taxonomy knowledge as hierarchical labels, correlations between attributes captured by manually defined or through word embedding models trained on extra text corpora, captured gaze point, and data augmentation technology.

For those generative methods in Table 2.5 & 2.6, the scenarios are describing the training process of the generator only. As described in the previous section, generative methods need first generate pseudo samples of unseen classes based on the corresponding semantic descriptions and then train the classifier. Though most of these methods follow the inductive scenario of using only seen classes information in training a generator, using samples generated based on unseen semantic information to train the classifier can be considered to break the unseen principle. Although some of the methods apply k-nearest neighbors as the classifier which does not require training to avoid this ambiguity, plenty of those generative methods are designed with a linear or non-linear classifier. The effectiveness of different types of classifiers can be influenced by a number of factors, such as differences in the database, differences in the distribution of generated samples due to the structure of the model, differences in the number of generated samples, and whether or not generated samples are used to train the classifier for seen classes as well. It is difficult and inappropriate to evaluate all the generative methods under unique structure of classifier such as the k-nearest neighbors to keep consistent setting with the embedding methods. As a result, compared with the embedding methods of Table 2.2 & 2.3 in the same period, most of those generative methods of Table 2.5 & 2.6 appear to achieve better performance. To construct rigorous comparisons, we advocate evaluating the embedding and generative methods separately. Moreover, as the current best models in both of these two families under the inductive scenario, i.e. IPN [98] and CE-GZSL [10], perform quite similar actually, we believe embedding and generative methods are of equal importance in zero-shot learning.

Moreover, as shown in these four tables, the methods with modified or fine-tuned backbones outperform their original counterparts published in the same year. Especially, the effectiveness of fine-tuning has been verified in the embedding method DVBE [86] and the generative method f-VAEGAN-D2 [96]. Fine-tuning leads to 2.4%, 12.0%, 3.4%, 2.1% absolute increment in the H values for DVBE on AwA2, CUB, aPY and SUN, respectively. Similar improvements can also be observed for the f-VAEGAN-D2 under the inductive and transductive scenarios. These results imply that fine-tuning the backbone overall benefits the generalized zero-shot learning especially on the CUB benchmarks.

Most outstanding embedding and generative methods under the inductive scenario often utilize additional knowledge. In this way, more adequate information can help better construct concepts of unseen classes through knowledge of seen classes. The validity of the employed additional knowledge is not accurately presented in these comparison

Table 2.2: Comparisons of embedding methods on AwA2, CUB, aPY and SUN.

Method	Scenario	Extra	AwA2			CUB			aPY			SUN			Average
			ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ranking
DeViSE (2013) [67]	I		17.1	74.7	**27.8**	23.8	53.0	**32.8**	4.9	76.9	**9.2**	16.9	27.4	**20.9**	13.8
SSE (2015) [64]	I		8.1	82.5	**14.8**	8.5	46.9	**14.4**	0.2	78.9	**0.4**	2.1	36.4	**4.0**	17.8
ESZSL (2015) [65]	I		5.9	77.8	**11.0**	12.6	63.8	**21.0**	2.4	70.1	**4.6**	11.0	27.9	**15.8**	17.0
SJE (2015) [38]	I		8.0	73.9	**14.4**	23.5	59.2	**33.6**	3.7	55.7	**6.9**	14.7	30.5	**19.8**	14.5
LatEm (2016) [41]	I		11.5	77.3	**20.0**	15.2	57.3	**24.0**	0.1	73.0	**0.2**	14.7	28.8	**19.5**	16.5
SAE (2017) [56]	I		1.1	82.2	**2.2**	7.8	54.0	**13.6**	0.4	80.9	**0.9**	8.8	18.0	**11.8**	18.3
DEM (2017) [13]	I		30.5	86.4	**45.1**	19.6	57.9	**29.2**	11.1	75.1	**19.4**	20.5	34.3	**25.6**	12.8
PSR (2018) [60]	I		20.7	73.8	**32.3**	24.6	54.3	**33.9**	13.5	51.4	**21.4**	20.8	37.2	**26.7**	11.5
LESAE (2018) [126]	I		21.8	70.6	**33.3**	24.3	53.0	**33.3**	12.7	56.1	**20.1**	21.9	34.7	**26.9**	11.8
RN (2018) [14]	I		30.0	93.4	**45.3**	38.1	61.1	**47.0**	–	–	–	–	–	–	9.5

Table 2.3: Comparisons of embedding methods on AwA2, CUB, aPY and SUN.

Method	Scenario	Extra	AwA2			CUB			aPY			SUN			Average
			ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ranking
TVN (2019) [51]	I		–	–	–	26.5	62.3	**37.2**	16.1	66.9	**25.9**	22.2	38.3	**28.1**	9.7
PQZSL (2019) [61]	I		31.7	70.9	**43.8**	43.2	51.4	**46.9**	27.9	64.1	**38.8**	35.1	35.3	**35.2**	8.0
CRnet (2019) [82]	I		52.6	78.8	**63.1**	45.5	56.8	**50.5**	32.4	68.4	**44.0**	36.5	34.1	**35.3**	3.8
DTNet (2020) [70]	I		–	–	–	44.9	53.5	**48.9**	25.5	59.9	**35.5**	–	–	–	8.0
LAF (2020) [63]	I		50.4	58.5	**54.2**	43.7	52.0	**47.5**	33.8	49.0	**40.0**	36.0	36.6	**36.3**	5.5
DVBE (2020)[1] [86]	I		63.6	70.8	**67.0**	53.2	60.2	**56.5**	32.6	58.3	**41.8**	45.0	37.2	**40.7**	2.5
LRSG-ZSL (2021) [76]	I		60.4	84.9	**70.6**	48.5	49.3	**48.9**	30.3	76.2	**43.4**	51.2	22.4	**31.2**	4.3
IPN (2021) [98]	I		67.5	79.2	**72.9**	60.2	73.8	**66.3**	37.2	66.0	**47.6**	–	–	–	1.0
TCN (2019) [72]	ST		61.2	65.8	**63.4**	52.6	52.0	**52.3**	24.1	64.0	**35.1**	31.2	37.3	**34.0**	5.5

Table 2.4: Comparisons of embedding methods on AwA2, CUB, aPY and SUN.

Method	Scenario	Extra	AwA2			CUB			aPY			SUN			Average
			ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ACC_U	ACC_S	**H**	ranking
LFGAA^1 (2019) [92]	I	$\mathbb{B}\,\mathbb{F}$	27.0	93.4	**41.9**	36.2	80.9	**50.0**	–	–	–	18.5	40.0	**25.3**	11.0
AREN (2019) [89]	I	$\mathbb{B}\,\mathbb{F}$	54.7	79.1	**64.7**	63.2	69.0	**66.0**	30.0	47.9	**36.9**	40.3	32.3	**35.9**	7.0
APN (2020) [94]	I	$\mathbb{B}\,\mathbb{F}\,\mathbb{K}$	56.5	78.0	**65.5**	65.3	69.3	**67.2**	–	–	–	41.9	34.0	**37.6**	6.3
DVBE (2020)^2 [86]	I	$\mathbb{F}$	62.7	77.5	**69.4**	64.4	73.2	**68.5**	37.9	55.9	**45.2**	44.1	41.6	**42.8**	3.5
GEM-ZSL (2021) [42]	I	$\mathbb{B}\,\mathbb{F}\,\mathbb{K}$	64.8	77.5	**70.6**	64.8	77.1	**70.4**	–	–	–	38.1	35.7	**36.9**	4.7
DAZLE (2020) [99]	ST	$\mathbb{B}$	60.3	75.7	**67.1**	56.7	59.6	**58.1**	–	–	–	52.3	24.3	**33.2**	8.3
RGEN (2020) [90]	ST	$\mathbb{B}\,\mathbb{F}$	67.1	76.5	**71.5**	60.0	73.5	**66.1**	30.4	48.1	**37.2**	44.0	31.7	**36.8**	4.8
AGAN (2022) [101]	ST	$\mathbb{B}$	64.1	80.3	**71.3**	67.9	71.5	**69.7**	–	–	–	40.9	42.9	**41.8**	3.7
LFGAA^2 (2019) [92]	T	$\mathbb{B}\,\mathbb{F}$	50.0	90.3	**64.4**	43.4	79.6	**56.2**	–	–	–	20.8	34.9	**26.1**	10.0
QFSL (2018) [84]	T	$\mathbb{F}$	66.2	93.1	**77.4**	71.5	74.9	**73.2**	–	–	–	51.3	31.2	**38.8**	2.3
STHS-S2V (2021) [102]	T		91.4	92.3	**91.8**	71.2	74.5	**72.8**	–	–	–	70.7	44.8	**54.8**	1.3

Table 2.5: Comparisons of generative methods on AwA2, CUB, aPY and SUN.

Method	Scenario	Extra	AwA2			CUB			aPY			SUN			Average
			ACC_U	ACC_S	H	ACC_U	ACC_S	H	ACC_U	ACC_S	H	ACC_U	ACC_S	H	ranking
f-CLSWGAN (2018) [6]	I		–	–	–	43.7	57.7	**49.7**	–	–	–	42.6	36.6	**39.4**	18.0
SRGAN (2019) [112]	I		–	–	–	31.3	60.9	**41.3**	22.3	78.4	**34.8**	22.1	38.3	**27.4**	15.3
LisGAN (2019) [111]	I		–	–	–	46.5	57.9	**51.6**	–	–	–	42.9	37.8	**40.2**	17.0
GDAN (2019) [114]	I		32.1	67.5	**43.5**	39.3	66.7	**49.5**	30.4	75.0	**43.4**	38.1	89.9	**53.4**	11.0
CADA-VAE (2019) [8]	I		55.8	75.0	**63.9**	51.6	53.5	**52.4**	–	–	–	47.2	35.7	**40.6**	15.7
f-VAEGAN-D2^1 (2019) [96]	I		57.6	70.6	**63.5**	48.4	60.1	**53.6**	–	–	–	45.1	38.0	**41.3**	15.0
f-VAEGAN-D2^2 (2019) [96]	I	$\mathbb{F}$	57.1	76.1	**65.2**	63.2	75.6	**68.9**	–	–	–	50.1	37.8	**43.1**	9.3
ZSML (2020) [117]	I		58.9	74.6	**65.8**	60.0	52.1	**55.7**	36.3	46.6	**40.9**	–	–	–	10.0
DE-VAE (2020) [9]	I		58.8	78.9	**67.4**	52.5	56.3	**54.3**	–	–	–	45.9	36.9	**40.9**	12.0
DR-VAE (2021) [106]	I		56.9	80.2	**66.6**	51.1	58.2	**54.4**	–	–	–	36.6	47.6	**41.4**	11.7

Table 2.6: Comparisons of generative methods on AwA2, CUB, aPY and SUN.

Method	Scenario	Extra	AwA2			CUB			aPY			SUN			Average
			ACC_U	ACC_S	H	ACC_U	ACC_S	H	ACC_U	ACC_S	H	ACC_U	ACC_S	H	ranking
M-VAE (2021) [107]	I		61.3	72.4	**66.4**	57.1	62.9	**59.8**	–	–	–	42.4	58.7	**49.2**	8.3
DGN (2021) [127]	I		60.1	76.4	**67.3**	53.8	61.9	**57.6**	36.5	61.7	**45.9**	48.3	37.4	**42.1**	8.5
DCRGAN (2021) [113]	I		–	–	–	55.8	66.8	**60.8**	37.2	71.7	**49.0**	47.1	38.5	**42.4**	6.3
CE-GZSL (2021) [10]	I		63.1	78.6	**70.0**	63.9	66.8	**65.3**	–	–	–	48.8	38.6	**43.1**	7.0
TGMZ (2021) [119]	I	$\mathbb{K}$	64.1	77.3	**70.1**	60.3	56.8	**58.5**	34.8	77.1	**48.0**	–	–	–	6.0
CKL+TR (2021) [48]	I	$\mathbb{K}$	61.2	92.6	**73.7**	57.8	50.2	**53.7**	30.8	78.9	**44.3**	–	–	–	8.0
APN+f-VAEGAN-D2 (2020) [94]	I	$\mathbb{B}\,\mathbb{F}\,\mathbb{K}$	62.2	69.5	**65.6**	65.7	74.9	**70.0**	–	–	–	49.4	39.2	**43.7**	8.0
AFGN (2022) [101]	ST	$\mathbb{B}$	68.1	82.9	**74.7**	69.8	77.1	**73.2**	–	–	–	53.1	45.9	**49.2**	3.7
f-VAEGAN-D2^3 (2019) [96]	T		84.8	88.6	**86.7**	61.4	65.1	**63.2**	–	–	–	60.6	41.9	**49.6**	4.3
f-VAEGAN-D2^4 (2019) [96]	T	$\mathbb{F}$	86.3	88.7	**87.5**	73.8	81.4	**77.3**	–	–	–	54.2	41.8	**47.2**	3.0
STHS-WGAN (2021) [102]	T		94.9	92.3	**93.6**	77.4	74.5	**75.9**	–	–	–	67.5	44.8	**53.9**	1.3

tables. One can refer to each relevant paper for more details.

When the methods in all the scenarios are compared together, for both the embedding and generative methods, one can find that methods STHS-S2V and STHS-WGAN [102] in the transductive scenario, attain the highest H values on most of the benchmarks. The unlabelled data with unseen classes attributes provide a detailed target guidance for the transformation of categorical knowledge, thus making such scenario the easiest generalized case. If one takes TCN [72] as the most similar method of RN [14] under the semantic transductive scenario (via accessing the unseen attributes during training), 18.1% and 5.3% absolute improvement have been achieved on AwA2 and CUB, respectively. Moreover, the gaps between the performances of the LFGAA [92] under both the semantic transductive and inductive scenarios also confirm the contribution of unseen attributes in training the model in generalized zero-shot learning.

In this section, the type of the classifiers or the number of synthesized pseudo samples for training is not collated here, as the impact of these implementation details on model performance is uncertain when the models are structured differently or applied on different databases. We focus on specifying differences in the implementation details which commonly lead to explicit changes in performance among the current representative methods. On the one hand, we acknowledge the contribution of those methods of adopting additional knowledge or modifications; on the other hand, showcasing numerical comparisons between different methods with different implementation settings may not be rigorous enough, which could lead to a misleading assessment of the capability of the model. We advocate researchers set up comparisons between the methods under the same implementation settings. Moreover, all the additional operations and/or auxiliary knowledge appear critically important and thus should keep clear and be stated explicitly for fair and precise evaluations.

2.7 Summary and Discussion

In this chapter, we have provided a comprehensive survey of image classification with zero-shot learning. We have put one main focus of this survey on the implementation issues. Particularly, with the methods steadily improved, different problem settings, and diverse experimental setups have emerged, and thus we have examined three implementation details that can boost the performance of zero-shot learning, i.e. whether the backbone structure has been modified, whether fine-tuning has been conducted, and whether additional knowledge has been used. By annotating these experimental details, we have collected a more careful comparison between various zero-shot methodologies. While generative methods appear to outperform embedding methods overall, we argue that the performance difference may be due to the different settings, thus suggesting that it may be fairer to compare them separately. Moreover, we observe that the current best models in both families perform quite similar under the inductive scenario. Thus we believe embedding and generative methods are of equal importance in zero-shot learning.

Through the comparisons, we can find that the zero-shot learning model has achieved outstanding performance in the transductive scenario. For the inductive scenario as the most rigorous one, the performances of the models still needs to be improved. While improving the backbone to obtain a more appropriate visual feature space can effectively improve the accuracies of the embedding methods, changing the space corresponding to the samples can be regarded as adjusting the difficulty of the original task. In contrast, the generative class of methods can achieve a high-level comprehensive performances based on the original visual feature space. This indicates that those embedding methods with settled backbone for inductive scenarios still have the potential to be improved.

Chapter 3

Efficient Self-Focus Mechanism for Coarse-Grained Generalized Zero-Shot Learning

Since most zero-shot learning tasks in reality are challenging where both the descriptions and samples are unavailable during training, in this chapter, we focus on the zero-shot learning under the inductive scenario in this paper. Following the embedding idea, we choose the visual feature space as the embedding target space and use the nearest neighbor search based on Euclidean distance to achieve classification. Additionally, a self-focus module, constructed as a 1-layer fully connected neural network with the activation function $sigmoid$ followed by $softmax$ operation, is developed to prevent the over-fitting during training. Via the focus ratios calculated by this module, the correlations between the location and the importance of each dimension in the embedding space are considered during the optimization. Since the embedding model and the proposed module are optimized together to minimize the intra-class distance, the embedding model only needs to focus on the ratio which the proposed module considers important. In other words, by applying the proposed self-focus mechanism, the over-fitting knowledge will be apportioned to the self-focus module, and the over-fitting problem for the embedding model can be alleviated. A series of experiments show that the proposed self-focus mechanism achieves outstanding performance on coarse-grained zero-shot classification tasks.

3.1 Methods

To give a better explanation, we will first define the zero-shot learning and generalized zero-shot learning tasks under the inductive scenario. We then introduce the model architecture in detail.

3.1.1 Problem Definition

Denote $\boldsymbol{X} = \{x_1, \ldots, x_N\}$, $\boldsymbol{Y} = \{y_1, \ldots, y_N\}$, $\boldsymbol{A} = \{a_1, \ldots, a_K\}$ as visual feature, class label, and semantic attribute sets respectively, with the total sample size N and class number K. First we divide the category or class set into two parts, $K = K^S + K^U$, and denote the first K^S classes as seen classes for convenience. Then the dataset is divided into training and test sets, as $\boldsymbol{X} = \boldsymbol{X}_{tr} \cup \boldsymbol{X}_{te}$, $\boldsymbol{Y} = \boldsymbol{Y}_{tr} \cup \boldsymbol{Y}_{te}$ and $N = N_{tr} + N_{te}$, where $\boldsymbol{X}_{tr} \cap \boldsymbol{X}_{te} = \emptyset$. Specifically, the training set only contains samples from seen classes, while the test set may contain those of both seen and unseen, i.e. $\forall y_i \in \boldsymbol{Y}_{tr}, y_i \leq K^S$ and $\forall y_m \in \boldsymbol{Y}_{te}, 0 < y_i \leq K$. Given a training set $\boldsymbol{D}_{tr} = \{(x_i, y_i, a_{y_i}) | x_i \in \boldsymbol{X}_{tr}\}$, the target of zero-shot learning is to train a classifier $C(x_i, a_1, ..., a_{K^S}) = \hat{y}_i$ on $\boldsymbol{D}_{tr}$ to achieve classification on a test set $\boldsymbol{D}_{te}$, where $\boldsymbol{D}_{te} = \{(x_m, y_m, a_{y_m}) | x_m \in \boldsymbol{X}_{te}, K^S < y_m \leq K\}$ for conventional zero-shot learning, and $\boldsymbol{D}_{te} = \{(x_m, y_m, a_{y_m}) | x_m \in \boldsymbol{X}_{te}, 0 < y_m \leq K\}$ for generalized zero-shot learning. In this chapter, we only focus on this strict generalized zero-shot learning case.

3.1.2 Model Architecture

We follow the DEM [13] to avoid aggravating the hubness problem[128]. Fig. 3.1 shows the schematic diagram of the entire framework of the proposed method, in which the processes of the embedding, optimization, and identification are denoted by the paths in purple, red, and blue, respectively.

Embedding. For each image sample, we use a pre-trained DNN to obtain a d-dimensional feature vector x_i, and settle this feature space as the target embedding space. Then we construct a learnable embedding function $f(\cdot)$ as an 1-layer fully connected neural network with activation function $relu$. The corresponding class attributes a_{y_i} of x_i will be mapped to the target space as the class prototype $f(a_{y_i})$. Therefore, the squared difference between the feature and the prototype for each dimension can be calculated as follows:

$$SD(x_i, f(a_{y_i})) = (x_i - f(a_{y_i}))^2. \tag{3.1}$$

Optimization. In order to associate the location with its corresponding dimensional importance, a self-focus module $W(\cdot)$ is designed as a 1-layer fully connected neural network with the activation function $sigmoid$ then followed by $softmax$ operation. With this module, a focus ratio vector for each prototype can be obtained as $W(f(a_{y_i}))$. The focused difference is defined in the following:

$$FD(x_i, f(a_{y_i})) = SD(x_i, f(a_{y_i})) \odot W(f(a_{y_i})), \tag{3.2}$$

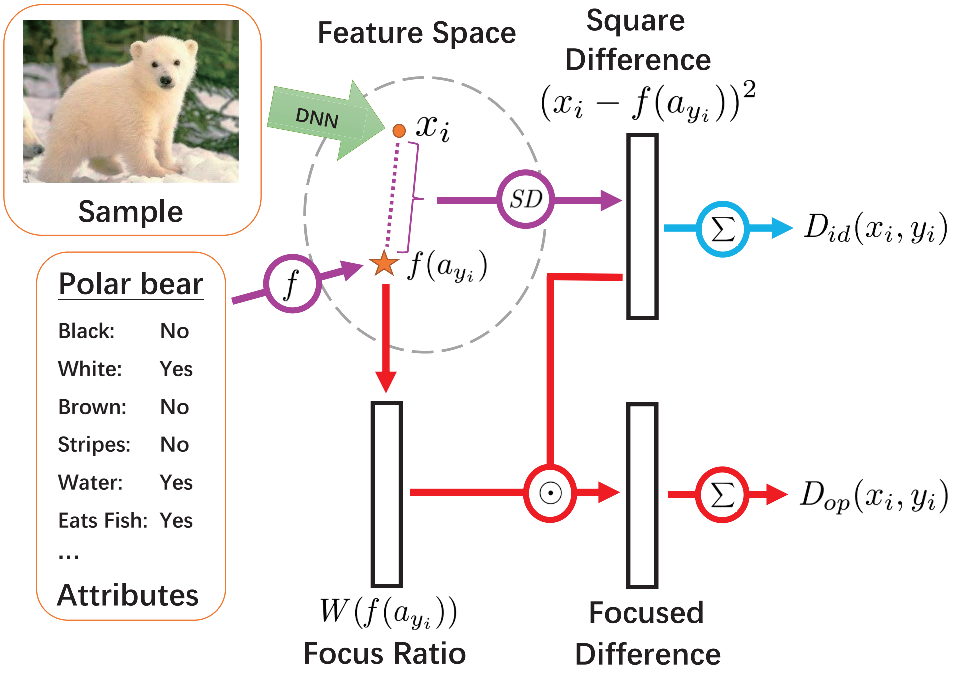


Fig. 3.1: Schematic diagram of the proposed self-focus deep embedding model.

where $\odot$ denotes element-wise multiplication. By adding up this focused difference for all the dimensions, we obtain the distance for optimization as

$$D_{op}(x_i, y_i) = \sum^{d} FD(x_i, f(a_{y_i})). \tag{3.3}$$

We can define the objective function as

$$\mathbb{L} = \sum_{i=1}^{N_{tr}} D_{op}(x_i, y_i) + \lambda \|\boldsymbol{\theta}\|^2, \tag{3.4}$$

where $\boldsymbol{\theta}$ represents all the parameters in $f(\cdot)$ and $W(\cdot)$, and λ denotes the regularization weight.

Identification. The designed self-focus module only participates in the optimization process. Therefore, the sum of the squared differences between an instance x_j and a prototype $f(a_c)$, which is the square of the Euclidean distance, will be directly used as the identification distance as below:

$$D_{id}(x_j, c) = \sum^{d} SD(x_j, f(a_c)). \tag{3.5}$$

Through the nearest neighbor search, the predicted label $\hat{y}_j$ for the instance x_j is calculated as the class holding the least identification distance as below:

$$\hat{y}_i = \underset{0<c\leq K}{\arg\min}\, D_{id}(x_i, c). \tag{3.6}$$

3.2 Target Embedding Space

In this chapter, we adopted an embedding method as our baseline where the semantic attributes are mapped to the visual feature space. The reason for selecting visual feature space as the target space is that such mapping direction will avoid aggravating the hubness problem. Here, we will first introduce the hubness problem in zero-shot learning and then present an explanation of why such mapping direction is relevant.

To measure how hubby a point in the search space is with respect to a sample set, one can count the number of occurrences of the point as the k-nearest neighbor of the sample in the set. The point with high occurrence frequency is called the hub. The hubness problem commonly occurs in high-dimensional spaces [129]. When the dimensionality is increased, the nearest neighbour search will return the hubs as the most likely result. It is shown in [130, 131] that the hubness problem is actually related to the pairwise similarities between the points. As the dimensionality of the space increases, the pairwise similarities between the points in a set tend to converge to a constant and the standard deviation converges to 0, thus causing an increase in hubness.

Consider the matrix representations of two datasets S_A and S_B in different space with a linear map W_m, with the objective function as

$$\mathcal{L}(W_M) = ||S_B - W_m S_A||^2 + \lambda ||W_m||^2. \tag{3.7}$$

We can solve this regression problem with a closed-form solution

$$W_m = S_B S_A^T (S_A S_A^T + \lambda I)^{-1}. \tag{3.8}$$

Then we have

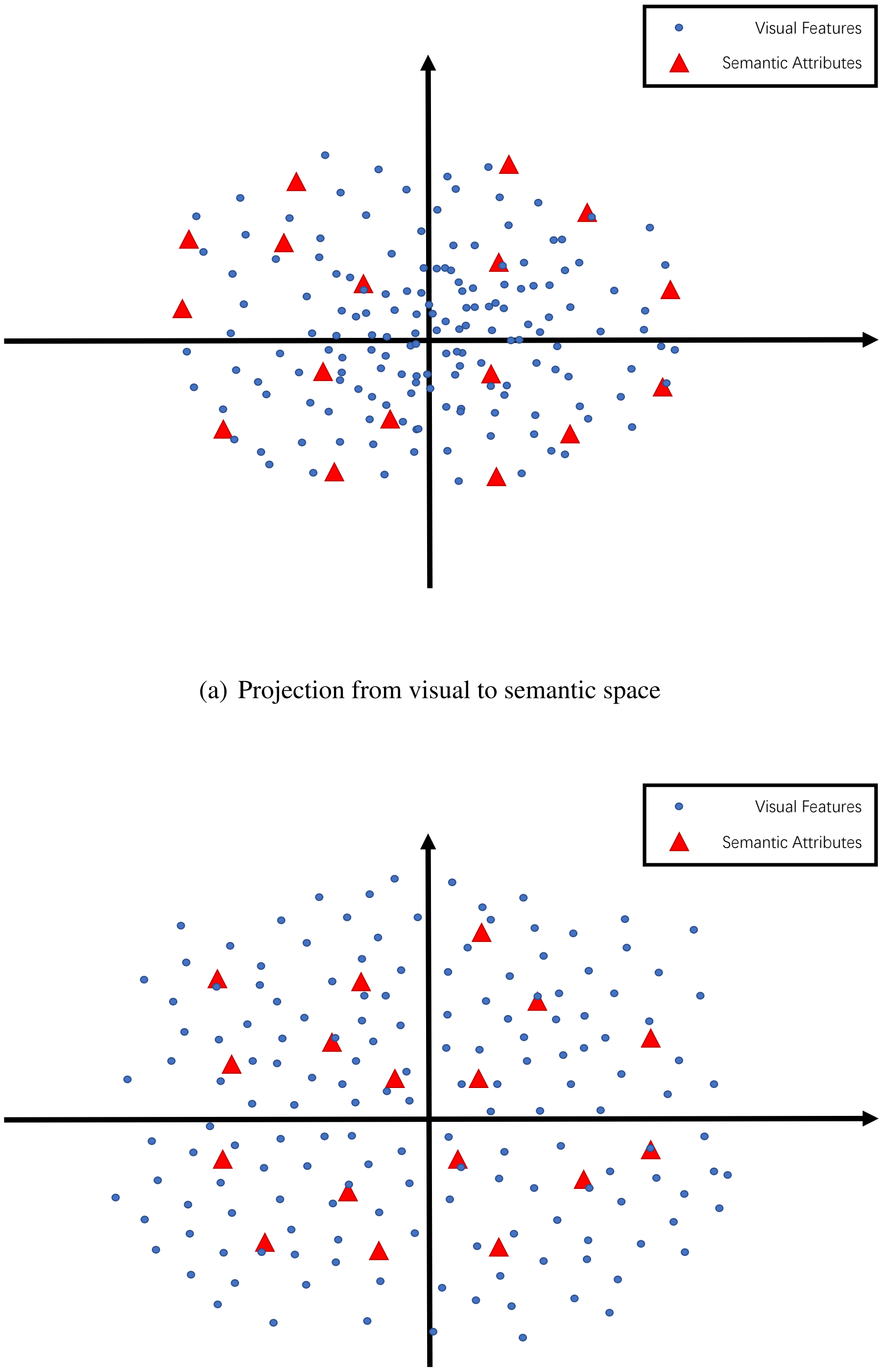
$$\begin{aligned} ||W_m S_A||^2 &= ||S_B S_A^T (S_A S_A^T + \lambda I)^{-1} S_A||^2 && (3.9) \\ &\leqslant ||S_B||^2 ||S_A^T (S_A S_A^T + \lambda I)^{-1} S_A||^2. && (3.10) \end{aligned}$$

By denote the largest single value of S_A as σ, we can further obtain

$$||S_A^T (S_A S_A^T + \lambda I)^{-1} S_A||^2 = \frac{\sigma^2}{\sigma^2 + \lambda} \leqslant 1. \tag{3.11}$$

Thus, we could obtain $||W_m S_A||^2 \leqslant ||S_B||^2$ by substituting Eq.3.11 into Eq.3.10. This inequality indicates that the mapped source dataset $W_m S_A$ seems to be closer to the origin of the space compared with the target dataset B.

Assume the visual feature space and semantic attribute space are the two spaces with a linear map. The 2-D Schematic diagrams shown in Fig. 3.2 present an intuitive explanation of the relation between the mapping direction and the hubness. If we project the



(a) Projection from visual to semantic space

(b) Projection from semantic to visual space

Fig. 3.2: Illustration of the effects of different projection directions on the hubness problem.

visual feature into the semantic attributes space, as shown in Fig. 3.2(a), with a pairwise optimization object function, it becomes the regression problem described above. As a result, the projected points for visual features will be shrunk towards the origin. In this case, the closer the semantic point is to the origin, the more likely it is to be the k-nearest neighbor of visual feature points. In other words, the points for semantic attributes which are closer to the origin are more likely to become the hubs, thus making the hubness problem worse. However, the reversed project direction shown in Fig. 3.2(b) makes the projected attributes shrink to the origin, which at least avoids aggravating the hubness problem. As the situations are the same for the regression based on the deep embedding model [13], in this chapter, we selected the baseline, the deep embedding model (DEM) [13], adopting the visual feature space as the embedding target space.

3.3 Experiments

3.3.1 Datasets and Settings

Datasets. To verify the superiority of the proposed mechanism, three coarse-grained and one fine-grained zero-shot learning benchmarks were selected, which are AwA1 [33], AwA2 [53], aPY [121], and CUB [34], respectively. The extracted features and data splits we use are consistent with the GBU setting [53] to prevent the pre-trained extractors from contacting the unseen classes during their training. More details can be found in Table 3.1.

Table 3.1: Statistics of the benchmarks, including granularity, number of attributes and data splits.

Granularity	Dataset	Number of	Number of Classes	Number of Samples		
		Attributes	(seen+unseen)	Train	Test$_{seen}$	Test$_{unseen}$
Coarse	AwA1	85	40+10	19832	4958	5685
	AwA2	85	40+10	23527	5882	7913
	aPY	64	20+12	5932	1483	7924
Fine	CUB	102	150+50	7057	1764	2967

Settings. All the DNN parameters are initialized with random weights and optimized using the Adam Optimizer [132]. The other setup details are listed in Table 3.2.

Table 3.2: Detailed experimental setups for each dataset.

Dataset	Embedding	Self-Focus	Learning Rate	Regularization Weight	Batch Size
	Layer Structures	Layer Structures		λ	
AwA1	85-1600-2048	2048-2048	1e-4	0.001	64
AwA2	85-1600-2048	2048-2048	1e-4	0.001	64
aPY	64-1600-2048-2048-2048-2048	2048-2048	1e-4	0.0001	64
CUB	312-1200-2048	2048-2048	1e-5	0.01	100

3.3.2 Zero-Shot Recognition

As in the generalized zero-shot learning task, the classification candidates are the combination of all the seen and unseen classes. Three criteria were commonly used to evaluate the model performance, namely, the accuracy of the seen classes ACC_S, the accuracy of the unseen classes ACC_U, and the harmonic mean of these two accuracies

$$H = \frac{2 \times \mathcal{ACC}_S \times \mathcal{ACC}_U}{\mathcal{ACC}_S + \mathcal{ACC}_U}. \tag{3.12}$$

In particular, all the accuracies here denote per-class average Top-1 accuracies, and the harmonic mean is the most important criterion since it measures the overall performance.

Table 3.3: Comparison results evaluated in per-class average Top-1 accuracies on conventional zero-shot learning tasks.

Method	AwA1	AwA2	aPY	CUB
SAE[56]	53.0	54.1	8.3	33.3
PSR-ZSL [60]	–	63.8	38.4	**56.0**
DEM[13]	68.4	67.1	35.0	51.7
Relation Net[14]	68.2	64.2	–	55.6
Triple Verification Net[51]	68.8	–	**41.3**	58.1
Baseline	68.1	67.1	38.9	49.2
Proposed	**70.4**	**68.1**	38.3	48.6

The accuracies of unseen classes in conventional zero-shot learning tasks are shown in Table 3.3 where the highest accuracy of each dataset is highlighted in bold, and the baseline represents the result of DEM in our own implementation. The proposed mechanism improves the performances of the baseline in AwA1 and AwA2. Meanwhile, it has no obvious negative effects on aPY and CUB. Thus, the self-focus mechanism actually leads to a better embedding space where unseen classes are more distinguishable.

Table 3.4 shows the generalized zero-shot learning performance of the proposed model compared with the state-of-the-art methods. In the coarse-grained tasks, our model has a significant increase in the unseen accuracies with a slight drop in unseen accuracies, achieving the best overall performances. The results in CUB also indicate that the proposed mechanism has no negative effects on fine-grained tasks.

3.3.3 Effectiveness of Self-Focus Module

Explanation of Effectiveness on Coarse-Grained Task

The proposed mechanism is designed to alleviate the over-fitting of the embedding model by apportioning the over-fitting knowledge to the proposed self-focus module. To achieve this goal, the self-focus module needs to learn the relation between the class-wise dimensional importance and the location in the target space for calculating the focus ratio. However, if the samples of each class are distributed too evenly in all dimensions, the model may become less effective as all the dimensions become equally important. In other words, to ensure the effectiveness of the proposed mechanism, for each training class, the sample dispersion on each dimension is required to be as unique as possible. From common sense, the intra-class distance for a coarse-grained database should be larger than that of a fine-grained database due to the broader definition of each class. This larger distance indicates a higher possibility of meeting the required unique sample dispersion. As a numerical illustration, for each dataset, we calculate the range $range_{dc}$ and variance var_{dc} of the sample features in each dimension, class by class. Then we define four criteria to measure the uniqueness of sample dispersion with respect to dimensions as below:

$$MMR = Mean^{C}(Mean^{D}(range_{dc})),\ MMV = Mean^{C}(Mean^{D}(var_{dc})),$$
$$MVR = Mean^{C}(Var^{D}(range_{dc})),\ MVV = Mean^{C}(Var^{D}(var_{dc})),$$

where $Mean()$ and $Var()$ represent the processes of calculating the mean and variance, respectively, and the superscript ^{C} and ^{D} denote that the calculations are over classes and dimensions respectively.

As shown in Table 3.5, the coarse-grained datasets obviously enjoy the higher four criteria than that of the fine-grained dataset, which approximately implies the coarse-grained ones have greater differences in sample dispersion in dimensions. Accordingly,

Table 3.4: Comparison results evaluated in per-class average Top-1 accuracies on generalized zero-shot learning tasks.

Method	AwA1			AwA2			aPY			CUB		
	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H
SAE[56]	1.8	77.1	3.5	1.1	82.2	2.2	0.4	**80.9**	0.9	7.8	54.0	13.6
PSR-ZSL [60]	–	–	–	20.7	73.8	32.3	13.5	51.4	21.4	24.6	54.3	33.9
DEM[13]	32.8	84.7	47.3	30.5	86.4	45.1	11.1	75.1	19.4	19.6	57.9	29.2
Relation Net[14]	31.4	**91.3**	46.7	30.0	**93.4**	45.3	–	–	–	**38.1**	61.1	**47.0**
Triple Verification Net[51]	27.0	67.9	38.6	–	–	–	16.1	66.9	25.9	26.5	**62.3**	37.2
Baseline	34.8	83.6	49.1	32.0	85.9	46.6	24.6	78.1	37.4	21.5	48.6	29.8
Proposed	**39.1**	83.2	**53.2**	**39.0**	84.5	**53.4**	**26.2**	78.5	**39.3**	21.9	47.5	30.0

Table 3.5: Comparisons of the datasets on the defined criterions.

Granularity	Dataset	MMR	MMV	MVR	MVV
Coarse	AwA1	4.1591	0.3874	5.3712	0.4665
	AwA2	3.8756	0.3597	5.4429	0.4989
	aPY	3.9562	0.4442	4.9741	1.1907
fine	CUB	1.7838	0.2105	2.1383	0.3043

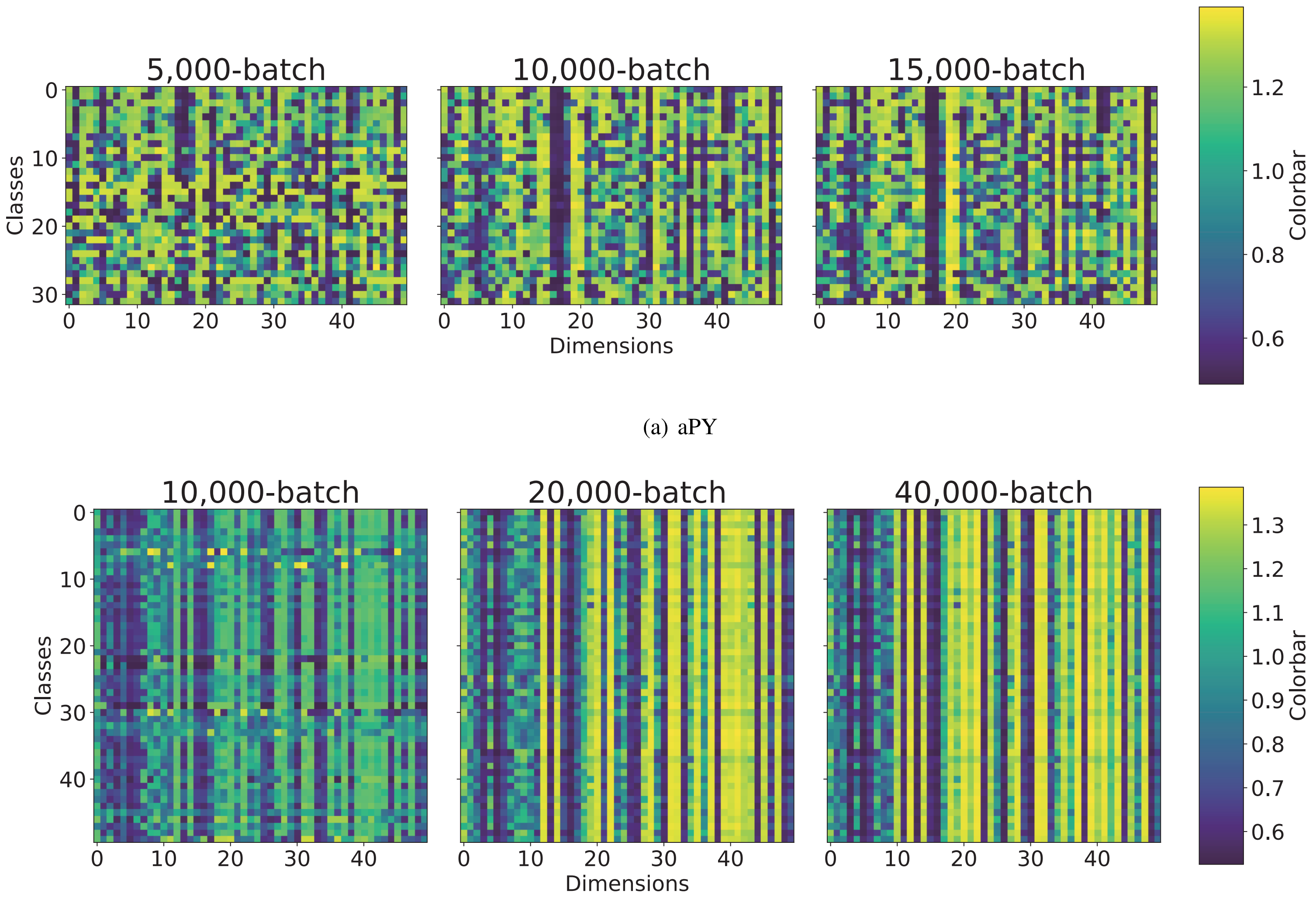
more information about the dimensional importance can be provided to train a better self-focus module, which explains why the proposed model performs significantly on them.

Efficiency of Self-Focus Mechanism

To evaluate the efficiency of the proposed mechanism, we study the performance of the proposed model during training on two coarse-grained datasets. The focus ratios of each category corresponding to the different training stages are shown in Fig. 3.3 in the form of heat maps. Additionally, the focus ratios are scaled by multiplying the number of dimensions d.

Since the dimensionality of the target space is very high (totally 2,048 dimensions), we only select the first 50 dimensions for a clearer display. In the beginning, the calculated focus ratios of each dimension for all categories are irregular. As the training progresses, the proposed self-focus module starts to capture the importance of each dimension based on the sample dispersions. As a result, focus ratios of some dimensions tend to be small for all the classes, focus ratios of some dimensions tend to be large for all the classes, and focus ratios of the rest dimensions vary according to the class. Namely, after a certain number of batches is trained, the embedding model will be optimized mainly based on the loss calculated from important dimensions. Compared with the original optimization process, it indeed avoids the over-fitting caused by over-focusing on all dimensions.

Figs. 3.4 and 3.5 shows the average training loss and overall performance H for each 1,000-batches in the two coarse-grained datasets, AwA2 and aPY. The proposed method initially declines slower than that of the baseline. This is because the self-focus module has not fully effectively learned the relationship between the location and the dimensional importance at the beginning of the training. Consequently, the unstable focus ratios bring more randomness to the optimization direction. As the number of batches increases, the



(a) aPY

(b) AwA2

Fig. 3.3: Partial focus ratios of each class corresponding to different training stages.

loss of the proposed method eventually drops to the same level as that of the baseline, and the overall performance also starts to increase and finally exceeds that of the baseline.

Fair Comparison with Baseline

Due to the architecture of the proposed self-focus module, our whole framework actually uses one more layer of the neural network than the baseline DEM. To fairly verify the effectiveness of the proposed module, we also evaluate the performance of DEM with a deeper neural network structure. Specifically, the extended layer for the deeper DEM has the same input-output size as that of the self-focus module.

The comparison results are demonstrated in Fig. 3.6 and 3.7, where DEM* denotes the baseline results reported on GitHub[1] and the rest of the DEM represents the results obtained in our own implementations corresponding to the structure with different numbers of layers. Especially for the aPY dataset, we select the architecture with a 6-layer neural network as the baseline due to the poor performance of the original one. It is obvious that for the three coarse-grained databases, in the case that extending the original embedding network cannot improve the performance of the model, employing the self-focus mechanism obviously improves the accuracies of unseen classes. Although the accuracies of seen classes have decreased slightly, the magnitudes are much smaller than the gains, which is also a sign of alleviating the over-fitting. As a result, the overall performance of the proposed model has been significantly improved which has verified its effectiveness. Additionally, in our own implementation, we could not achieve the reported performance on CUB, which still verifies that the proposed module has no negative effects on fine-grained data.

Embedding Result

In the target mapping space, the correlations between the focus ratios and positions of prototypes are concerned due to the proposed self-focus mechanism. We select the dimensions with the large difference to compare the mapping results in Fig. 3.8, which visually indicated that the self-focus mechanism makes the mapping position biased compared with the baseline.

3.3.4 Alleviation of Class-Level Over-Fitting

Class-level over-fitting (CO) for zero-shot learning is defined as the phenomenon that the model is inclined to select the seen class leading to a poor generalization of unseen classes [51]. As shown in Fig. 3.9, for the embedding based zero-shot learning models,

[1] Available at https://github.com/lzrobots/DeepEmbeddingModel_ZSL

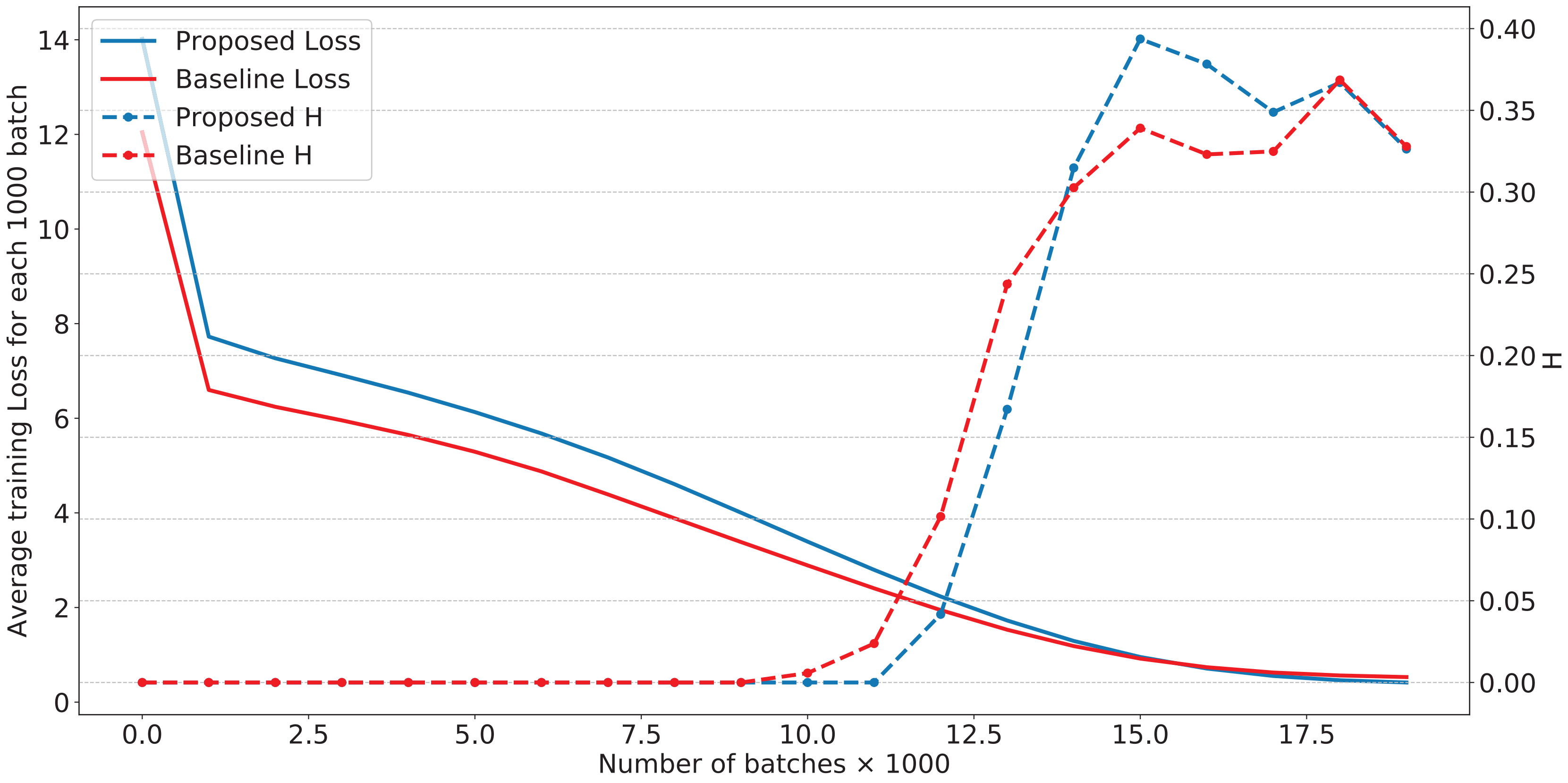


Fig. 3.4: Evaluation of the efficiency during training on coarse-grained datasets APY.

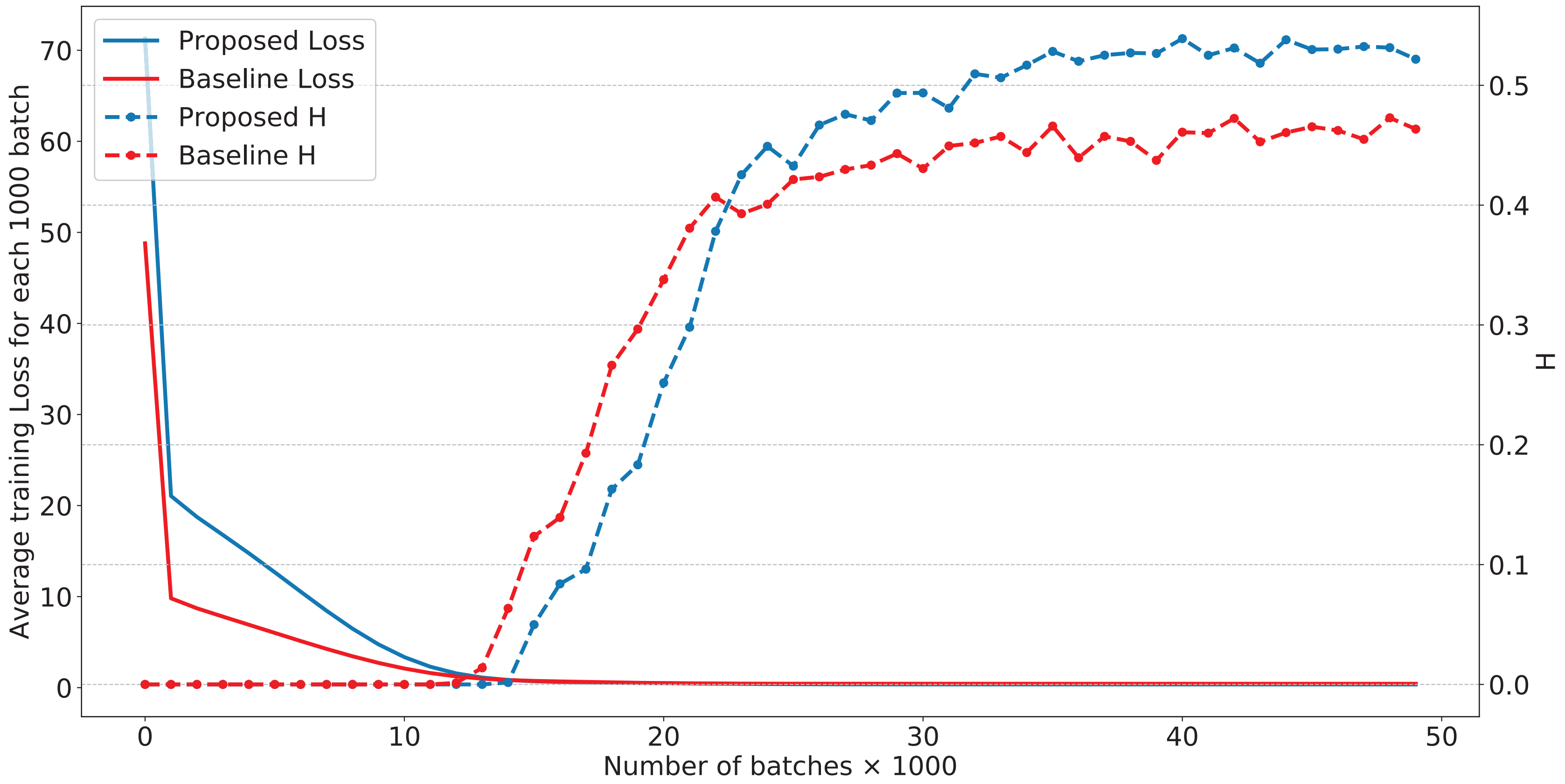
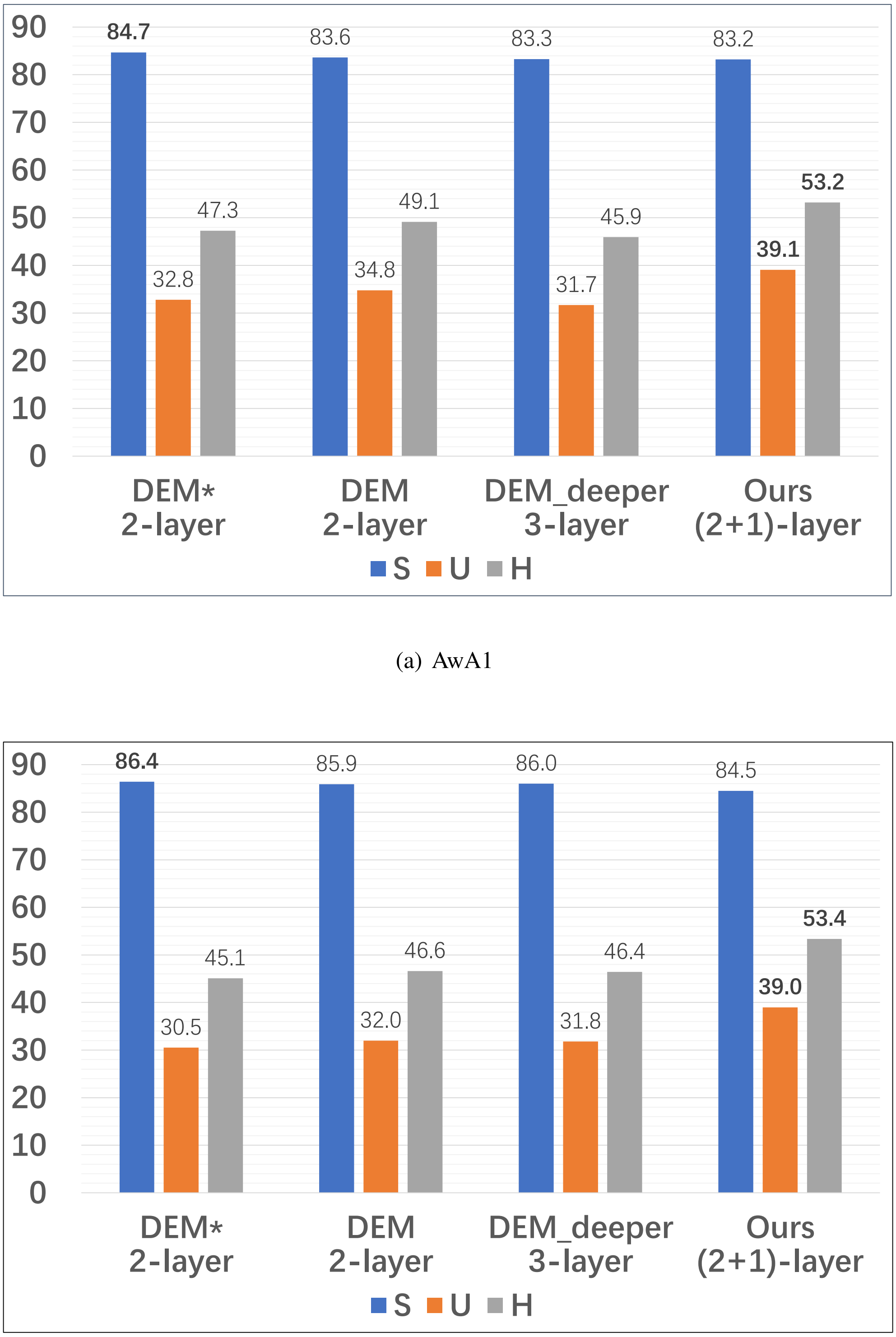


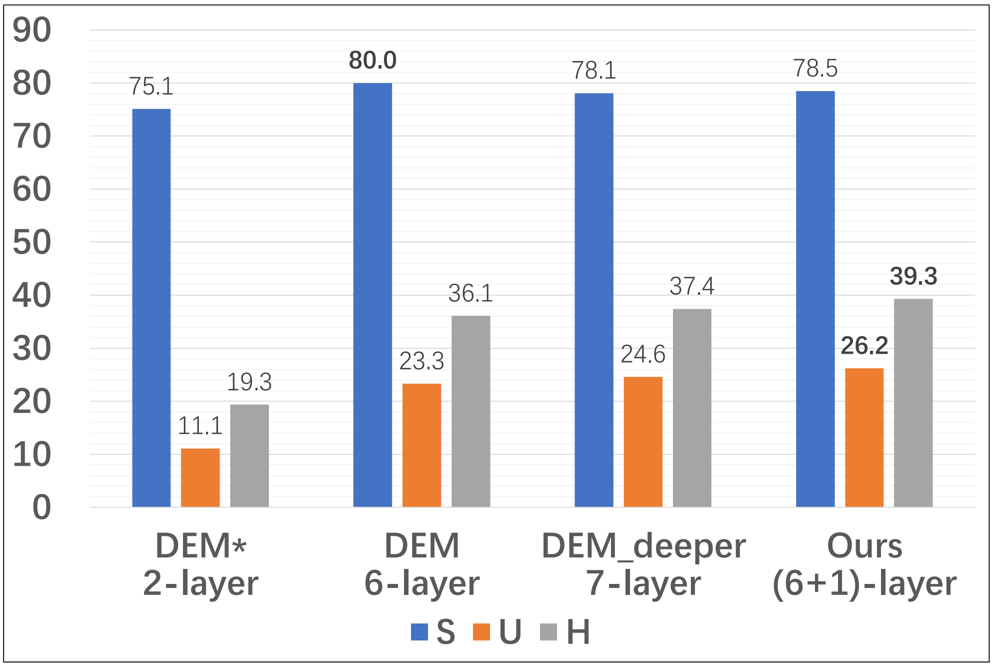
Fig. 3.5: Evaluation of the efficiency during training on coarse-grained datasets AwA2.



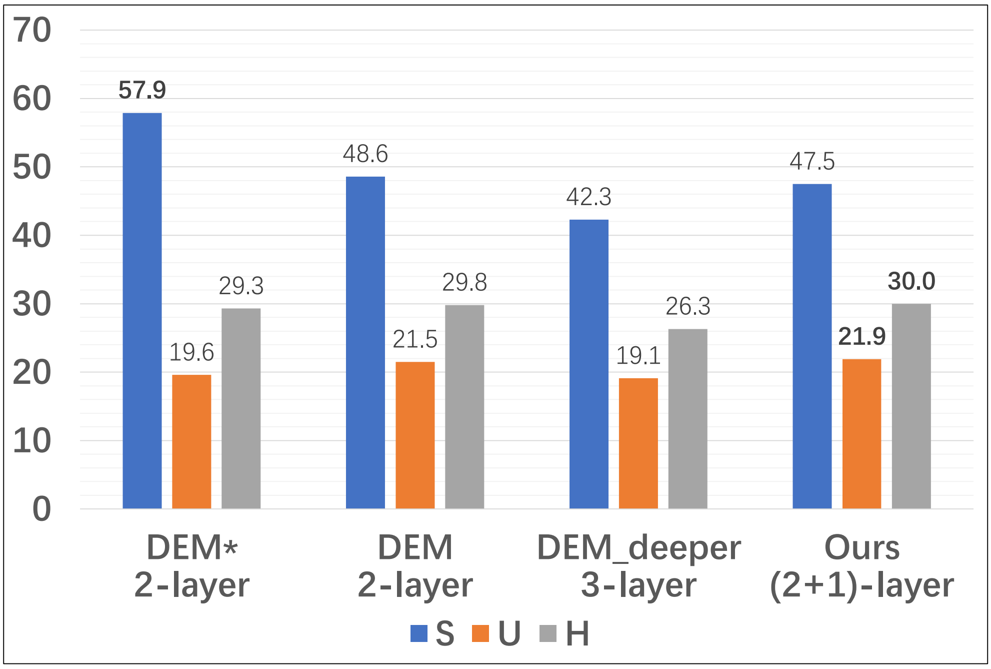
(a) AwA1

(b) AwA2

Fig. 3.6: Comparison between the proposed model and the DEM with a deeper structure on AWA1 and AWA2.

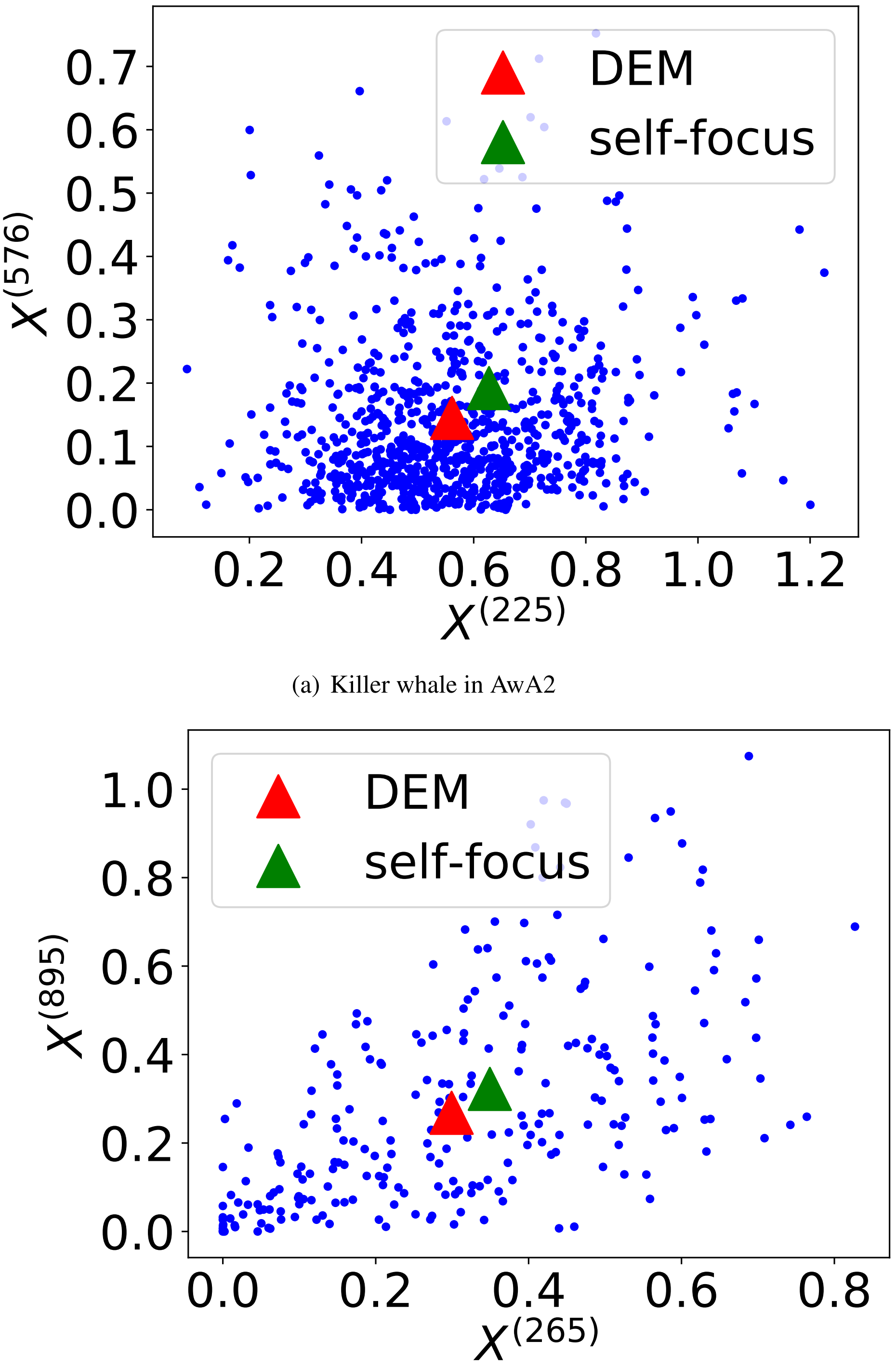


(a) aPY



(b) CUB

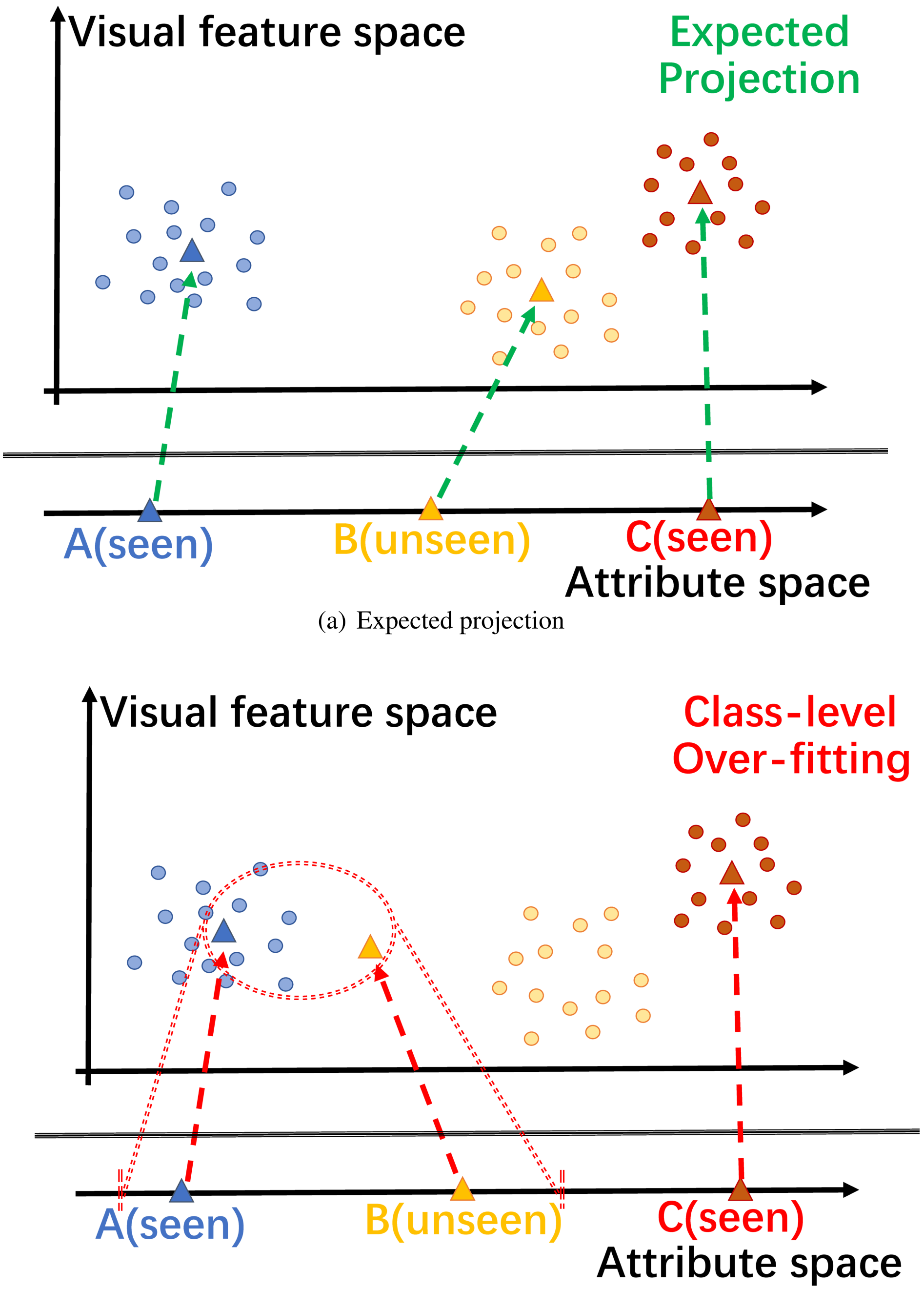
Fig. 3.7: Comparison between the proposed model and the DEM with a deeper structure on aPY and CUB.



(a) Killer whale in AwA2

(b) Bicycle in aPY

Fig. 3.8: 2D plot of chosen dimensions and classes in AwA2 and aPY, where the subscript represents the dimension number.



(a) Expected projection

(b) Class-level over-fitting

Fig. 3.9: The intuitive description of the CO problem for the embedding based zero-shot learning model: (a): The expected projection; (b): The undesirable projection caused by class-level over-fitting.

the CO is mainly expressed as an inappropriately mapping. Due to excessive attention to the training target, the embedding model projects a large area in the source space to a very small area in the target space, thereby affecting the results of the nearest neighbor search.

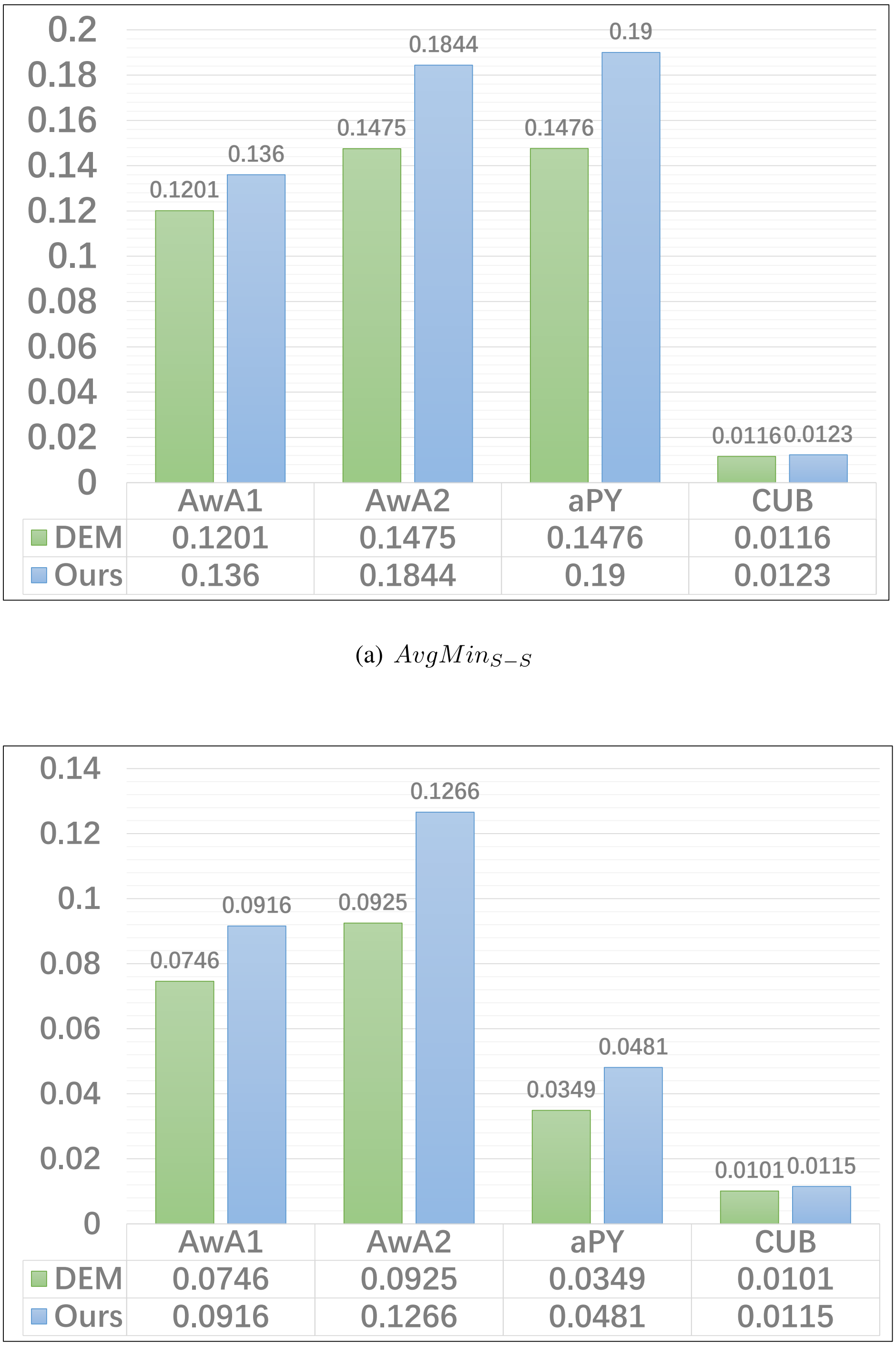
In order to better quantify the CO problem, for each class, we calculate the distance between this class and its nearest seen class. Since the values for some dimensions are always 0 after mapping, we exclude these dimensions when calculating the distance to make it more representative. However, the difference in dimensionality will lead to imprecise measurement. Instead of using the Euclidean distance between two points, we utilize the mean of squared differences in all the dimensions as the distance measure. Then the average values of these distances for all the seen and unseen classes can be separately defined as two metrics $AvgMin_{S-S}$ and $AvgMin_{U-S}$. It should be noted here that the inter-class distances between unseen and seen classes can be used to measure the CO problem, where the lower the values are, the more severe the CO is. While there is no absolute relationship between the inter-class distances of seen classes and the CO problem. However, as a demonstration of the embedding results, we computed both of them.

As the evaluation shown in Fig.3.10, in the coarse-grained zero-shot learning tasks, employing the proposed self-focus mechanism significantly increases the average minimum inter-class distances for both seen and unseen classes. For fine-grained databases, the proposed method has no significant improvement and the reason has been analyzed in the previous section.

3.4 Summary and Discussion

Through learning the relationship between the location and dimensional importance in the target embedding space, a self-focus mechanism is designed for the coarse-grained zero-shot learning tasks to introduce the focus ratio of each dimension to the training process. Since these focus ratios lead the embedding model to focus on essential dimensions, they effectively alleviate the over-fitting of the embedding model suffered during training. In other words, the over-fitting information is apportioned by the proposed self-focus module. As the proposed module does not participate in the identification process, the embedding model becomes more generalized for the unseen classes in zero-shot learning. Extensive analysis and experimental results have validated the effectiveness and superiority of the proposed method in coarse-grained zero-shot learning.

The self-focus mechanism discussed in this chapter is designed for those embedding methods with pairwise optimization loss constructed by a mean squared error on the embedding space. Though it could alleviate the class-level over-fitting via learning a more general embedding function, such architecture can not be applied to those methods with



	AwA1	AwA2	aPY	CUB
🟩 DEM	0.1201	0.1475	0.1476	0.0116
🟦 Ours	0.136	0.1844	0.19	0.0123

(a) $AvgMin_{S-S}$

	AwA1	AwA2	aPY	CUB
🟩 DEM	0.0746	0.0925	0.0349	0.0101
🟦 Ours	0.0916	0.1266	0.0481	0.0115

(b) $AvgMin_{U-S}$

Fig. 3.10: Evaluation of class-level over-fitting problem of the proposed model compared with the baseline DEM.

learnable metrics. However, comparing the experimental results, the learnable metrics such as the learned similarity in relation network [14] also achieve significant performance in zero-shot learning. This kind of method also suffers the class-level over-fitting problem that leads to poor performance on unseen classes in the generalized zero-shot learning task. Therefore, we proposed two adversarial frameworks for relation network based methods in the following chapters.

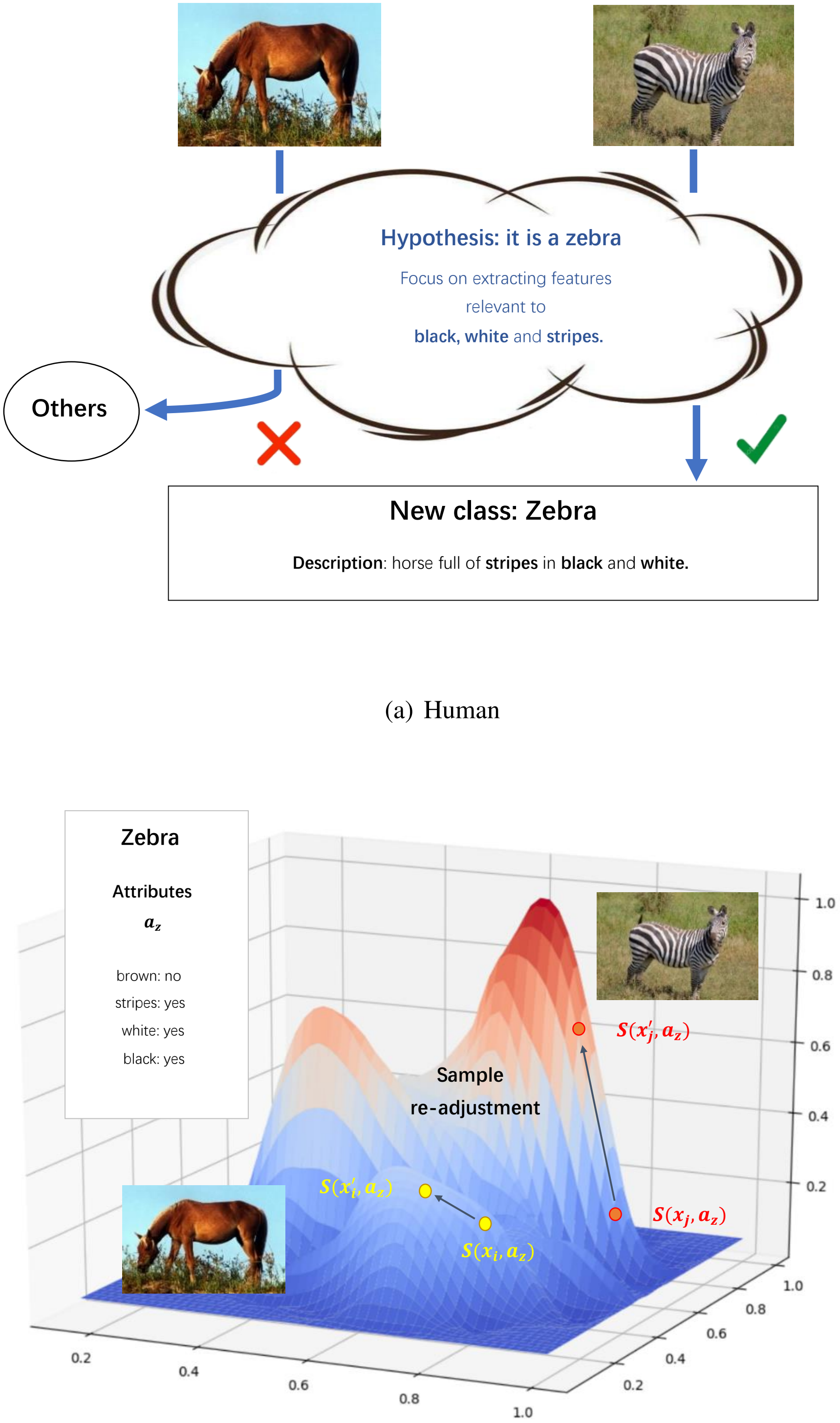
Chapter 4

Adversarial Relation Network for Generalized Zero-shot Learning

The notion that humans can construct the concepts of categories through language descriptions without any visual information brings good insight into zero-shot learning. Most methods try to recognize the unknown categories (unseen classes) by transferring the learned knowledge of source categories (seen classes) through the side information. Currently, zero-shot image classification is the most common zero-shot learning task where semantic attributes and word vectors are widely used as the side information [33, 34, 53, 121]. Moreover, the most stringent and practical zero-shot learning task is defined as inductive generalized zero-shot learning where all information about unseen classes is unavailable during training, and the generalized task requires that the targets contain both seen and unseen categories [52].

Researchers have proposed various zero-shot methods, most of which can be summarized as generative methods [6–8] and embedding methods [13, 51, 56, 133]. However, both types of these methods have drawbacks in handling the inductive generalized zero-shot learning task. The generative methods need to utilize the attributes of the target classes during training, which actually breaks the principle of 'unknown'; on the other hand, the embedding methods are commonly not discriminative enough and tend to classify target samples as seen classes.

In this chapter, we propose an adversarial framework called adversarial Relation Network (advRN) based on an embedding method for the inductive generalized zero-shot learning scenario. Inspired by the human recognition process [134] revealing that the brain will make a hypothesis about incoming data based on previous knowledge, we design a sample re-adjustment (SR) process during the test to enhance the awareness of the unseen classes attributes. In this process, the adversarial noises [135] are reversely exploited to support the hypothesis that samples are from unseen classes. Namely, a gradient-based worst-case perturbation is subtracted from the visual features in order to



(a) Human

(b) Proposed method

Fig. 4.1: Processes of the new class recognition: (a) for human, (b) for the proposed method. Humans make hypotheses based on previous knowledge and then make the judgement. Similarly, the proposed method re-adjusts the instances based on descriptive attributes before recognition.

encourage high responses of the unseen classes. On the other hand, starting from the adversarial training method, we further show that a robust classifier can be built with a gradient penalty (GP) regularization that proves insensitive to small perturbations on samples of seen classes. Overall, when the SR and GP are both applied, the resulting advRN enjoys the appealing property that a small gradient adjustment on the target samples will not affect the classification of seen classes too much but substantially increase the classification accuracy on unseen classes.

The schematic diagram of the recognition process is illustrated in Fig. 4.1. Given an instance x, the proposed method will adjust x to better fit the unseen class attributes. Applying the strategies of both SR and GP, the proposed model has the appealing feature that small sample adjustment imposed by our model can lead to a high response to unseen classes while not affecting the recognition of seen classes due to the robust adversarial training. Namely, compared with the adjusted instance from seen classes, such as horse, only the true zebra instance x'_j will gain after adjustment a high confidence $S(x'_j, a_z)$ due to the robust nature of the recognizer (incurred by the gradient penalty). Detailed justification can be seen in Section 4.2.

In summary, the proposed GP and SR actually form an adversarial framework which makes the model sensitive enough to the unseen classes due to the true unseen hypotheses, while being simultaneously sufficiently robust to avoid misleading by the false unseen hypotheses. The main contributions of this chapter can be listed in the following:

1. A gradient penalty, derived from the adversarial training and constraining the derivatives of the relation scores w.r.t input samples, is constructed as a regularization term during training to strengthen the robustness of the recognizer for seen classes.

2. A sample re-adjustment is designed based on the gradient during the test to make the recognizer more inclined to the unseen classes.

3. To the best of our knowledge, this is the first work using the framework of adversarial examples to obtain significant improvement on inductive generalized zero-shot learning. The proposed adversarial relation network achieves state-of-the-art performance and is competitive compared to those methods taking advantage of unseen class attributes during training.

Since our approach is equivalent to the reverse application of perturbations in the adversarial samples, the entire framework is quite similar to a combination of adversarial attacks and defence. In the next section, we will first introduce the concepts related to robust adversarial training for a better understanding of the proposed framework.

4.1 Adversarial Sample

As the deep learning model has defeated human performance in tasks such as image classification [54], speech recognition [136] and reading lips [137], adopting deep learning techniques in real-world scenarios has made the security of learning algorithms increasingly important. However, it is revealed that the DNNs are commonly vulnerable to the adversarial perturbations [135, 138]. Thus, training a model robust enough to these well-designed perturbations becomes an active and challenging research topic, especially in some security-critic fields such as medical diagnosis and autonomous driving [139].

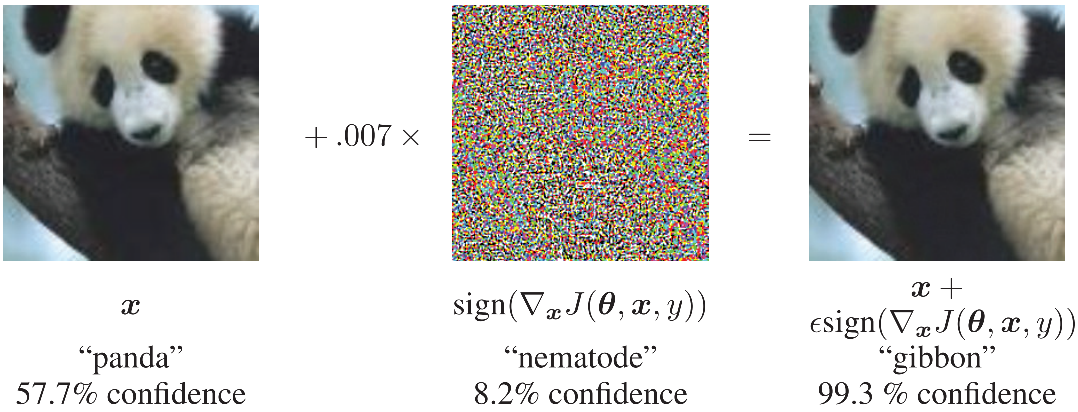

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Fig. 4.2: An example of adversarial sample leading to misclassification of a panda as a gibbon.

The adversarial perturbations are well-crafted and imperceptible but can mislead the model to return unexpected results. Fig 4.2 illustrate an example of such perturbation. The original panda picture is classified correctly by the trained DNN based model with a confidence of 57.7%. By adding a certain adversarial perturbation to the original image, the modified image becomes an adversarial sample that the human can hardly distinguish from the original one, but the trained model will misclassify it as $gibbon$ with confidence 99.3%.

Defining an original sample, also called a clean sample, and its corresponding label as x and y, respectively. Then we can define the adversarial example of x as x'. The difference between x and x' is denoted as $d(x, x')$ where l_1, l_2,...,l_{inf} norm are commonly used for measuring the difference and it is usually constrained by a small value $d(x, x') \leqslant \epsilon$. With a trained DNN based classifier, one can seek the adversarial example by constructing x' satisfying:

$$d(x, x') \leqslant \epsilon, \quad \text{such that} \quad class(x') \neq class(x). \tag{4.1}$$

where the $class$ denotes the prediction made by the trained classifier. Basically for a more tractable way, one can obtain the adversarial sample by solving the optimization problem as follows:

$$x' = \underset{d(x_{adv},x)\leqslant\epsilon}{\operatorname{argmax}}\ L(x_{adv}, y), \tag{4.2}$$

where the $L(\cdot,\cdot)$ represents the loss function for training the classifier. With the calculated adversarial sample, one can mislead the classifier to return an undesired result so-called adversarial attack. The most straightforward adversarial defence strategy is to include the adversarial samples into the training set to promote the robustness of the model.

4.2 Methodology

In this section, we will first define the zero-shot learning task formally and then detail the proposed adversarial relation network.

4.2.1 Problem Definition

In this chapter, we still focus on the most challenging scenario, i.e. inductive generalized zero-shot learning, where the specific information of the targets is always limited till the test and the models are required not only to recognize the unseen classes but also to be discriminative enough between the seen and unseen classes.

Following the idea of Relation Network (RN) [14], the classification is performed by comparing the relation scores of the sample and the semantic attributes for all the classes. Denote $\boldsymbol{X} = \{x_1, ..., x_N\}$, $\boldsymbol{Y} = \{y_1, ..., y_N\}$, $\boldsymbol{A} = \{a_1, ..., a_K\}$ as visual features, class labels, and semantic attributes sets, respectively, with the total sample number N and class number K. We define $N = N_{tr}^S + N_{te}^S + N^U$ and $K = K^S + K^U$ to divide the whole dataset into a source set (seen set) $\boldsymbol{X}_{tr} \times \boldsymbol{Y}_{tr} \times \boldsymbol{A}_{tr} = \{(x_i, y_i, a_{y_i}), i = 1, 2, ..., N_{tr}^S\}$ and a target set (unseen set) $\boldsymbol{X}_{te} \times \boldsymbol{Y}_{te} \times \boldsymbol{A}_{te} = \{(x_m, y_m, a_{y_m}), m = N' + 1, ..., N\}$ constructed in terms of 3-tuple. Here $\boldsymbol{X}_{tr} \cap \boldsymbol{X}_{te} = \emptyset$ and $\forall y_i \in \boldsymbol{Y}_{tr} : y_i \leq K^S$ are required in the inductive scenario. With a trained relation score function $S(x, a)$ based on the source set, the zero-shot learning target is to achieve the classification by satisfying $S(x_m, a_{y_m}) > S(x_m, a_k)$ for all $k \neq y_m$. Specifically, $0 < k \leq K$, $N' = N_{tr}^S$ for generalized zero-shot learning, and $K^S < k \leq K$, $N' = N_{tr}^S + N_{te}^S$ for conventional zero-shot learning.

4.2.2 Adversarial Relation Network

Relation Network [14]. In RN, the visual features $\boldsymbol{X}$ are extracted as training samples by a pre-trained DNN. A learned embedding function $f_\theta(\cdot)$ will project the semantic attribute vectors into the embedded space (visual feature space) as the prototypes of the corresponding classes. After a feature and a prototype are concatenated, their relation score will be calculated via a learned metric function $g_\phi(\cdot)$. By calculating the mean squared error (MSE) between the one-hot labels and the relation scores, a loss function for optimizing parameters θ and ϕ is designed as follows:

$$L_R(\boldsymbol{X}_{tr}, \boldsymbol{Y}_{tr}, \boldsymbol{A}_{tr}; \theta, \phi) = \frac{1}{N_{tr}^S} \frac{1}{K^S} \sum_{i=1}^{N_{tr}^S} \sum_{k=1}^{K^S} [g_\phi(x_i \oplus f_\theta(a_k)) - r(x_i, a_k)]^2, \tag{4.3}$$

$$r(x_i, a_k) = \begin{cases} 0 & k \neq y_i, \\ 1 & k = y_i, \end{cases} \tag{4.4}$$

where $\oplus$ represents the concatenation operation and $r(x_i, a_k)$ indicates whether x_i and a_k represent the same class. The $g_\phi(x_i \oplus f_\theta(a_k))$ can be regarded as the required relation score function $S(x_i, a_k)$.

AdvRN with Gradient Penalty. In order to achieve a robust perception of seen classes, we design a robust training process encouraging that any small perturbation on instances of seen classes would not affect the network's output:

$$\min_{\theta,\phi} \max_{\boldsymbol{\epsilon}:\|\boldsymbol{\epsilon}\|_p \leq \sigma} L_R(\boldsymbol{X}_{tr} + \boldsymbol{\epsilon}, \boldsymbol{Y}_{tr}, \boldsymbol{A}_{tr}; \theta, \phi), \tag{4.5}$$

where the robustness against any small perturbation defined as $\boldsymbol{\epsilon}$ is required in the inner problem.

Since the minimax problem is usually difficult to solve, the first-order Taylor expansion at feature point x can be used to approximately solve this non-convex problem. The inner problem can then be relaxed as follows:

$$\max_{\boldsymbol{\epsilon}} L_R(\boldsymbol{X}_{tr}, \boldsymbol{Y}_{tr}, \boldsymbol{A}_{tr}; \theta, \phi) + \nabla_{\boldsymbol{X}_{tr}} {L_R}^{\mathsf{T}} \boldsymbol{\epsilon} \qquad \text{s.t.} \qquad \|\boldsymbol{\epsilon}\|_p \leq \sigma. \tag{4.6}$$

To solve the optimization problem, we first introduce Lemma 4.2.2 from [140]. The optimization problem $\max_{\boldsymbol{\epsilon}} \nabla_{\boldsymbol{X}_{tr}} {L_R}^{\mathsf{T}} \boldsymbol{\epsilon}$ s.t. $\|\boldsymbol{\epsilon}\|_p \leq \sigma$ has a closed form solution $\boldsymbol{\epsilon} = \sigma \ \mathrm{sign}(\nabla L_R)(\frac{|\nabla L_R|}{\|\nabla L_R\|_{p^*}})^{\frac{1}{p-1}}$ where p^* is the dual of p, i.e., $\frac{1}{p^*} + \frac{1}{p} = 1$.

Since L_R is independent of $\boldsymbol{\epsilon}$, the optimal $\boldsymbol{\epsilon}$ should have a norm of σ. Then the maximum problem can be solved by Lagrangian multiplier method with defined $f(\boldsymbol{\epsilon}) \equiv \nabla_{\boldsymbol{X}_{tr}} {L_R}^{\mathsf{T}} \boldsymbol{\epsilon}$ and $g(\boldsymbol{\epsilon}) \equiv \|\boldsymbol{\epsilon}\|_p = \sigma$. Set $\nabla f(\boldsymbol{\epsilon}) = \lambda \nabla g(\boldsymbol{\epsilon})$, we have

$$\nabla f(\boldsymbol{\epsilon}) = \lambda \nabla g(\boldsymbol{\epsilon}) \tag{4.7}$$

$$\nabla_{\boldsymbol{X}_{tr}} L_R = \lambda \frac{\boldsymbol{\epsilon}^{p-1}}{p(\sum_i \boldsymbol{\epsilon}_i^p)^{1-\frac{1}{p}}} \tag{4.8}$$

$$\nabla_{\boldsymbol{X}_{tr}} L_R = \frac{\lambda}{p} (\frac{\boldsymbol{\epsilon}}{\sigma})^{p-1} \tag{4.9}$$

$$\nabla_{\boldsymbol{X}_{tr}} {L_R}^{\frac{p}{p-1}} = (\frac{\lambda}{p})^{\frac{p}{p-1}} (\frac{\boldsymbol{\epsilon}}{\sigma})^p \tag{4.10}$$

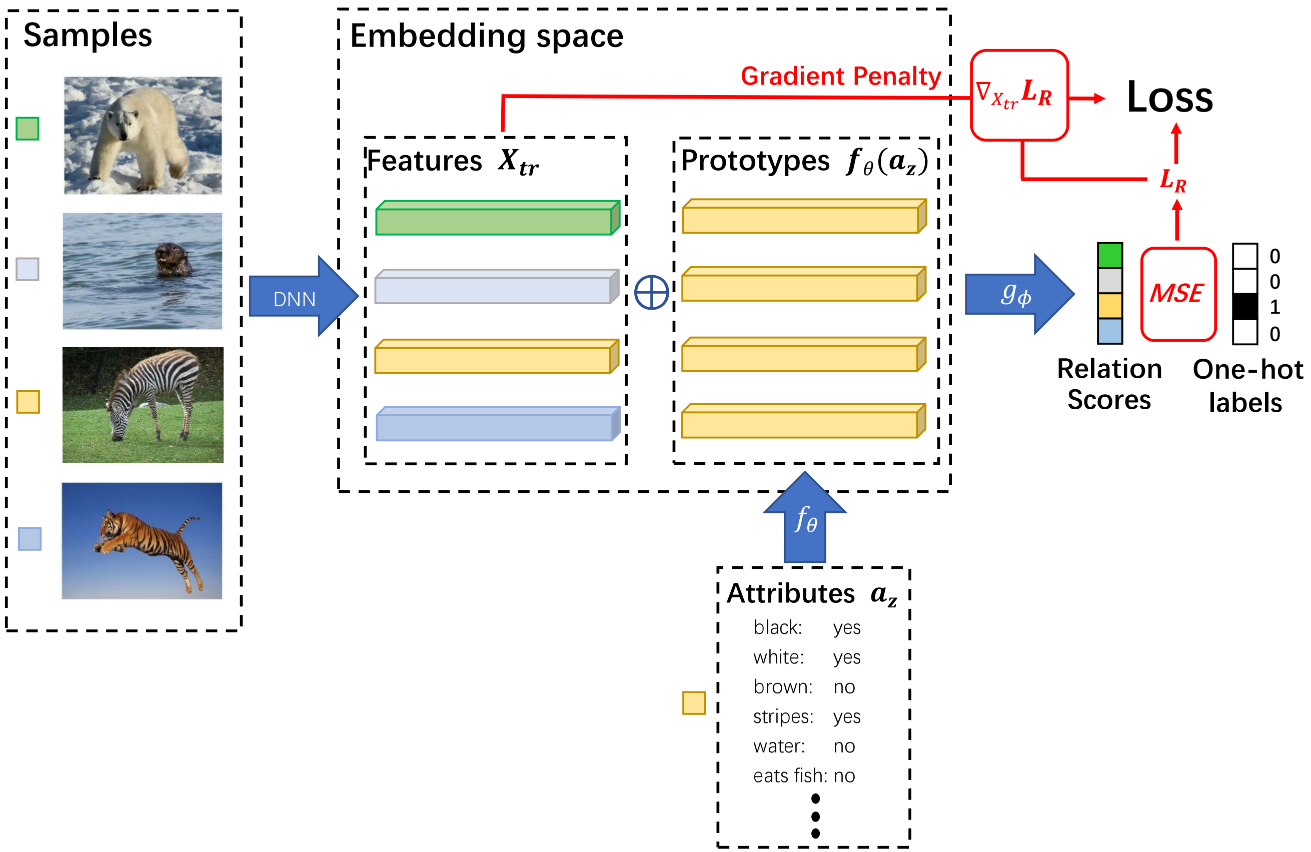


Fig. 4.3: Flowchart of the training process in advRN. As an example, the semantic attribute of Zebra is embedded into the visual feature space and copied several times used to calculate the relation scores. With these relation scores, the loss function for RN and the further defined gradient penalty can be obtained. The DNN denotes the pre-trained feature extractor, $\oplus$ represents the concatenating operations, f_θ and g_ϕ are the embedded and metric function to be learned, respectively.

By summing over two sides, we get

$$\sum \nabla_{\boldsymbol{X}_{tr}} L_R{}^{\frac{p}{p-1}} = \sum (\frac{\lambda}{p})^{\frac{p}{p-1}} (\frac{\boldsymbol{\epsilon}}{\sigma})^p \tag{4.11}$$

$$\|\nabla L_R\|_{p^*}^{p^*} = (\frac{\lambda}{p})^{p^*} * 1 \tag{4.12}$$

$$(\frac{\lambda}{p}) = \|\nabla L_R\|_{p^*} \tag{4.13}$$

By combining Equation (4.9) and (4.13), we can easily obtain

$$\boldsymbol{\epsilon} = \sigma \ \mathrm{sign}(\nabla L_R)(\frac{|\nabla L_R|}{\|\nabla L_R\|_{p^*}})^{\frac{1}{p-1}} \tag{4.14}$$

This completes the proof. □

Using Lemma 4.2.2, we can obtain a closed form solution of problem (4.6). Then, with this solution and a settled $p = 2$, the approximation of the minmax problem becomes

$$\min_{\theta,\phi} L_R(\boldsymbol{X}_{tr}, \boldsymbol{Y}_{tr}, \boldsymbol{A}_{tr}; \theta, \phi) + \sigma\|\nabla L_R\|_2. \tag{4.15}$$

Here, instead of only minimizing the loss function (4.3), we additionally minimize a gradient penalty $\sigma\|\nabla L_R\|_2$. Adding this gradient penalty into the loss as a regularization term is approximately equivalent to injecting adversarial noises into the samples during training, as shown above. As a result, the trained model will be less sensitive to any small changes in samples for seen classes.

In practice, to make the model more resistant to over-fitting, the regularization term $||\theta||^2$ can be usually applied. Consequently, the final optimization loss function for training the advRN can be summarized as below:

$$L_{total} = L_R + \sigma\|\nabla L_R\|_2 + \alpha\,||\theta||^2, \tag{4.16}$$

where σ and α are the trade-off parameters.

For illustration, we also plot Fig. 4.3 which briefly shows the proposed training process.

4.2.3 Sample Re-adjustment with Gradient Guidance

As an imitation of the hypothesis testing cognition, we design our test process in the following. The test process consists of three parts: gradient calculation with the unseen classes hypothesis, sample re-adjustment, and classification as shown in Fig. 4.4.

Similar to hypothesis testing cognition, we first make an unseen-against-seen hypothesis that each test instance x_m belongs to unseen categories. Then we can construct an

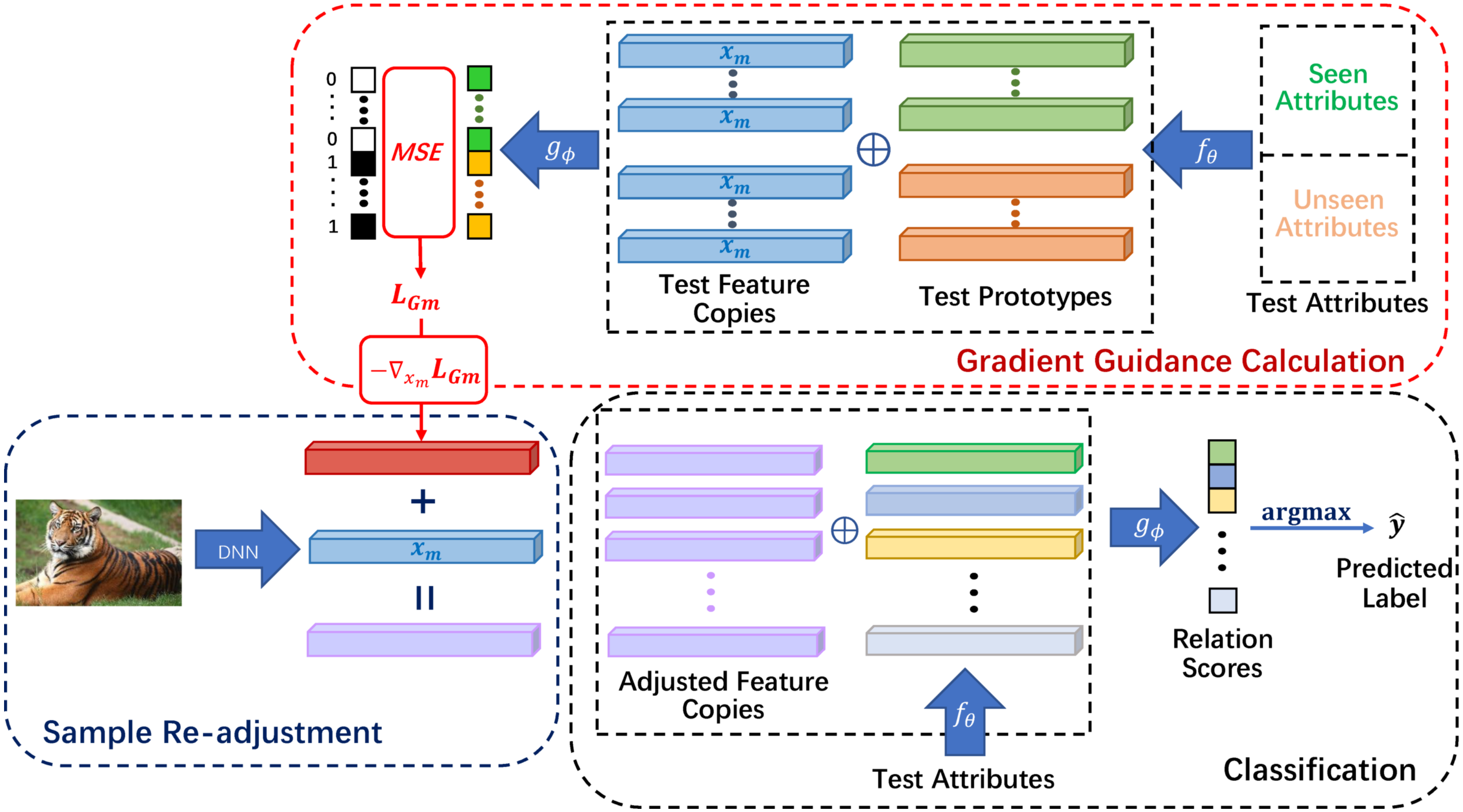


Fig. 4.4: Flowchart of the proposed test process. A gradient is calculated which makes the model tend to recognize the feature as an unseen class; then, the instance feature is re-adjusted with the gradient. The DNN denotes the pre-trained feature extractor, $\oplus$ represents the concatenating operations, f_θ and g_ϕ are the need to be learned embedded and metric functions, respectively.

objective function L_{Gm} for this hypothesis as follows:

$$L_{Gm} = \frac{1}{K}\sum_{k=1}^{K}(g_\phi(x_m \oplus f_\theta(a_k)) - l(k))^2, \tag{4.17}$$

$$l(k) = \begin{cases} 0 & 0 < k \leqslant K^S, \\ 1 & K^S < k \leqslant K, \end{cases} \tag{4.18}$$

where $l(k)$ indicates whether the k-th class is an unseen one. This objective function actually measures the total cost with the hypothesis that each test sample was from unseen categories.

Inspired by the adversarial idea as discussed in Section 4.2.2 that gradient-based perturbations on samples tend to deceive the classification, we introduce a similar gradient but adjust the instances in the opposite direction. Namely, the gradient of the loss function w.r.t. the input instance $\nabla_{x_m} L_{Gm}$ presents the worst-case perturbation which would change the output of the classifier. On the contrary, adjusting the sample in a reversed gradient direction would then lead to more confidence in the original unseen-against-seen hypothesis.

As a result, the adjusted features become:

$$x'_m = x_m - \lambda \nabla_{x_m} L_{Gm}, \tag{4.19}$$

where λ denotes the adjustment rate. After the feature instance is adjusted, we can directly predict its label as the category with the maximum relation score between the adjusted feature and the semantic attributes:

$$\hat{y}_m = \arg\max_{k \in K} \, g_\phi(x'_m \oplus f_\theta(a_k)) \tag{4.20}$$

In a short summary, by applying the SR, the model is adapted so that a high response can be obtained for small adversarial perturbations to unseen classes, consequently enhancing the awareness of the unseen classes. On the other hand, robust training with GP has the property that any small perturbations on the samples of seen classes would not change the network's output (as discussed in the previous subsection). This would greatly alleviate the limitation of previous inductive GSZL methods, i.e., they are commonly not discriminative enough and tend to classify target samples as seen classes. The detailed steps for the whole framework are presented in Algorithm 1.

Remarks on Entire Adversarial Framework.

The training and test processes form an entire adversarial framework for the inductive generalized zero-shot learning. Adding the gradient penalty during training can be deemed

Algorithm 1: Procedure of the proposed training & testing

```
Training:
Input: dataset $\boldsymbol{X}_{tr}$ and $\boldsymbol{Y}_{tr}$with the total sample number $N_{tr}^{S}$ and
 $A = \{a_1, ..., a_{K_S}\}$ with class number $K^S$, learning rate $\gamma$, regularization weight
 $\sigma$, $\alpha$ and batch size $N_b$
Parameter: $\theta$ and $\phi$
for random batch $\{x_i\}_{i=1}^{N_b} \sim X_{tr}$, $\{y_i\}_{i=1}^{N_b} \sim Y_{tr}$ with $K_\tau$ classes
    Compute the $L_R(\phi, \theta)$ through Eq. 4.3
    Calculate the total loss $L_{total}(\phi, \theta)$ for the batch samples through Eq. 4.16
    Optimize $\theta$ and $\phi$ based on the calculated total loss $L_{total}(\phi, \theta)$:
    $\theta \leftarrow \theta - \gamma \cdot \nabla_\theta L_{total}(\phi, \theta)$
    $\phi \leftarrow \phi - \gamma \cdot \nabla_\phi L_{total}(\phi, \theta)$
end for
return $\theta$ and $\phi$
Testing:
Input: Trained embedding function $f_\theta(\cdot)$ and metric function $g_\phi(\cdot)$; seen class
number $K^S$ and unseen class number $K^U = \{K^S + 1, \ldots, K\}$;
$A = \{a_1, ..., a_K\}$; instance $x_m \in \boldsymbol{X}^U$
Calculate the object function $L_{Gm}$ for instance through Eq. 4.17
Perturb the instance as $x_m^{'}$ through Eq. 4.19
Predict the final label $\widehat{y}_m$ of instance $x_m$ with the perturbed instance through Eq.4.20
return predicted label $\widehat{y}_m$
```

Table 4.1: Statistic information for zero-shot learning benchmark in terms of scale, granularity, attribute dimension, class size and sample size.

benchmark	scale	granularity	attribute dimension	class size		sample size		
				train(seen)	unseen	train	test$_{seen}$	test$_{unseen}$
AwA1	medium	coarse	85	27+13	10	19832	4958	5685
AwA2	medium	coarse	85	27+13	10	23527	5882	7913
aPY	small	coarse	64	15+5	12	5932	1483	7924
CUB	medium	fine	312	100+50	50	7057	1764	2967

as an adversarial defence for the seen classes, making the classifier sufficiently robust to small sample perturbations. On the other hand, small sample re-adjustment in the test phase can be regarded as an adversarial attack that tends to mislead the classifier into choosing the unseen classes. In summary, applying the strategies of both SR and GP, the proposed model has the appealing feature that small sample re-adjustment can lead to high response to unseen classes while not affecting the recognition of seen classes due to the robust adversarial training.

4.3 Experiment

4.3.1 Setup

Dataset. To evaluate the performance of the proposed method, we select four commonly used zero-shot learning benchmark datasets: AwA1 [33], AwA2 [53], aPY [121] and CUB [34]. Detailed statistic information for each benchmark is shown in Table 4.1. AwA1 is a medium-scale coarse-grained dataset containing 50 classes of animals with 85 attributes. AwA2 keeps the same setting as AwA1 but contains different samples. aPY is a small-scale coarse-grained dataset with 32 object classes and 64 attributes, and CUB is a medium-scale fine-grained dataset with 200 bird classes described in 312 attributes.

Setting. All quantitative evaluations were conducted in the inductive generalized zero-shot learning scenario. We follow the attributes, features and train/test splits proposed in [53] (GBU setting) to avoid overlapping of categories between the test set and the set for the training feature extractor. The average per-class top-1 accuracy is selected to measure the performance. Specifically, for generalized zero-shot learning, criteria $\mathcal{ACC}_S$ and $\mathcal{ACC}_U$ denote accuracies for the seen and unseen categories, respectively, and H is

Table 4.2: Specific experiment settings for ZSL benchmarks in terms of layer structures, learning rate, batch size, regularization weight, and adjustment rate.

benchmark	Training process						Test process
	Layer structures		Learning rate	Batch size	Regularization weight		Adjustment rate
	f_θ	g_ϕ	γ	N_b	σ	α	λ
AwA1	$85 \times 1600 \times 2048$	$4096 \times 400 \times 1$	1×10^{-5}	32	1.5	1×10^{-5}	10000/32
AwA2	$85 \times 1600 \times 2048$	$4096 \times 400 \times 1$	1×10^{-5}	32	1.5	1×10^{-5}	10000/32
aPY	$65 \times 1200 \times 2048$	$4096 \times 400 \times 1$	1×10^{-4}	32	1.5	1×10^{-5}	5000/32
CUB	$312 \times 1200 \times 2048$	$4096 \times 1200 \times 1$	1×10^{-5}	32	1.5	1×10^{-5}	2500/32

defined as the harmonic mean of $\mathcal{ACC}_S$ and $\mathcal{ACC}_U$ to evaluate the overall performance as follows:

$$H = \frac{2 \times \mathcal{ACC}_S \times \mathcal{ACC}_U}{\mathcal{ACC}_S + \mathcal{ACC}_U}. \tag{4.21}$$

The learned embedding function f_θ is designed as a two-layer neural network with the activation function $relu$. The learned metric function g_ϕ engages a similar setting to that of f_θ, but the activation function of the last layer is replaced by $sigmoid$. The specific values of hyperparameters for each benchmark are shown in Table 4.2 where the adjustment rate λ is roughly chosen from {2500/32, 5000/32, 10000/32}[2] by the validation set.

4.3.2 Results

Since we focus on inductive generalized zero-shot learning tasks, here we only compared the proposed methods with the models designed under inductive scenario. It is observed that the proposed model achieves the most significant performance for unseen accuracy and harmonic mean compared to the state-of-the-art methods as shown in Table 4.3. As the aim of our method is to learn a more generalized recognition model, it is reasonable that the accuracies for seen classes are not the highest. Compared to the baseline RN, while $\mathcal{ACC}_S$ was decreased by 11.8%, 9.4% and 2.8%, $\mathcal{ACC}_U$ has been significantly promoted by 19.2%, 19.3% and 6.8% in AwA1, AwA2 and aPY, respectively. More importantly, overall, the proposed model attains the best harmonic mean in all the four datasets. Taking a closer examination of CUB, we observe an increase in both $\mathcal{ACC}_S$ and $\mathcal{ACC}_U$ over RN, showing that the proposed advRN model obtains a more generalized recognizer. It is noted that, even without using the test information during training, our model leads to comparable performance with most of the semantic transductive generalized zero-shot learning methods.

We first plot in Fig. 4.5 the performance of advRN as the adjustment rate λ increases. As observed, when sample re-adjustment is not applied (i.e., $\lambda = 0$), the gradient penalty improves the accuracy of known or seen classes, but it actually leads to an over-fitting on the seen classes and consequently causes the decline in $\mathcal{ACC}_U$.

4.3.3 Further Analysis

In order to further demonstrate the effectiveness, we take AwA2 as one illustrative example to have a closer examination. However, after combining the sample re-adjustment, the proposed advRN method improves the recognition of unseen classes significantly. A small adjustment (with a small λ) leads to a sharp increase in $\mathcal{ACC}_U$ and a slight decrease in $\mathcal{ACC}_S$; this alleviates the aforementioned over-fitting problem. As we observed

[2] This division by 32 is due to the effect of batch size during the derivation in the test process.

Table 4.3: Comparisons between the proposed advRN and the other state-of-the-art methods. Values are accuracies in %.

Method	AwA1			AwA2			aPY			CUB		
	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H
SAE [56]	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9	7.8	54.0	13.6
PSR-ZSL [60]	–	–	–	20.7	73.8	32.3	13.5	51.4	21.4	24.6	54.3	33.9
DEM [13]	32.8	84.7	47.3	30.5	86.4	45.1	11.1	75.1	19.4	19.6	57.9	29.2
Triple Verification Net [51]	27.0	67.9	38.6	–	–	–	16.1	66.9	25.9	26.5	62.3	37.2
LFGAA+Hybrid [92]	–	–	–	27.0	**93.4**	41.9	–	–	–	36.2	**80.9**	50.0
MIVAE [141]	35.9	87.8	51.0	32.9	90.4	48.2	15.3	**87.4**	26.0	31.5	60.4	41.4
RN [14]	31.4	**91.3**	46.7	30.0	**93.4**	45.3	21.2	68.8	32.4	38.1	61.1	47.0
advRN(ours)	**50.6**	79.5	**61.8**	**49.3**	84.0	**62.2**	**28.0**	66.0	**39.3**	**44.3**	62.6	**51.9**

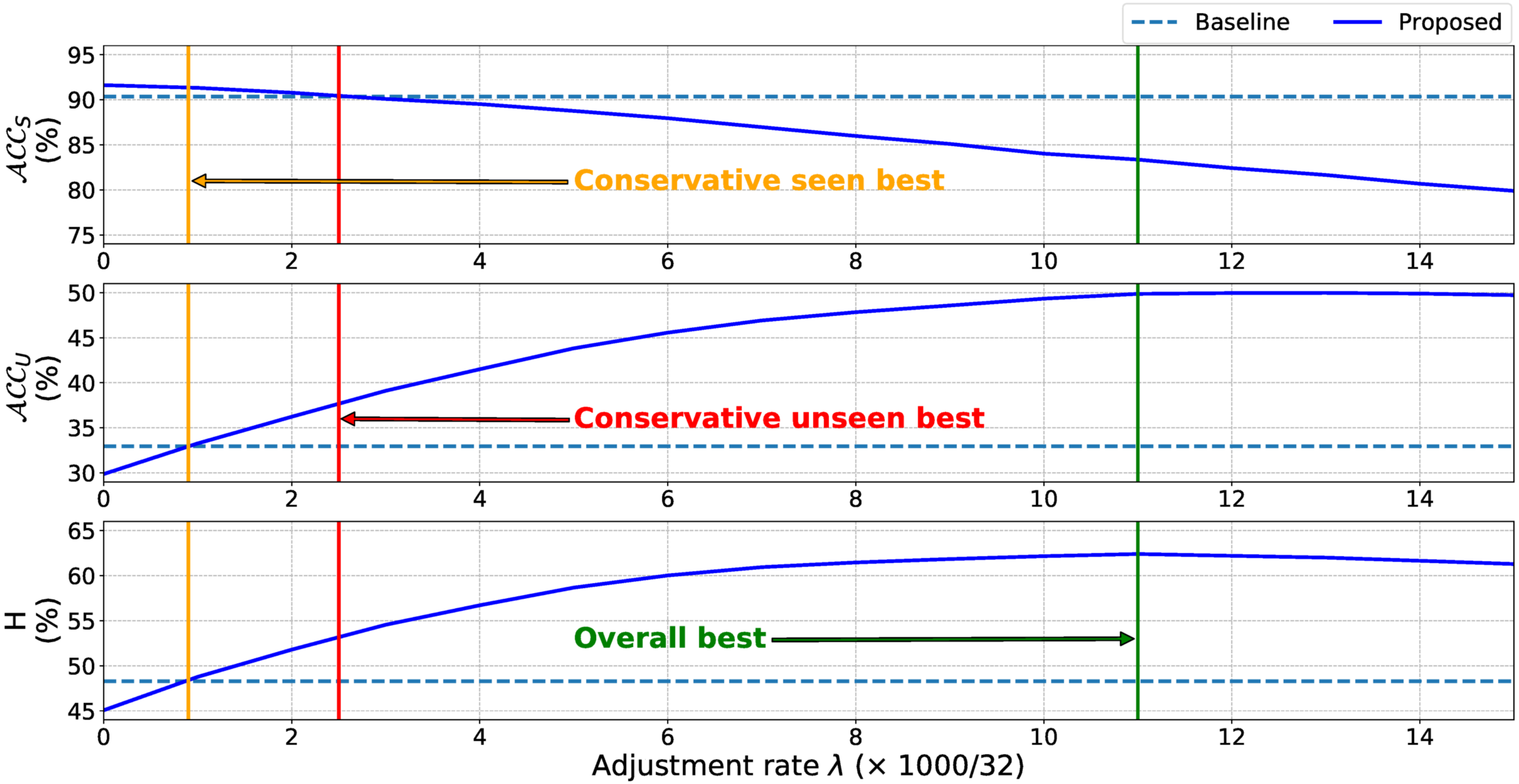


Fig. 4.5: Accuracies ($\mathcal{ACC}_S$, $\mathcal{ACC}_U$ and H) vs. adjustment ratio λ for the proposed method on AwA2. RN without sample re-adjustment is also provide as the baseline.

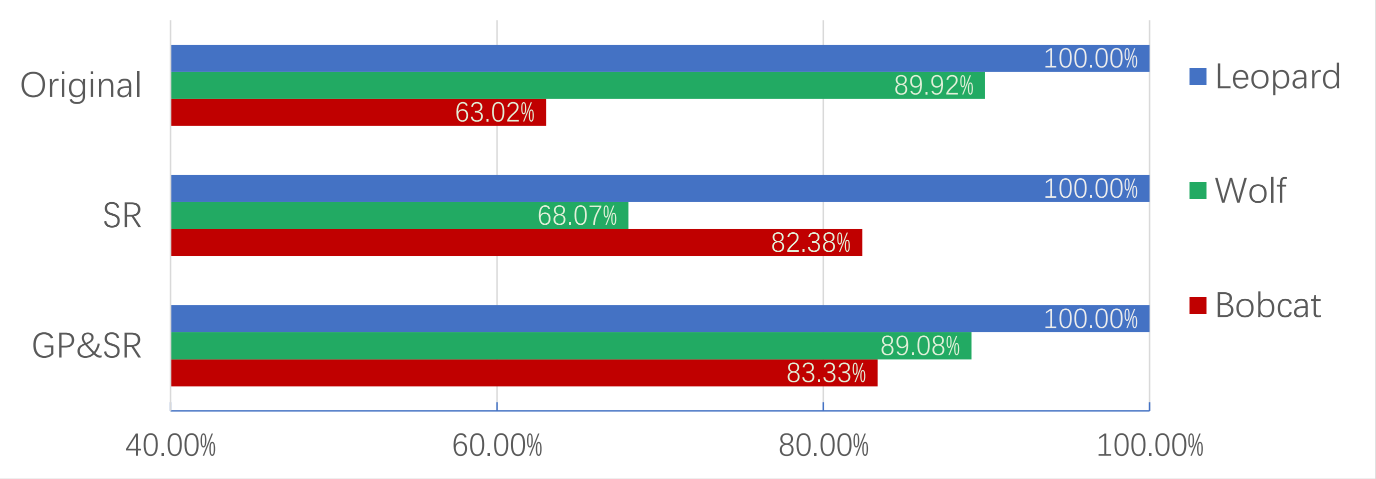


Fig. 4.6: Accuracy histogram for bobcat, wolf and leopard in AwA2.

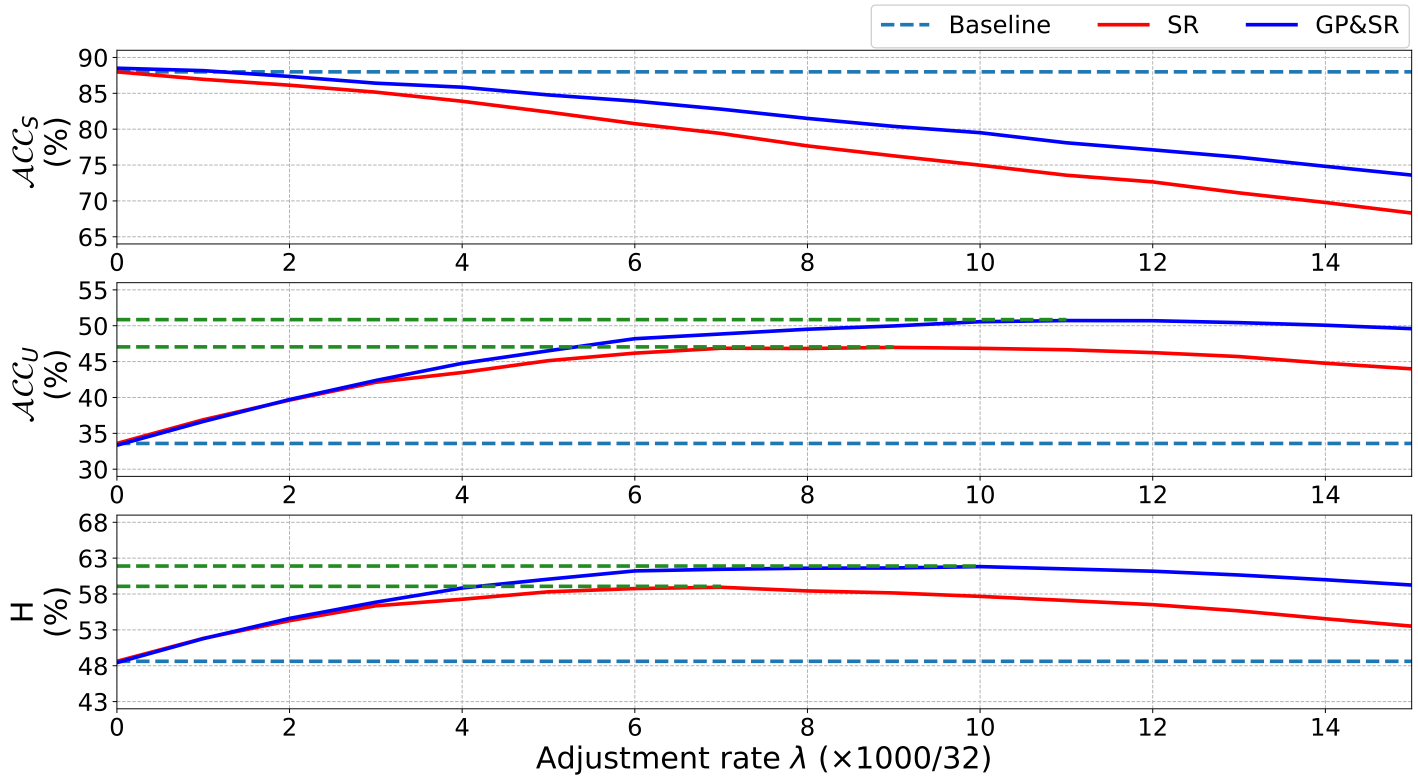


Fig. 4.7: Accuracies ($\mathcal{ACC}_S$, $\mathcal{ACC}_U$ and H) vs. adjustment ratio λ on AwA1. GP largely increases the robustness of the recognizer especially on seen classes. SR means that only sample re-adjustment is applied, GP&SR denotes both gradient penalty and sample re-adjustment are applied and the dashed line represents the baseline performance as reference.

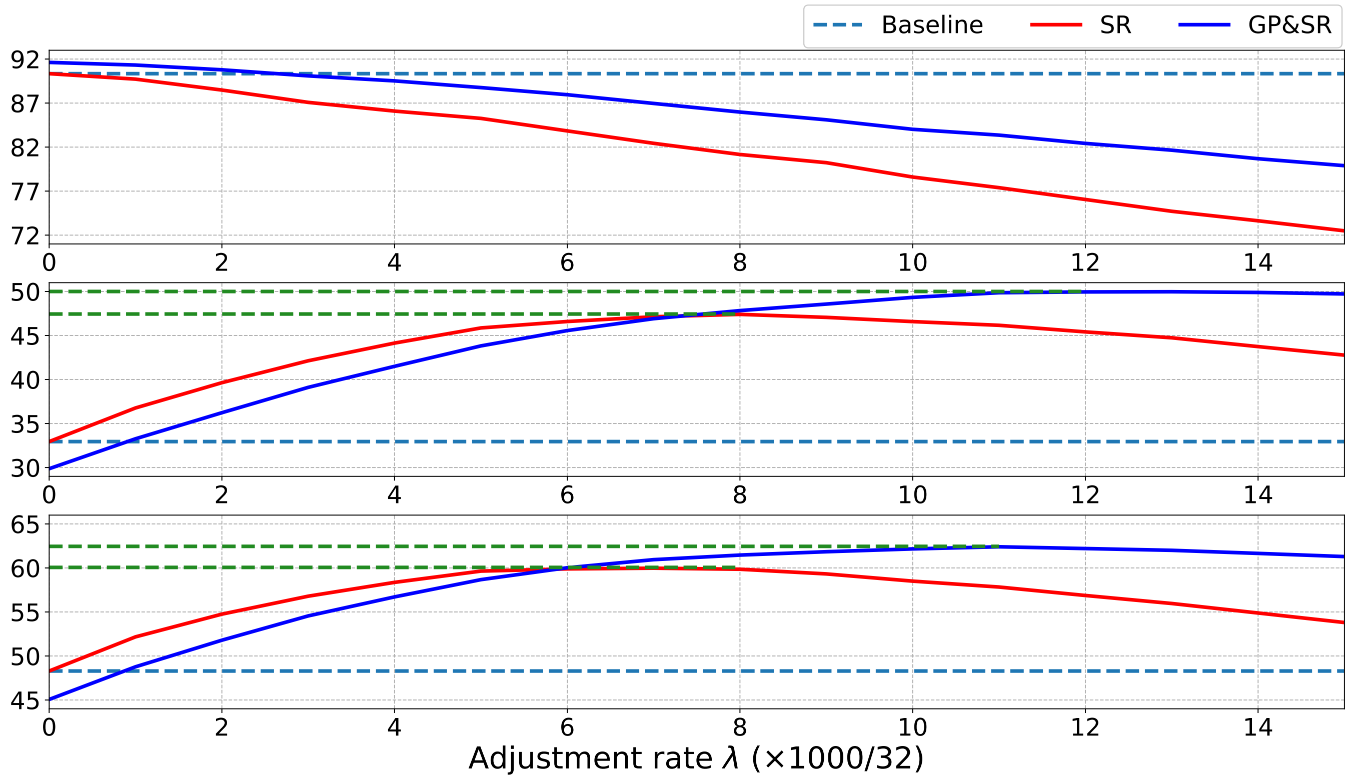


Fig. 4.8: Accuracies ($\mathcal{ACC}_S$, $\mathcal{ACC}_U$ and H) vs. adjustment ratio λ on AwA2. GP largely increases the robustness of the recognizer especially on seen classes. SR means that only sample re-adjustment is applied, GP&SR denotes both gradient penalty and sample re-adjustment are applied and the dashed line represents the baseline performance as reference.

in Fig. 4.5, for the proposed method, the conservative seen best node corresponds to the performance level with $\mathcal{ACC}_U$ the same as that of the baseline, the conservative unseen best node corresponds to the performance level with $\mathcal{ACC}_S$ the same as that of the baseline, and the overall best node denotes the performance with the highest H. When λ is in the region between the two conservative nodes, both $\mathcal{ACC}_S$ and $\mathcal{ACC}_U$ are beyond the baseline, indicating that a more discriminative recognizer can be achieved.

For a more intuitive view, classification accuracies for three confusing classes in AwA2 are demonstrated in Fig. 4.6 where bobcat is one of the unseen classes, leopard and wolf are two seen classes. Applying SR, 20.31% more bobcat samples can be correctly recognized than without applying SR. Simultaneously, the adversarial training prevents a 21.01% accuracy drop caused by the misleading sample re-adjustment for the wolf.

To illustrate the importance of the gradient penalty, we also compare the performances of the proposed advRN with and without the gradient penalty. As seen in Fig. 4.7 and 4.8, we can inspect an obvious difference between the models. For the model trained without the gradient penalty, the learned recognizer appears more sensitive to perturbations, especially for seen classes. As a result, $\mathcal{ACC}_S$ falls sharply when λ increases, and the overall performance is mediocre. In comparison, with the gradient penalty, the learned recognizer shows high robustness to seen classes. Moreover, while promoting the recognition of the seen classes, this gradient penalty can also increase the model's generalization to unseen classes, as the upper bound of the unseen accuracy is also slightly increased in Fig. 4.7 and 4.8.

4.4 Summary and Discussion

In this chapter, we have designed a sample re-adjustment process for generalized zero-shot learning to increase the model's awareness of unseen classes. However, this adversarial perturbation based adjustment may mislead the correct judgment about the seen class. Therefore, with the idea of adversarial training, we have also designed a gradient penalty which makes the model less sensitive to the input perturbations for those seen classes. Applying both the sample re-adjustment and gradient penalty, we attain a more generalized model that has achieved superior performance compared to the state-of-the-art methods. Extensive quantitative experimental results have validated the effectiveness of our work on four benchmark datasets.

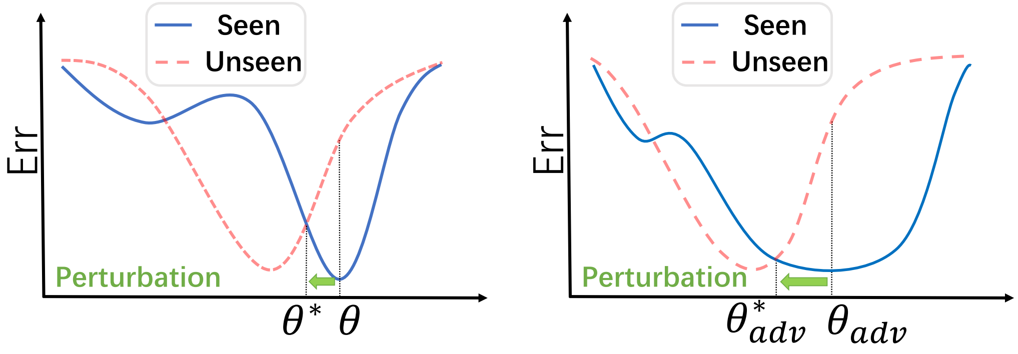
The proposed adversarial framework in this chapter is applied to obtain robustness in sample space. Though such design improves the model to achieve a higher comprehensive performance in generalized tasks, it only considered perturbations on a single sample. In other words, each time doing the recognition, the framework only enhances the sensitivity to unseen classes based on the knowledge from a single instance. The classification knowledge common to multiple individuals is not effectively utilized. To overcome this drawback, we propose an adversarial framework for parameter space in the next chapter.

Chapter 5

Instance-Specific Perturbation on Parameters for Relation Network Based Generalized Zero-Shot Learning

In this chapter, we consider a strict generalized zero-shot learning scenario in which only the seen set (samples and attributes for seen classes) are available during training. For those methods that rely on learned metrics, we propose an adversarial perturbation mechanism during the test to alleviate over-fitting in zero-shot learning while avoiding the use of unseen class attributes during training. Based on a trained embedding model with the metric, for each instance, the highest confidence seen and unseen inferences can be obtained. Then perturbations on model parameters are derived through a min-max problem for obtaining small directions which minimize the seen confidences while maximizing unseen confidences. By subtracting these perturbations, the parameters become instance-specific to alleviate over-fitting on seen classes and consequently make fairer predictions. Additionally, to protect the seen instance from being excessively affected by the perturbation, a parameter-wise adversarial training process is developed to ensure the model robustness on seen classes. In the optimization process, the parameters are optimized via another min-max problem favoring the direction of simultaneously minimizing both the original loss function and that of a corresponding maximum one in the local region. Such a process has been proved to feed back a more generalized model with a flat local minimization loss [142].

Consequently, as illustrated in Fig. 5.1, predictions of most of the seen classes keep consistent, while the perturbation mainly enhanced unseen confidences for unseen class instances. Note that the whole framework is designed with the focus on alleviating the over-fitting in generalized zero-shot learning, the discrimination between unseen classes may not be improved. In other words, by applying the proposed framework, the improve-



(a) Normally trained (b) Adversarially trained

Fig. 5.1: An example of test errors (seen and unseen) vs. model parameters showing the proposed perturbation with (b) and without (a) adversarial training.

ment of the model in generalized zero-shot learning is restricted by its performance in conventional zero-shot learning. The main contributions of this chapter are summarized below:

- We developed a parameter-wise adversarial training process for metric learning based embedding methods in generalized zero-shot learning to ensure the model robustness on seen classes while proposing a novel parameter perturbation mechanism to enhance the model sensitiveness on unseen classes. The framework effectively alleviates the over-fitting problem in generalized zero-shot learning, avoiding employing unseen information during training.

- To the best of our knowledge, this is the first work applying the parameter perturbation framework to improve the zero-shot classification. Compared with the ideal perturbating features [17], the proposed framework can consider multiple instances jointly to avoid extreme perturbation in parameter space. Taking Relation network [14] as the baseline, the proposed framework attains outstanding performance and can even outperform the state-of-the-arts which access unseen attributes in their training on AWA1 and AWA2 benchmarks.

5.1 Methodology

In this section, we first review the definition of inductive zero-shot learning. We then introduce the parameter-wise adversarial training (PAT) process as well as the model perturbation mechanism (MPM) in detail separately.

5.1.1 Problem Definition

Denote $\boldsymbol{X} = \{x_1, ..., x_N\}$, $\boldsymbol{Y} = \{y_1, ..., y_N\}$, $\boldsymbol{A} = \{a_1, ..., a_K\}$ as visual feature, class label, and semantic attribute sets respectively, with the total sample size N and class number K. First we divide the category or class set into two parts, $K = K^S + K^U$, and denote the first K^S classes as seen classes for convenience. Then the dataset is divided into training and test sets, as $\boldsymbol{X} = \boldsymbol{X}_{tr} \cup \boldsymbol{X}_{te}$, $\boldsymbol{Y} = \boldsymbol{Y}_{tr} \cup \boldsymbol{Y}_{te}$ and $N = N_{tr} + N_{te}$, where $\boldsymbol{X}_{tr} \cap \boldsymbol{X}_{te} = \emptyset$. Specifically, the training set only contains samples from seen classes, while the test set may contain those of both seen and unseen, i.e. $\forall y_i \in \boldsymbol{Y}_{tr}, y_i \leq K^S$ and $\forall y_m \in \boldsymbol{Y}_{te}, 0 < y_i \leq K$. Given a training set $\boldsymbol{D}_{tr} = \{(x_i, y_i, a_{y_i}) | x_i \in \boldsymbol{X}_{tr}\}$, the target of zero-shot learning is to train a classifier $C(x_i, a_1, ..., a_{K^S}) = \hat{y}_i$ on $\boldsymbol{D}_{tr}$ to achieve classification on a test set $\boldsymbol{D}_{te}$, where $\boldsymbol{D}_{te} = \{(x_m, y_m, a_{y_m}) | x_m \in \boldsymbol{X}_{te}, K^S < y_m \leq K\}$ for conventional zero-shot learning, and $\boldsymbol{D}_{te} = \{(x_m, y_m, a_{y_m}) | x_m \in \boldsymbol{X}_{te}, 0 < y_m \leq K\}$ for generalized zero-shot learning. In this chapter, we only focus on this strict generalized zero-shot learning case.

5.1.2 Parameter-wise Adversarial Training

In the embedding methods based on metric learning, the semantic attribute vectors a_k are projected into the visual feature space by a learnable embedding function $f_\theta(\cdot)$. Then a metric function $g_\phi(\cdot)$ is trained to capture the relation score (similarity) between the feature x_i and the attribute vector a_k as follows:

$$s(x_i, a_k; \phi, \theta) = g_\phi(x_i \oplus f_\theta(a_k)), \tag{5.1}$$

where $\oplus$ represents the concatenation and $r(x_i, a_k)$ indicates whether x_i and a_k represent the same class. The parameter θ and ϕ are optimized in a meta-learning process. In each optimizing episode, N_b features (corresponding to totally K_τ classes) are sampled from the training set to construct a mini-classification task. Then, by computing the MSE between the relation scores and the one-hot label, a loss function can be designed for optimizing model parameters θ and ϕ:

$$\begin{gathered} L_R(\phi, \theta) = \frac{1}{K_\tau} \frac{1}{N_b} \sum_{k=1}^{K_\tau} \sum_{i=1}^{N_b} [s(x_i, a_k; \phi, \theta) - r(x_i, a_k)]^2, \\ r(x_i, a_k) = \begin{cases} 0 & k \neq y_i, \\ 1 & k = y_i. \end{cases} \end{gathered} \tag{5.2}$$

Our parameter-wise adversarial training process aims to obtain a set of more robust model parameters on seen classes. These parameters should be resistant to perturbations

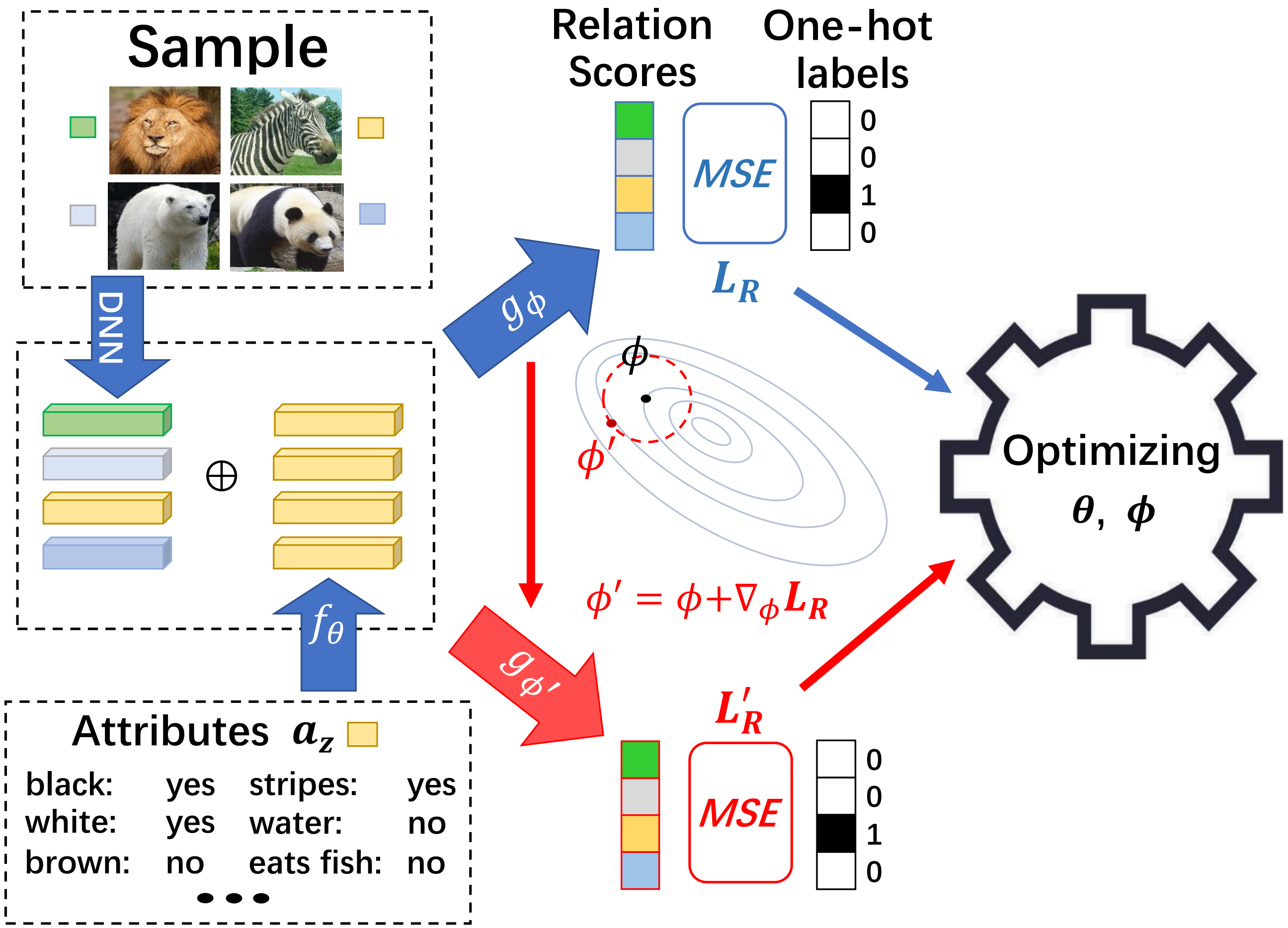


Fig. 5.2: Flowchart of the proposed framework during training.

when calculating the relation scores for seen classes. To achieve this goal, we first define a local region of ϕ with radius ϵ as:

$$B(\phi; \epsilon) = \{\omega : ||\omega - \phi||_2 \leq \epsilon\}. \tag{5.3}$$

Then the corresponding maximum loss in the local region can be represented as:

$$L_R^{'}(\phi, \theta) = \max_{\phi^{'} \in B(\phi; \epsilon)} L_R(\phi^{'}, \theta)). \tag{5.4}$$

To make the radius of the local region more flexible for each optimizing episode, we calculate the maximum corresponding $\phi^{'}$ through gradient descent to approximately calculate region maximum loss:

$$L_R^{'}(\phi, \theta) \approx L_R(\phi + \sigma \nabla_\phi L_R(\phi, \theta), \theta), \tag{5.5}$$

where σ denotes a flat rate. This additional derived loss term is in a similar fashion as the adversarial model perturbation loss [142], which has been proved to lead to flatter local minima. The flattening property actually ensures the desired immunity to perturbations. Finally, the optimization loss for each episode can be summarized as below:

$$L_R^{ep}(\phi, \theta) = L_R(\phi, \theta) + \alpha L_R^{'}(\phi, \theta) + \beta ||\theta||_2, \tag{5.6}$$

where α and β are the trade-off parameters. A diagram of the training process can be found in Fig. 5.2.

5.1.3 Model Perturbation Mechanism

With a trained similarity metric $s(\cdot, \cdot; \phi, \theta)$, the inference for instance x_m can be regarded as the class k with the highest similarity. To implement the proposed model perturbation mechanism during test, we first calculate the unseen inferences for the instance as:

$$\widetilde{y}_m^u = \mathop{\arg\max}_{k \in \{K^S+1, ..., K\}} s(x_m, a_k; \phi, \theta). \tag{5.7}$$

With such unseen inferences, we could use $\boldsymbol{X}_{te}$ (or $\boldsymbol{X}_{tr}$ when the test instances are not allowed to be inferred together) to construct K^U support sets $\{\boldsymbol{X}_{support}^j | j = 1, ..., K^U\}$ where for each, the instances have the same unseen inference. In order to consider the instances jointly in the prediction process, each time an instance x_m with unseen inference j is formed into a group $\{x_l^j\}_{N_g}$ with other $N_g - 1$ support samples with the same unseen inferences from the support set $\boldsymbol{X}_{support}^j$. Then we define a loss function measuring the distance between these inferences and the unseen inference one-hot label as below:

$$\begin{aligned} L_P(\phi, \theta) =& \frac{1}{K} \frac{1}{N_g} \sum_{k=1}^{K} \sum_{l=1}^{N_g} [s(x_l^j, a_k; \phi, \theta) - l(k)]^2, \\ l(k) =& \begin{cases} 0 & k \neq K^S + j, \\ 1 & k = K^S + j. \end{cases} \end{aligned} \tag{5.8}$$

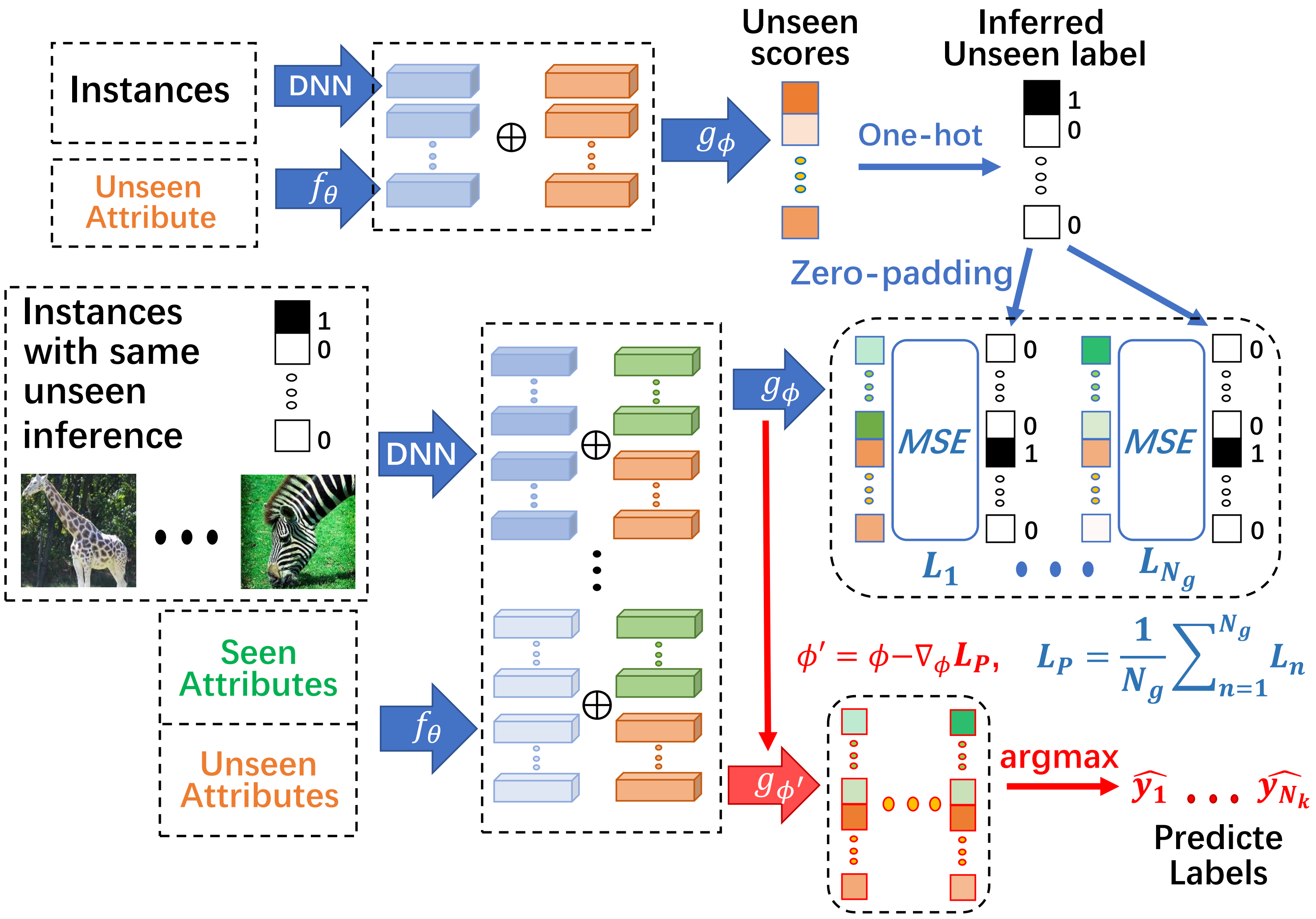


Fig. 5.3: Flowchart of the proposed framework during the test.

Specifically, since the first K^S classes are defined as the seen classes, $k = K^S + j$ stands for that a_k is the corresponding attribute vector for the j-th unseen class. In other words, $l(k) = 1$ only holds when a_k is the corresponding inferred unseen class attributes of x_l^j.

Minimizing such a loss function is equivalent to strengthening the confidence of the unseen inferences for the instance group. Therefore, we apply an iteratively adversarial perturbation to the parameter ϕ according to this loss:

$$\phi[0] = \phi \tag{5.9}$$

$$\phi[p+1] = \phi[p] - \frac{1}{2^p}\lambda\nabla_{\phi[p]}L_P(\phi[p], \theta), \tag{5.10}$$

where λ denotes the perturbation rate.

Finally, with p step perturbation, the predicted label of instance x_m with is calculated as:

$$\widehat{y}_m = \arg\max_{k \in \{1,\dots,K\}} s(x_m, a_k; \phi[p], \theta). \tag{5.11}$$

A diagram of the test process can be found in Fig. 5.3. Following the proposed mechanism, the sensitivity to unseen classes is substantially increased by employing adversarial perturbation on the model. Though this operation may weaken the confidence of the seen classes, the robustness ensured by the adversarial training process can mitigate this impact. Thereby, the whole framework can effectively alleviate over-fitting in generalized zero-shot learning. The detailed steps for the whole framework are presented in Algorithm 2.

5.2 Experiment

In this section, we first review the datasets used in this work as well as the evaluation protocol in detail. We then report the comparison results. After that, we set out ablation study and sensitivity analysis to take closer examinations on our proposed method. All the experiments are implemented based on ubuntu 16.04, PyTorch 1.7.1 and Nvidia RTX1080Ti.

5.2.1 Datesets and Evaluation Protocol

We evaluate the proposed framework on four commonly used zero-shot learning benchmarks: Animals with Attributes 1 (AWA1) [33] and 2 [53], attribute Pascal and Yahoo (aPY) [[121] and Caltech-UCSD-Birds 200-2011 dataset (CUB) [34]. AwA1 and AwA2 are two medium-scale coarse-grained animal classification datasets containing 40 seen classes and 10 unseen classes. aPY is a small-scale coarse-grained dataset with 20 seen

Algorithm 2: Procedure of the proposed training & testing

```
Training:
Input: dataset $\boldsymbol{X}_{tr}$ and $\boldsymbol{Y}_{tr}$with the total sample number $N_{tr}$ and
  $A = \{a_1, ..., a_{K^S}\}$ with class number $K^S$, learning rate $\gamma$, flat rate $\sigma$ and batch
  size $N_b$
Parameter: $\theta$ and $\phi$
for random batch $\{x_i\}_{i=1}^{N_b} \sim X_{tr}$, $\{y_i\}_{i=1}^{N_b} \sim Y^S$ with $K_\tau$ classes
    Compute the $L_R(\phi, \theta)$ through Eq. 5.2
    Calculate region maximum loss $L_R^{'}(\phi, \theta)$ through Eq. 5.5
    Optimize $\theta$ and $\phi$ through Eq. 5.6:
    $\theta \leftarrow \theta - \gamma \cdot \nabla_\theta L_R^{ep}(\phi, \theta)$
    $\phi \leftarrow \phi - \gamma \cdot \nabla_\phi L_R^{ep}(\phi, \theta)$
end for
return $\theta$ and $\phi$
Testing:
Input: Trained similarity metric $s(\cdot, \cdot, \theta, \phi)$; seen class number $K^S$ and unseen
class number $K^U$; attributes $A = \{a_1, ..., a_K\}$; perturbation rate $\lambda$ and step $p$,
instance $x_m \in \boldsymbol{X}_{te}$; (support group size $N_G$ and support sets
$\{\boldsymbol{X}_{support}^j | j = 1, ..., K^U\}$ if exist)
Calculate the unseen inferences $\widetilde{y}_m^u$ for instance through Eq. 5.7
if support sets $\{\boldsymbol{X}_{support}^j | j = 1, ..., K^U\}$ exist
    Construct the group $\{x_l^{\widetilde{y}_m^u}\}_{N_g}$ with the instance and the corresponding support
    set $\boldsymbol{X}_{support}^{\widetilde{y}_m^u}$
else
    Construct the group $\{x_l^{\widetilde{y}_m^u}\}_1$ only use the single test instance.
end if
Calculate the perturbed parameter $\phi[p]$ through Eq. 5.10
Predict the final label $\widehat{y}_m$ of instance $x_m$ with the perturbation mechanism through
  Eq.5.11
return predicted label $\widehat{y}_m$
```

Table 5.1: Dataset statistics.

benchmark	attribute dimension	sample size		
		train	test$_{seen}$	test$_{unseen}$
AwA1	85	19832	4958	5685
AwA2	85	23527	5882	7913
aPY	64	5932	1483	7924
CUB	312	7057	1764	2967

classes and 12 unseen classes, and CUB is a medium-scale fine-grained bird dataset with 150 seen classes and 50 unseen classes. More details can be found in Table 5.1.

We follow the data splits in the GBU setting [53] to avoid class overlapping between the test set for zero-shot learning and the training set for the pre-trained DNN. The performance is evaluated through top-1 average per-class accuracy ACC. Specifically, ACC_S and ACC_U denote the seen and unseen classes' accuracies in generalized zero-shot learning, respectively. In addition, a harmonic mean of these accuracies is defined to measure the overall performance as follows:

$$H = \frac{2 \times \mathcal{ACC}_S \times \mathcal{ACC}_U}{\mathcal{ACC}_S + \mathcal{ACC}_U}. \tag{5.12}$$

We take Relation Network [14], termed shortly as RN and CRNet [82], as two typical baselines to validate the effectiveness of our proposed method. In implementing RN, both of the embedding f_θ and metric functions g_ϕ are designed as three-layer neural networks with most of the activation functions as relu except g_ϕ using sigmoid at the output layer. The input and output dimensions of f_θ are consistent with the attributes and feature spaces, respectively, and g_ϕ outputs a scalar as the inferred similarity of the input feature pair. The detailed setting for the whole framework can be found in Table 5.2. For CRNet, the structure of the metric function is the same, while multiple embedding functions as two-layer neural networks $\{f_\theta^k | k = 1, ..., K\}$ are designed for embedding. A K-clustering is first applied over the seen attributes, then the difference between the attributes and the corresponding k-th cluster centre will be regarded as the input for f_θ^k. And the final embedded feature can be obtained as the sum of all the outputs of the embedding functions. The detailed setting for the whole framework can be found in Table 5.3.

Table 5.2: Specific experiment settings for the proposed framework applied on RN.

benchmark	Training							Test		
	Layer structure		Learning	Batch	flat rate	Regularization rate		Perturbation		Group
	f_θ	g_ϕ	rate	size	σ	α	β	rate λ	step p	size N_G
AwA1	$85{\times}1600 \times 2048$	$4096{\times}400 \times 1$	1×10^{-5}	32	0.5	0.1	1×10^{-5}	0.35	3	128
AwA2	$85{\times}1600 \times 2048$	$4096{\times}400 \times 1$	1×10^{-5}	32	0.5	0.1	1×10^{-5}	0.35	3	128
aPY	$64{\times}1200 \times 2048$	$4096{\times}400 \times 1$	1×10^{-4}	32	5	0.3	1×10^{-5}	6	3	128
CUB	$312{\times}1200 \times 2048$	$4096{\times}1200 \times 1$	1×10^{-5}	32	0.5	0.1	1×10^{-5}	0.6	3	128

Table 5.3: Specific experiment settings for the proposed framework applied on CRnet.

benchmark	Training										Test		
	K	Layer structure		learning rate		batch	flat rate	Regularization rate			Perturbation		group
		f_θ	g_ϕ	f_θ	g_ϕ	size	σ	α	β_θ	β_ϕ	rate λ	step p	size N_G
AwA1	3	85 ×2048	4096 ×2048 × 1	5×10^{-5}	1×10^{-5}	32	0.3	0.1	1×10^{-5}	1×10^{-4}	0.04	3	128
AwA2	3	85 ×2048	4096 ×2048 × 1	5×10^{-5}	1×10^{-5}	32	0.3	0.1	1×10^{-5}	1×10^{-4}	0.04	3	128
aPY	4	64 ×2048	4096 ×2048 × 1	5×10^{-5}	1×10^{-5}	32	0.3	0.1	1×10^{-5}	1×10^{-4}	0.04	3	128
CUB	3	312 ×2048	4096 ×2048 × 1	1×10^{-5}	0	32	0.3	0.1	1×10^{-5}	0	0.12	3	128

5.2.2 Comparisons with State-of-the-Art Methods

Since our motivation is to alleviate the over-fitting, we conduct comparisons mainly in the generalized zero-shot learning scenario as shown in Table 5.5. Additionally, Table 5.6 provides the comparison with the baseline in conventional zero-shot learning to verify that the developed parameter-wised adversarial training has no negative effects on the baseline.

Since our work focus on improving the embedding methods based on metric learning, the comparisons in Table 5.5 are divided into three parts. The first part contains the results of the methods which employed knowledge from unseen attributes when training the classifier (mainly the generative methods). Both the second and third parts provide the results for the embedding methods where the former applies settled distance metrics while the latter learns metrics during training. CRnet [82] and RN [14] as two metric learning based embedding methods are selected as the baselines. As shown in Table 5.5, though some of our implementations of the baselines (marked with *) return poorer performance compared with the reported results, the proposed mechanism significantly improves them, and the results outperform all the other embedding methods based on metric learning. The proposed method achieves the H values of 70.0% and 69.9% on datasets AWA1 and AWA2, respectively, which even reach the level of those methods taking advantage of additional unseen information during training. We have mentioned that the performance of the proposed framework is bounded by the conventional zero-shot learning performance of the baseline. Since the baseline only obtains 40.0% and 58.5% conventional zero-shot learning accuracies, our model does not reach the top on CUB and aPY. However, it is still competitive with the other embedding methods, particularly, the framework performs significantly better when multiple instances are considered in the perturbation mechanism compared to the AdvRN approach which shares a similar motivation.

In this section, considering RN as an example, we mainly verify the effectiveness of each part in our proposed framework. The curves of three performance metrics vs. perturbation rate λ are plotted in Figs. 5.4, 5.5, 5.6 and 5.7 which demonstrates the sensitivities of the models to λ. It is obvious that, compared with the proposed model, ACC_S of the baseline drops significantly while λ increases. It means the proposed adversarial training process effectively moderates the misleading of perturbations on the seen class, thus enabling the model to survive stronger perturbations to improve the unseen recognition. As a result, the metric H of the proposed method retains a series of stable high values over a wide range of the perturbation rate. In other words, benefiting from the proposed PAT process, the improved robustness leads the model to become more adaptable for the perturbation parameters.

Table 5.4: Comparisons between the methods with the proposed mechanism and the other state-of-the-art methods. ψ denotes methods additionally leveraging unseen attributes during training and the superscript * represents the results reproduced in our own implementation.

	Method	AwA1			AwA2			aPY			CUB		
		$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H
ψ	f-CLSWGAN [6]	57.9	61.4	59.6	–	–	–	–	–	–	43.7	57.7	49.7
	TCN [72]	49.4	76.5	60.0	61.2	65.8	63.4	24.1	64.0	35.1	52.6	52.0	52.3
	CADA-VAE [8]	57.3	72.8	64.1	55.8	75.0	63.9	–	–	–	51.6	53.5	52.4
	DAZLE [99]	–	–	–	60.3	75.7	67.1	–	–	–	56.7	59.6	58.1
	OCD-CVAE [143]	–	–	–	59.5	73.4	65.7	–	–	–	44.8	59.9	51.3
	Disentangled-VAE [106]	60.7	72.9	66.2	56.9	**80.2**	66.6	–	–	–	51.1	58.2	54.4
	GCM-CF [144]	–	–	–	60.4	75.1	67.0	37.1	56.8	44.9	61.0	59.7	60.3
	TGMZ [119]	65.1	69.4	67.2	**64.1**	77.3	**70.1**	**34.8**	**77.1**	**48.0**	60.3	56.8	58.5
	CE-GZSL [10]	**65.3**	**73.4**	**69.1**	63.1	78.6	70.0	–	–	–	**63.9**	**66.8**	**65.3**

Table 5.5: Comparisons between the methods with the proposed mechanism and the other state-of-the-art methods. ζ denotes embedding methods with settled metrics, ξ denotes embedding methods with learned metrics and the superscript * represents the results reproduced in our own implementation.

	Method	AwA1			AwA2			aPY			CUB		
		$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H
ζ	SAE [56]	1.8	77.1	3.5	1.1	82.2	2.2	0.4	**80.9**	0.9	7.8	54.0	13.6
	DEM [13]	32.8	**84.7**	47.3	30.5	86.4	45.1	11.1	75.1	19.4	19.6	57.9	29.2
	PSR-ZSL [60]	–	–	–	20.7	73.8	32.3	13.5	51.4	21.4	24.6	54.3	33.9
	Triple Verification Net [51]	27.0	67.9	38.6	–	–	–	16.1	66.9	25.9	26.5	62.3	37.2
	LFGAA+Hybrid [92]	–	–	–	27.0	**93.4**	41.9	–	–	–	36.2	**80.9**	50.0
	DVBE [86]	–	–	–	**63.6**	70.8	**67.0**	32.6	58.3	41.8	53.2	60.2	**56.5**
	DPR-GZSL [145]	**54.7**	81.9	**65.6**	–	–	–	**34.9**	65.5	**45.5**	**66.6**	48.9	56.4
ξ	RN [14]	31.4	**91.3**	46.7	30.0	**93.4**	45.3	21.2	68.8	32.4	38.1	61.1	47.0
	CRnet [82]	58.1	74.7	65.4	52.6	78.8	63.1	32.4	68.4	44.0	45.5	56.8	50.5
	advRN [17]	50.6	79.5	61.8	49.3	84.0	62.2	28.0	66.0	39.3	44.3	62.6	51.9
	CRnet*	51.7	83.9	64.0	47.6	86.6	61.4	15.9	**83.0**	26.7	47.1	56.4	51.4
	RN*	21.8	88.9	34.2	21.5	90.9	34.8	16	79.7	26.6	41.2	**63.1**	49.9
	Padv-CRnet(ours)	59.8	78.2	67.7	58.1	81.0	67.7	28.7	69.9	40.7	48.8	57.0	52.6
	Padv-RN(ours)	**63.1**	78.8	**70.0**	**62.2**	79.8	**69.9**	**33.8**	64.6	**44.4**	**50.2**	57.7	**53.7**

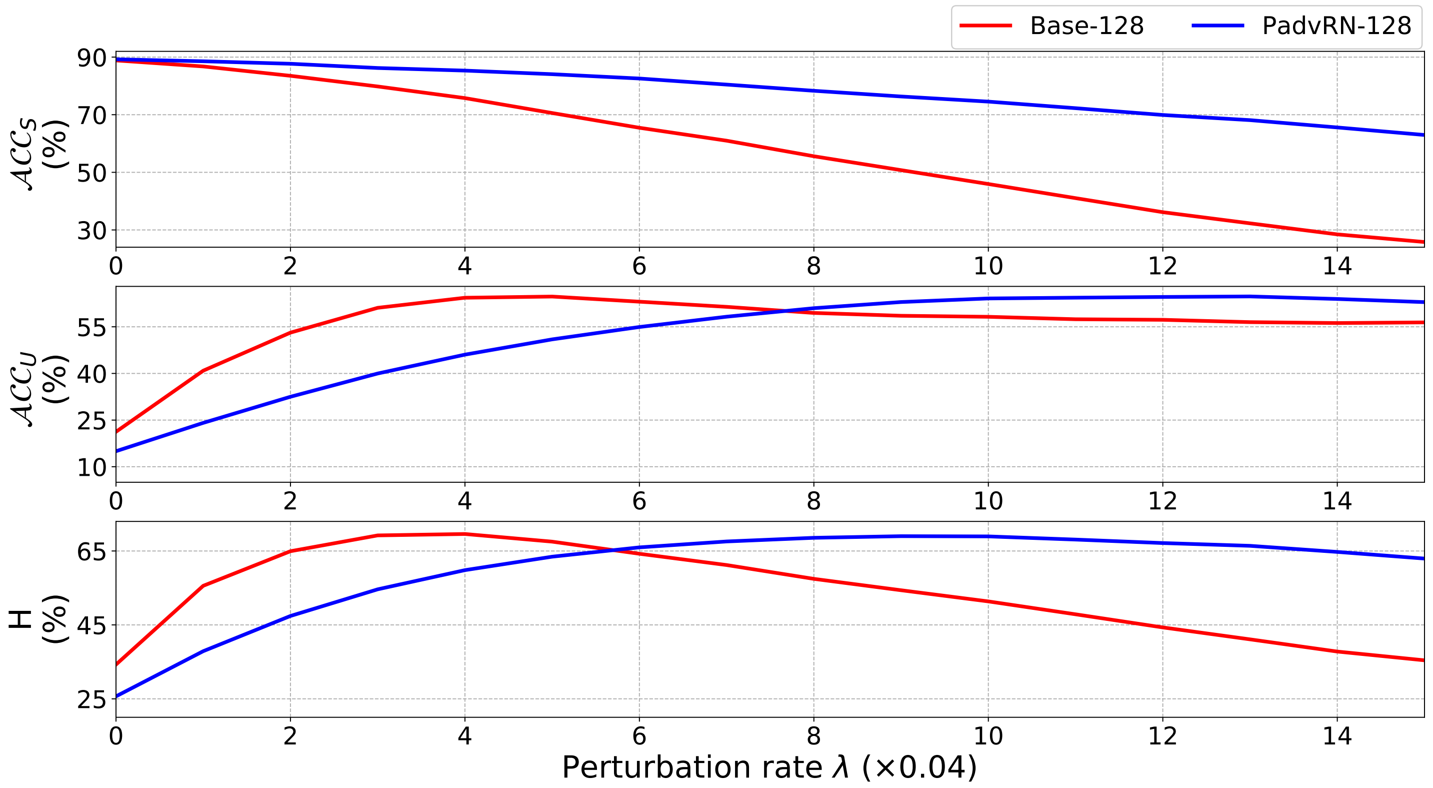


Fig. 5.4: Accuracies (ACC_S, ACC_U and H) vs. perturbation rate λ on AWA1

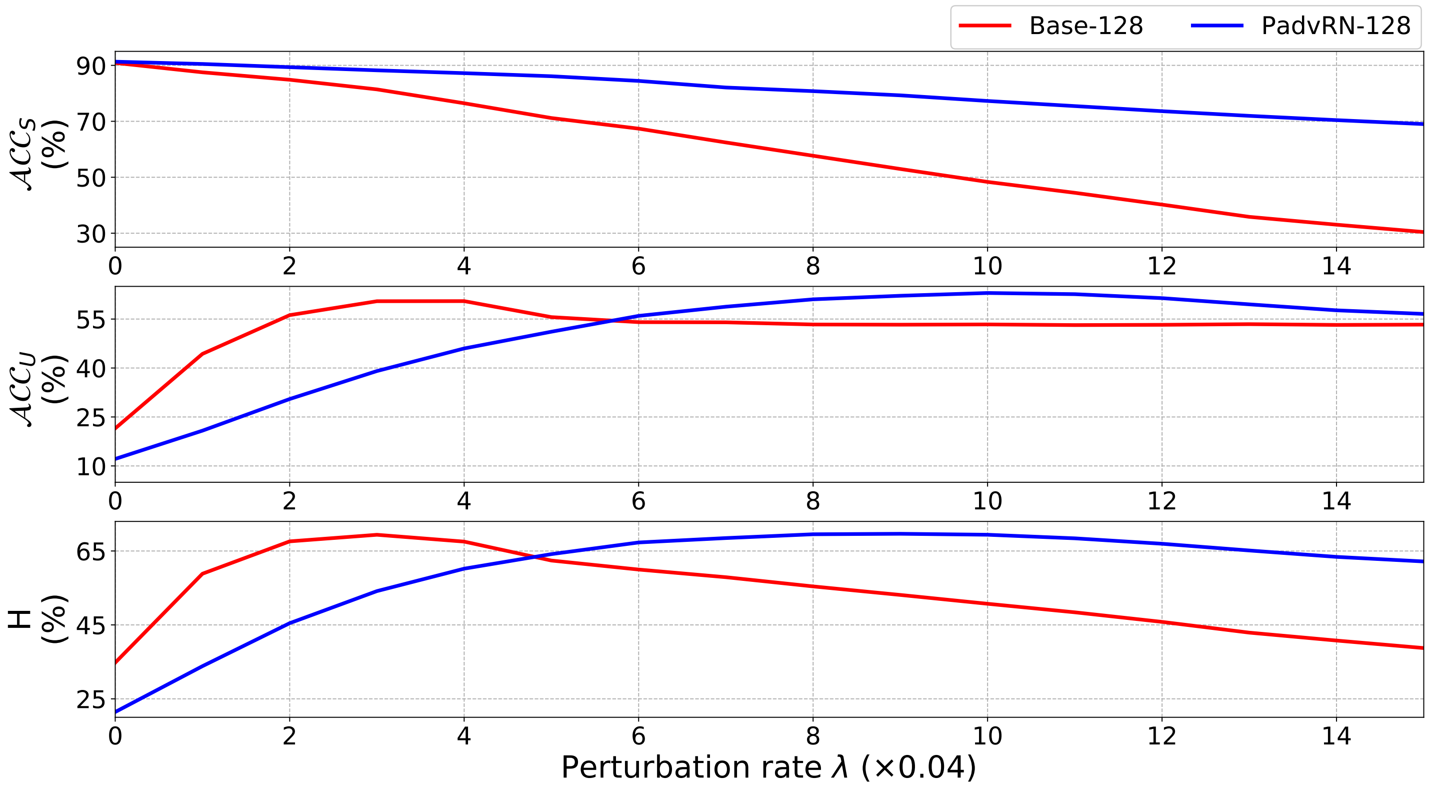


Fig. 5.5: Accuracies (ACC_S, ACC_U and H) vs. perturbation rate λ on AWA2

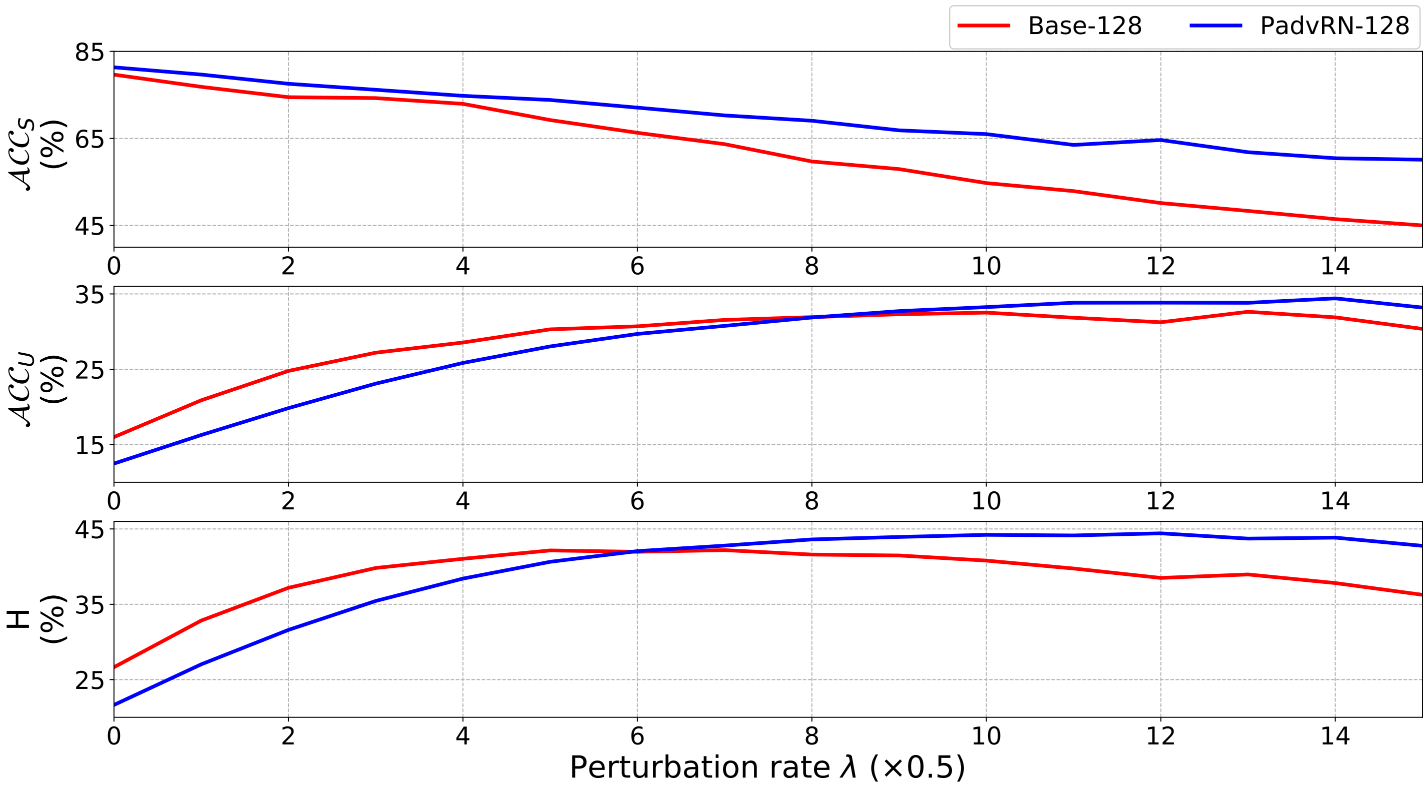


Fig. 5.6: Accuracies (ACC_S, ACC_U and H) vs. perturbation rate λ on aPY

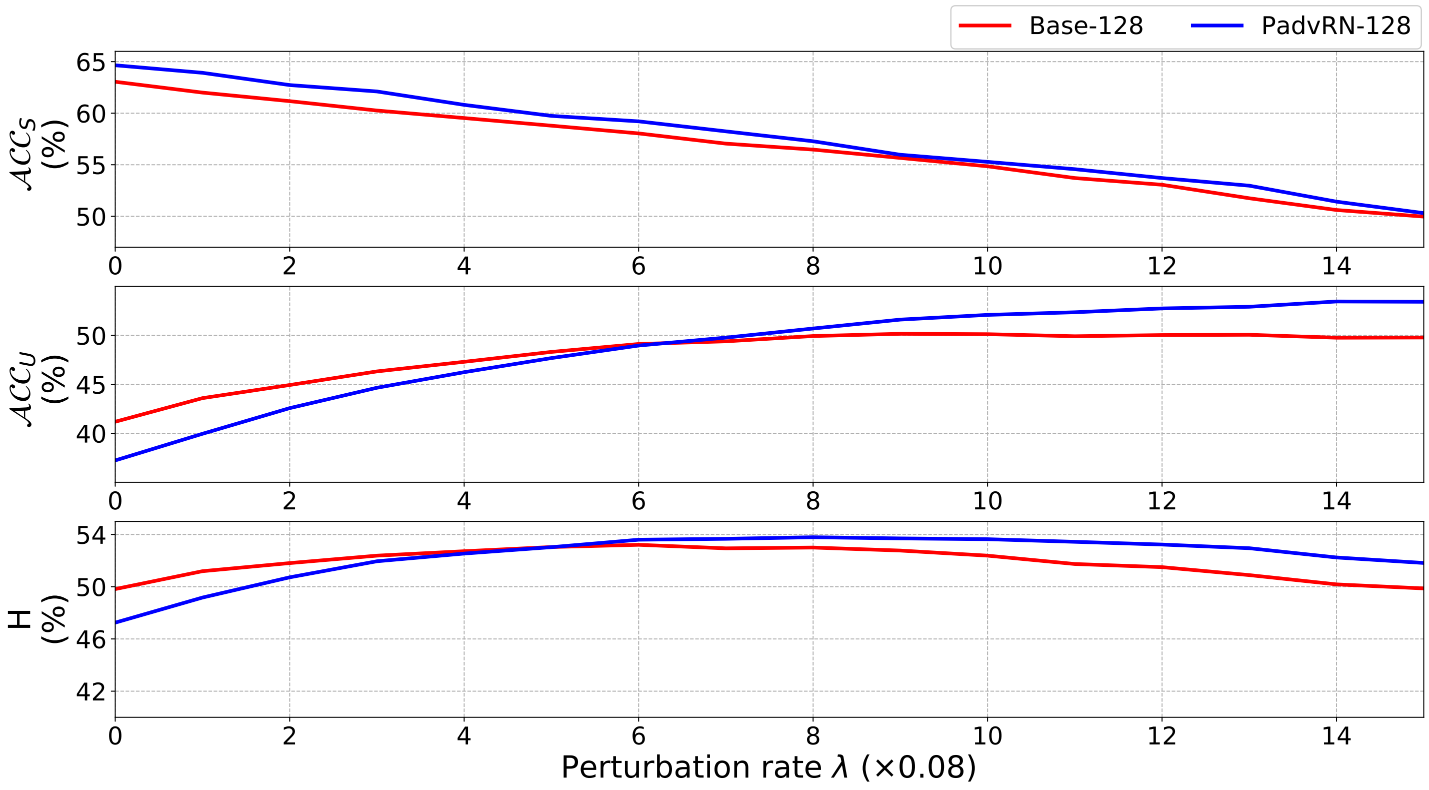
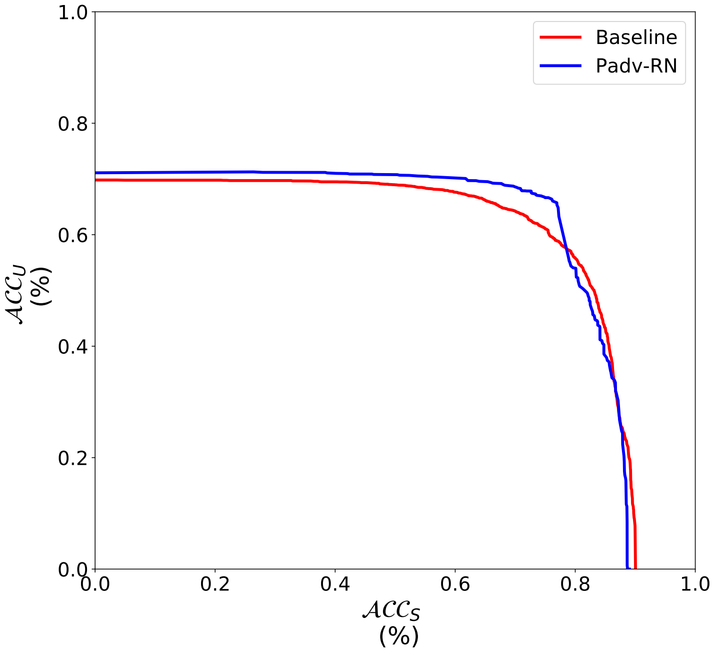
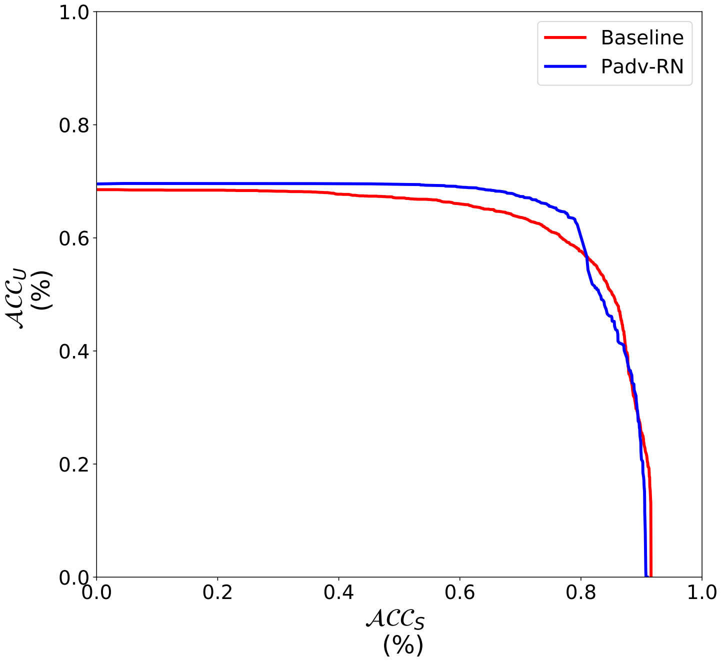


Fig. 5.7: Accuracies (ACC_S, ACC_U and H) vs. perturbation rate λ on CUB

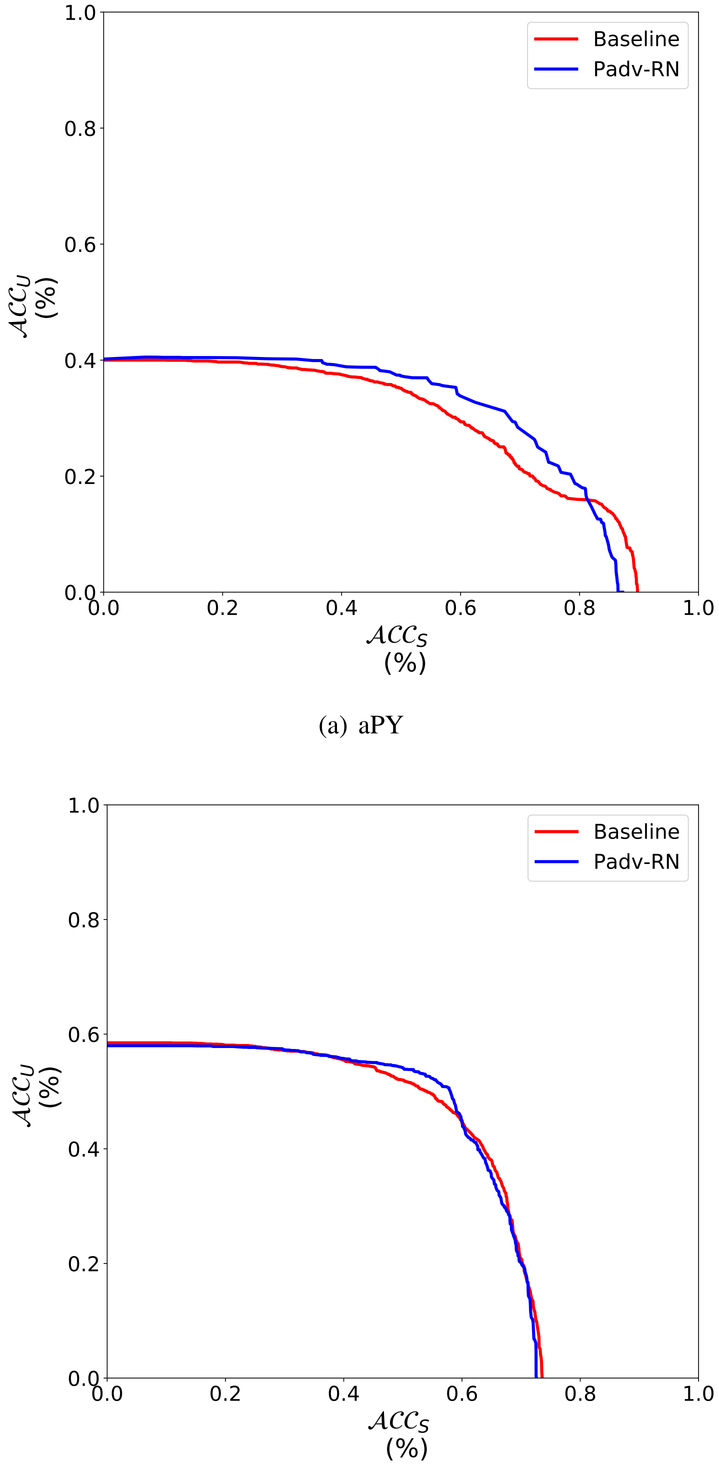


(a) AwA1



(b) AwA2

Fig. 5.8: Seen vs. unseen trade-off curve for the baseline RN and the proposed method on (a) AwA1, (b) AwA2



(a) aPY

(b) CUB

Fig. 5.9: Seen vs. unseen trade-off curve for the baseline RN and the proposed method on (a) aPY, (b) CUB

Table 5.6: Comparisons between the proposed Padv-RN, Padv-CRnet, and their baseline in conventional zero-shot learning.

Method	AwA1	AwA2	aPY	CUB
CRnet*	69.1	68.7	39.7	56.7
Padv-CRnet	69.4	68.8	40.4	57.0
RN*	69.8	68.5	40.0	58.5
Padv-RN	69.9	69.5	40.7	58.0

5.2.3 Effectiveness Analysis

For a quantitative display, an ablation study is presented in Table 5.7. Applying the multi-instance based perturbation benefits both the baseline and the proposed model. Employing simply the adversarial training ruins the sensitivity of unseen classes since the enhanced robustness may lead to a lower generalization. However, once combining the adversarial training with the proposed perturbation mechanism, we observe clear performance gains over the case where only the perturbation mechanism is applied in general.

To verify the effectiveness of the proposed model for alleviating over-fitting. The unseen vs. seen trade-off curves are demonstrated in Fig. 5.8 and 5.9. This curve is obtained by simply increasing the predicted scores for all the seen or unseen classes. The proposed framework leads to a larger area under the curve which means the distinguishability between seen and unseen classes is more outstanding. The highest achieved H values in these curves are listed in Table 5.8

To avoid misunderstandings, here we will elaborate that our approach is not a transductive method. In our method, a group of test samples are only used to calculate the perturbation, which was not involved in training. Actually, even if not using a group, one single test sample can also boost the performance. To clarify, we listed the results under different test group settings in the second part of Tab. 5.7 , where '1test', '4train', '32test' and '128test' denote calculating the perturbation based on: one single instance, grouped with 3 training samples as support, grouped with other 31, and 127 test samples, respectively. We highlight some points as follows:

1). Parameter-wise Adversarial Training mainly makes the trained model adapted to a wider range of perturbations, i.e. more practicable, and also improves the performance with Model Perturbation Mechanism in some benchmarks.

2). Model Perturbation Mechanism is not a training process. Similar to an adversarial at-

Table 5.7: Ablation study on the proposed framework and specific comparisons for different test group setting. where Padv-RN represents the proposed framework with perturbation group size 128.

Method	AwA1			AwA2			aPY			CUB		
	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H	$\mathcal{ACC}_U$	$\mathcal{ACC}_S$	H
RN* (baseline)	21.8	88.9	34.2	21.5	90.9	34.8	16	79.7	26.6	41.2	63.1	49.9
Padv-RN w/o MPM	15.9	89.2	27	12.2	91.3	21.5	12.5	81.3	21.7	37.2	64.7	47.2
Padv-RN w/o PAT	64.0	76.7	69.8	60.7	81.3	69.5	30.8	67.9	42.4	49.2	58.2	53.3
Padv-RN	63.1	78.8	**70.0**	62.2	79.8	**69.9**	33.8	64.6	**44.4**	50.2	57.7	**53.7**
Padv-RN$_{1test}$	63.2	70.1	66.5	58.2	72.5	64.6	30.3	67.2	41.8	46.1	59.6	52.0
Padv-RN$_{4train}$	60.8	75.2	67.2	59.6	74.7	66.3	32	63.4	42.5	47.8	58.2	52.5
Padv-RN$_{32test}$	63.9	75.4	69.2	62.7	78	69.5	35.2	59.1	44.1	50.8	56.8	53.6
Padv-RN$_{128test}$	63.1	78.8	70.0	62.2	79.8	69.9	33.8	64.6	44.4	50.2	57.7	53.7

Table 5.8: The best H value in the unseen vs. seen trade-off curve.

	AwA1	AwA2	aPY	CUB
RN*	67.3	67.4	41.2	52.1
Padv-R	70.6	70.3	44.4	53.8

tack that obtains the worst case of the task, we reversely employ the perturbation and find the best case of unseen inference for each instance (or group), in the neighborhood of the parameters. In other words, it is an advanced inference process that involves calculating the gradients, while the trained model is not updated.

3). **Padv-RN**$_{1test}$ and **Padv-RN**$_{4train}$ could be regarded as the results strictly following the inductive ZSL setting where the support of the training samples could be considered as employing class prototypes from training samples which does not break the inductive settings. Under the same setting, the proposed method outperforms the previous work [29]. Moreover, as a slight variant of our work, the perturbation could be calculated with the support of training samples which improves the performance.

4). The perturbation is instance-specific, which only depends on the instance (or a group of instances). Other test samples won't participate in or affect the inferences. This is different from transductive learning, where all test samples are available during training. Employing only 32 instances sharing the perturbation inside a group during the test, we find the results could be significantly improved.

In addition, we study the model sensitivity to each hyper-parameter by settling the others same as the setting for comparisons. The results in AwA2 are displayed in Fig. 5.10 as one illustrative example.

5.3 Summary and Discussion

In this chapter, we have introduced a model adversarial perturbation framework for generalized zero-shot learning to alleviate the over-fitting problem without accessing any unseen information during training. Instance-specific parameters are obtained through a parameter adversarial training process to increase the sensitivity of unseen classes, while an adversarial training process improves the resistance to such perturbation for seen classes. Comparisons of four commonly used benchmarks demonstrate the effectiveness of the proposed framework. With the advantage of being able to consider instances jointly, the performance of the proposed framework is far superior to the framework with similar motivation and is even competitive with those generative methods in two of the benchmarks.

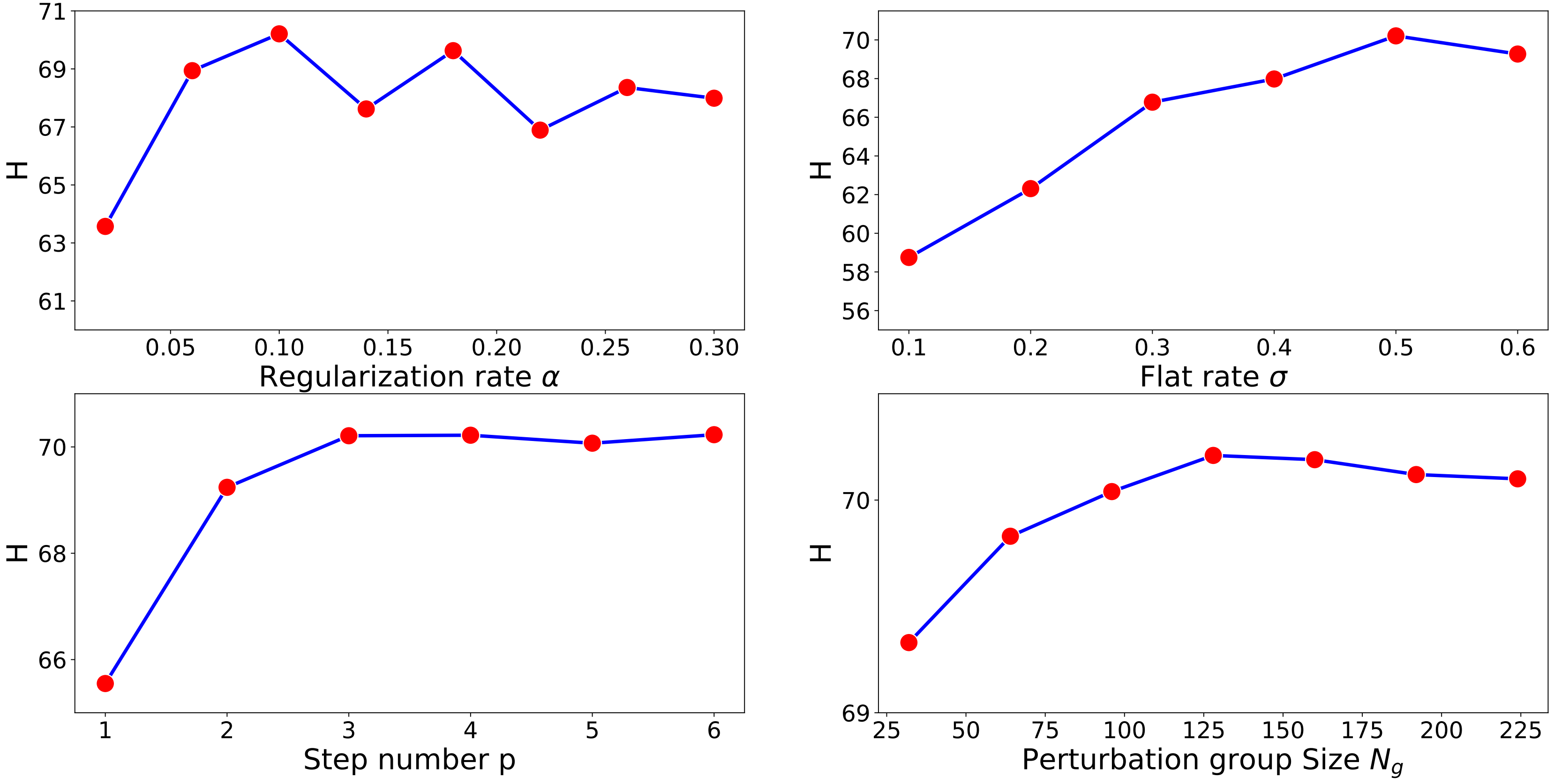


Fig. 5.10: Sensitivity study in AwA2.

Our main motivation for this work is to alleviate the over-fitting of the embedding methods with learned metrics for zero-shot classification. The proposed mechanism can only enhance the sensitivity of recognizing unseen classes, while the discrimination ability between unseen classes is not promoted. In other words, the predictions for instances of unseen classes that have been misclassified as seen might be corrected, whereas those instances that have been incorrectly classified as unseen would not. Furthermore, one more step back-propagation is required in each batch in our proposed adversarial training, which may incur additional training time. How to design a faster yet more efficient algorithm remains one open problem.

Chapter 6

Conclusions

In this dissertation, we have presented a comprehensive survey to introduce the zero-shot learning field thoroughly. To overcome the problem that models return low confidence for unseen classes, we designed three metrics for the embedding based methods, one for the regression-based model and two for learned metrics-based models. This chapter will first briefly review the whole dissertation and then discuss the future works.

6.1 Review of the Dissertation

The survey section specifically introduced zero-shot learning in motivation, learning scenarios, task settings, commonly used benchmarks, representative methods, and comparison results. This research field aims to emulate humans' efficient knowledge transfer capability to construct category concepts from descriptions, thus circumventing the dependence of conventional DNN based models on a large number of labelled samples in practical tasks. To clarify whether the concept of the target class was constructed accurately, the researchers constructed a generalized zero-shot learning task to verify the model's generalization, i.e., the model was required to distinguish between seen and unseen classes while performing the classification. In facing such challenging tasks, different problem settings and diverse experimental setups have emerged, making the comparison results not fair for verifying the model's effectiveness, thus hindering further research in the field. Motivated by this situation, we constructed comparison results based on the implementation details of zero-shot methods to evaluate the model's effectiveness more fairly and critically. Whether the backbone structure has been modified, whether fine-tuning has been conducted, and whether additional knowledge has been used are annotated to delineate in more detail the premises corresponding to the performance of each model. Comparing the methods belonging to the same learning scenario under the exact specific implementation relevant settings helps rigorously clarify the superiority of each proposed model.

Moreover, to overcome the over-fitting problem in generalized zero-shot tasks, i.e., the model focuses excessively on seen classes and becomes insensitive to unseen classes, we have introduced three improvements on metrics for those embedding based zero-shot methods. For a regression-based deep embedding method, we proposed a focus ratios based metric that achieves alleviating class-level over-fitting via employing the correlations between the location and the importance of each dimension in the embedding space. During training, the over-fitted knowledge is shared by the proposed module and the baseline model. Then abandoning the proposed module during instance inference will likely alleviate the over-fitting problem.

As those embedding methods with learned metrics perform outstanding, We design an adversarial framework for the relation network to learn a more generalized model. In the proposed framework, instead of predicting the label on an original instance, an individual that most resembles the unseen class is selected from the neighbourhood to make the inference. This sample perturbation-based relation metric achieves a high response for unseen classes. In order to avoid disrupting the perception of classification of seen classes by the perturbations, a robust training process is additionally applied to keep the prediction for seen roughly consistent, thus leading to a performance that significantly outperforms the baseline.

Finally, we developed the adversarial framework for the parameter space. A parameter perturbation based relation metric inherits the advantages of the feature-based one and achieves more generalized perturbation by considering multiple instances in conjunction. Instead of calculating the perturbation through a single instance, combining the instance with a support group sampled from the training set could contribute to a more classification representative direction for the perturbation, which leads to more precise enhancement for the unseen sensitivity. As a result, the improved model becomes more generalized. When the test instances could be inferences together, the support group constructed by test instances could further boost the model performance.

6.2 Future Work

In this dissertation, our aim is to improve the metrics for those embedding methods with a settled backbone under the inductive learning scenario. We will carry on our studies based on the following ideas:

- Such designs can be extended to apply these improved mechanisms and frameworks to those embedding methods with a modified backbone or those under other learning scenarios.

- These metrics could also be applied in those generative methods to train the classifier. More verification could be carried out to show whether the generalizability brought by these improved metrics is compatible with that of the pseudo-samples, thus further enhancing the generative methods in the discriminating phase.

- The proposed adversarial framework can be modified and applied to the training process of the generator. The samples located at the classification boundary can be obtained by perturbation which could be utilized to construct additional supervision or regularization term.

- We could try to figure out a way to combine the two proposed adversarial frameworks to enhance unseen sensitivity by adapting the perturbations in both sample and parameter spaces. The compatibility of these two frameworks has to be verified. Since perturbing one of the instances and parameters will cause a change in the perturbation of the other one. As a result, the whole process may be relevant to an iterative forward and backward propagation to achieve the perturbation on multi-spaces. Therefore, the simplification of the computation should also be concerned.

Publication List

Here is the publication list during my Ph.D. study:

1. Haochuan Jiang, Guanyu Yang, Kaizhu Huang, Rui Zhang, “W-Net: One-Shot Arbitrary-Style Chinese Character Generation with Deep Neural Networks”, International Conference on Neural Information Processing, Springer, 2018, pp. 483–493.

2. Guanyu Yang, Kaizhu Huang, Rui Zhang, John Y. Goulermas, Amir Hussain, “Self-focus Deep Embedding Model for Coarse-Grained Zero-Shot Classification”, International Conference on Brain Inspired Cognitive Systems, **Best student paper**, Springer, 2019, pp. 12–22.

3. Guanyu Yang, Kaizhu Huang, Rui Zhang, John Y. Goulermas, Amir Hussain, “Inductive generalized zero-shot learning with adversarial relation network”, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 724–739.

4. Guanyu Yang, Kaizhu Huang, Rui Zhang, John Y. Goulermas, Amir Hussain, “Coarse-grained generalized zero-shot learning with efficient self-focus mechanism”, Neurocomputing, vol. 463, pp. 400–410, 2021.

5. Guanyu Yang, Zihan Ye, Rui Zhang, Kaizhu Huang, “A comprehensive survey of zero-shot image classification: methods, implementation, and fair evaluation”, Applied Computing and Intelligence, vol. 2, no. 1, pp. 1–31, 2022.

The following works have been submitted and are being reviewed:

1. Guanyu Yang, Kaizhu Huang, Rui Zhang, “Instance-Specific Model Perturbation Improves Generalized Zero-Shot Learning”, Partern Recognition, 2022 [Submitted]

Appendix

Here is the used dataset list introduced in this dissertation:

1. Animals with Attributes (AwA2) [53] contains 30,475 images from public web sources for 50 highly descriptive animal classes with at least 92 labelled examples per class. For example, the attributes include *stripes*, *brown*, *eats fish* and so on.

2. Caltech-UCSD-Birds-200-2011 datasets (CUB)) [34] is a fine-grained dataset with a large number of classes and attributes, containing 11,788 images from 200 different types of birds annotated with 312 attributes.

3. SUN Attribute (SUN) [120] is a fine-grained dataset, medium-scale in class number, containing 14,340 scene images annotated with 102 attributes, e.g. *sailing/boating*, *glass*, and *ocean*.

4. The dataset Attribute Pascal and Yahoo (aPY) [121] is a small-scale dataset with 64 attributes and 32 object classes, including animals, vehicles, and buildings.

The extracted features by pretrained backbone proposed in [53] can be found through the link:
https://www.mpi-inf.mpg.de/departments/computer-vision-and-machine-learning/research/zero-shot-learning/zero-shot-learning-the-good-the-bad-and-the-ugly/

Reference

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] S. J. Pan, “Transfer learning,” in *Data Classification: Algorithms and Applications*. CRC Press, 2014, pp. 537–570.

[3] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *Proceedings of International Conference on Machine Learning Deep Learning Workshop*, vol. 2, 2015.

[4] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, “Matching networks for one shot learning,” *Advances in Neural Information Processing Systems*, vol. 29, pp. 3630–3638, 2016.

[5] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[6] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, “Feature generating networks for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5542–5551.

[7] M. B. Sariyildiz and R. G. Cinbis, “Gradient matching generative networks for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2168–2178.

[8] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, “Generalized zero- and few-shot learning via aligned variational autoencoders,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8247–8255.

[9] P. Ma and X. Hu, “A variational autoencoder with deep embedding model for generalized zero-shot learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 733–11 740.

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

[11] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *Proceedings of 2nd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” vol. 27, pp. 2672–2680, 2014.

[13] L. Zhang, T. Xiang, and S. Gong, “Learning a deep embedding model for zero-shot learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2021–2030.

[14] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[15] G. Yang, K. Huang, R. Zhang, J. Y. Goulermas, and A. Hussain, “Self-focus deep embedding model for coarse-grained zero-shot classification,” in *Proceedings of International Conference on Brain Inspired Cognitive Systems*, 2019, pp. 12–22.

[16] G. Yang, K. Huang, R. Zhang, J. Y. Goulermas, and A. Hussain, “Coarse-grained generalized zero-shot learning with efficient self-focus mechanism,” *Neurocomputing*, vol. 463, pp. 400–410, 2021.

[17] G. Yang, K. Huang, R. Zhang, J. Y. Goulermas, and A. Hussain, “Inductive generalized zero-shot learning with adversarial relation network,” in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020, pp. 724–739.

[18] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim, and X.-Z. Wang, “A review of generalized zero-shot learning methods,” *Computing Research Repository*, vol. abs/2011.08641, 2020. [Online]. Available: https://arxiv.org/abs/2011.08641

[19] W. Wang, V. W. Zheng, H. Yu, and C. Miao, “A survey of zero-shot learning: Settings, methods, and applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–37, 2019.

[20] K. Huang, A. Hussain, Q.-F. Wang, and R. Zhang, *Deep Learning: Fundamentals, Theory and Applications*. Springer,ISBN 978-3-030-06072-5, 2019.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[23] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological Review*, vol. 94, no. 2, p. 115, 1987.

[24] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1577–1584.

[25] M. Elhoseiny, B. Saleh, and A. Elgammal, "Write a classifier: Zero-shot learning using purely textual descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2584–2591.

[26] J. Lei Ba, K. Swersky, S. Fidler *et al.*, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4247–4255.

[27] Y.-J. Chen, Y.-J. Chang, S.-C. Wen, Y. Shi, X. Xu, T.-Y. Ho, Q. Jia, M. Huang, and J. Zhuang, "Zero-shot medical image artifact reduction," in *IEEE 17th International Symposium on Biomedical Imaging*, 2020, pp. 862–866.

[28] Q. Chen, W. Wang, K. Huang, and F. Coenen, "Zero-shot text classification via knowledge graph embedding for social media data," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9205–9213, 2022.

[29] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han, "From zero-shot learning to conventional supervised classification: Unseen visual data synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1627–1636.

[30] H. Jiang, G. Yang, K. Huang, and R. Zhang, "W-net: one-shot arbitrary-style chinese character generation with deep neural networks," in *Proceedings of International Conference on Neural Information Processing*, 2018, pp. 483–493.

[31] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 646–651.

[32] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2635–2644.

[33] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset." California Institute of Technology, 2011.

[35] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958.

[36] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425–1438, 2015.

[37] D. Parikh and K. Grauman, "Relative attributes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 503–510.

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

[39] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," in *Proceedings of 2nd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.5650

[40] D. Wang, Y. Li, Y. Lin, and Y. Zhuang, "Relational knowledge transfer for zero-shot learning," in *Proceedings of Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2145–2151.

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

[42] Y. Liu, L. Zhou, X. Bai, Y. Huang, L. Gu, J. Zhou, and T. Harada, "Goal-oriented gaze estimation for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3794–3803.

[43] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng, "Zero-shot learning through cross-modal transfer," *Advances in Neural Information Processing Systems*, vol. 26, pp. 935–943, 2013.

[44] R. Qiao, L. Liu, C. Shen, and A. Van Den Hengel, "Less is more: zero-shot learning from online textual documents with noise suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2249–2257.

[45] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal, "Link the head to the"" beak"": Zero shot learning from noisy text description at part precision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5640–5649.

[46] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[47] C. Samplawski, E. Learned-Miller, H. Kwon, and B. M. Marlin, "Zero-shot learning in the presence of hierarchically coarsened labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 926–927.

[48] C. Xie, H. Xiang, T. Zeng, Y. Yang, B. Yu, and Q. Liu, "Cross knowledge-based generative zero-shot learning approach with taxonomy regularization," *Neural Networks*, vol. 139, pp. 168–178, 2021.

[49] D. Jayaraman, F. Sha, and K. Grauman, "Decorrelating semantic visual attributes by resisting the urge to share," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1629–1636.

[50] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.

[51] H. Zhang, Y. Long, Y. Guan, and L. Shao, "Triple verification network for generalized zero-shot learning," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 506–517, 2019.

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

[53] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2018.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[56] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3174–3183.

[57] Y.-H. Hubert Tsai, L.-K. Huang, and R. Salakhutdinov, “Learning robust visual-semantic embeddings,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3571–3580.

[58] Y. Yu, Z. Ji, J. Guo, and Z. Zhang, “Zero-shot learning via latent space encoding,” *IEEE Transactions on Cybernetics*, vol. 49, no. 10, pp. 3755–3766, 2018.

[59] G. Liu, J. Guan, M. Zhang, J. Zhang, Z. Wang, and Z. Lu, “Joint projection and subspace learning for zero-shot recognition,” in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2019, pp. 1228–1233.

[60] Y. Annadani and S. Biswas, “Preserving semantic relations for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7603–7612.

[61] J. Li, X. Lan, Y. Liu, L. Wang, and N. Zheng, “Compressing unknown images with product quantizer for efficient zero-shot classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5463–5472.

[62] Y. Guo, G. Ding, J. Han, and Y. Gao, “Sitnet: Discrete similarity transfer network for zero-shot hashing.” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1767–1773.

[63] Y. Liu, X. Gao, Q. Gao, J. Han, and L. Shao, “Label-activating framework for zero-shot learning,” *Neural Networks*, vol. 121, pp. 1–9, 2020.

[64] Z. Zhang and V. Saligrama, “Zero-shot learning via semantic similarity embedding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4166–4174.

[65] B. Romera-Paredes and P. Torr, “An embarrassingly simple approach to zero-shot learning,” in *Proceedings of International Conference on Machine Learning*, 2015, pp. 2152–2161.

[66] J. Weston, S. Bengio, and N. Usunier, “Wsabie: Scaling up to large vocabulary image annotation,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011, pp. 2764–2770.

[67] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, “Devise: A deep visual-semantic embedding model,” *Advances in Neural Information Processing Systems*, vol. 26, pp. 2121–2129, 2013.

[68] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Label-embedding for attribute-based classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 819–826.

[69] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[70] Z. Ji, H. Wang, Y. Pang, and L. Shao, “Dual triplet network for image zero-shot learning,” *Neurocomputing*, vol. 373, pp. 90–97, 2020.

[71] V. K. Verma and P. Rai, “A simple exponential family framework for zero-shot learning,” in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 792–808.

[72] H. Jiang, R. Wang, S. Shan, and X. Chen, “Transferable contrastive network for generalized zero-shot learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9765–9774.

[73] Y. Long, L. Liu, and L. Shao, “Attribute embedding with visual-semantic ambiguity removal for zero-shot learning,” in *Proceedings of the British Machine Vision Conference*, 2016.

[74] Y. Wang, H. Zhang, Z. Zhang, and Y. Long, “Asymmetric graph based zero shot learning,” *Multimedia Tools and Applications*, vol. 79, no. 45-46, pp. 33 689–33 710, 2020.

[75] A. Li, Z. Lu, J. Guan, T. Xiang, L. Wang, and J. Wen, “Transferrable feature and projection learning with class hierarchy for zero-shot learning,” *International Journal of Computer Vision*, vol. 128, no. 12, pp. 2810–2827, 2020.

[76] B. Xu, Z. Zeng, C. Lian, and Z. Ding, “Semi-supervised low-rank semantics grouping for zero-shot learning,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2207–2219, 2021.

[77] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of 5th International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[78] X. Wang, Y. Ye, and A. Gupta, “Zero-shot recognition via semantic embeddings and knowledge graphs,” in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6857–6866.

[79] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, “Rethinking knowledge graph propagation for zero-shot learning,” in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 487–11 496.

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

[81] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *Proceedings of 5th International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rJY0-Kcll

[82] F. Zhang and G. Shi, “Co-representation network for generalized zero-shot learning,” in *Proceedings of International Conference on Machine Learning*, 2019, pp. 7434–7443.

[83] R. L. Hu, C. Xiong, and R. Socher, “Correction networks: Meta-learning for zero-shot learning,” 2018. [Online]. Available: https://openreview.net/forum?id=r1xurn0cKQ

[84] J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song, “Transductive unbiased embedding for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1024–1033.

[85] Y. Li, J. Zhang, J. Zhang, and K. Huang, “Discriminative learning of latent features for zero-shot recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7463–7471.

[86] S. Min, H. Yao, H. Xie, C. Wang, Z.-J. Zha, and Y. Zhang, “Domain-aware visual bias eliminating for generalized zero-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 664–12 673.

[87] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

[88] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[89] G.-S. Xie, L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, and L. Shao, “Attentive region embedding network for zero-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9384–9393.

[90] G.-S. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, and L. Shao, “Region graph embedding network for zero-shot learning,” in *Proceedings of European Conference on Computer Vision*, 2020, pp. 562–580.

[91] Y. Zhu, J. Xie, Z. Tang, X. Peng, and A. Elgammal, “Semantic-guided multi-attention localization for zero-shot learning,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 14 917–14 927, 2019.

[92] Y. Liu, J. Guo, D. Cai, and X. He, “Attribute attention for semantic disambiguation in zero-shot learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6698–6707.

[93] I. Skorokhodov and M. Elhoseiny, “Class normalization for (continual)? generalized zero-shot learning,” in *Proceedings of 9th International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=7pgFL2Dkyyy

[94] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, “Attribute prototype network for zero-shot learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 969–21 980, 2020.

[95] H. Zheng, J. Fu, T. Mei, and J. Luo, “Learning multi-attention convolutional neural network for fine-grained image recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5209–5217.

[96] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, “f-vaegan-d2: A feature generating framework for any-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 275–10 284.

[97] Y. Zhu, J. Xie, B. Liu, and A. Elgammal, “Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9844–9854.

[98] L. Liu, T. Zhou, G. Long, J. Jiang, X. Dong, and C. Zhang, “Isometric propagation network for generalized zero-shot learning,” in *Proceedings of 9th International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=-mWcQVLPSPy

[99] D. Huynh and E. Elhamifar, “Fine-grained generalized zero-shot learning via dense attribute-based attention,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4483–4493.

[100] Z. Ji, Y. Fu, J. Guo, Y. Pang, Z. M. Zhang *et al.*, “Stacked semantics-guided attention model for fine-grained zero-shot learning,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 5998–6007, 2018.

[101] T. Shermin, S. W. Teng, F. Sohel, M. Murshed, and G. Lu, “Integrated generalized zero-shot learning for fine-grained classification,” *Pattern Recognition*, vol. 122, p. 108246, 2022.

[102] L. Bo, Q. Dong, and Z. Hu, “Hardness sampling for self-training based transductive zero-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 499–16 508.

[103] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 3483–3491, 2015.

[104] A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy, "A generative model for zero shot learning using conditional variational autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2018, pp. 2188–2196.

[105] V. K. Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4281–4289.

[106] X. Li, Z. Xu, K. Wei, and C. Deng, "Generalized zero-shot learning via disentangled representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 1966–1974.

[107] N. Bendre, K. Desai, and P. Najafirad, "Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts," in *Proceedings of IEEE International Conference on Image Processing*, 2021, pp. 1284–1288.

[108] M. Gull and O. Arif, "Generalized zero-shot learning using identifiable variational autoencoders," *Expert Systems with Applications*, vol. 191, p. 116268, 2022.

[109] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *Proceedings of 6th International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1kG7GZAW

[110] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5767–5777, 2017.

[111] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7402–7411.

[112] Z. Ye, F. Lyu, L. Li, Q. Fu, J. Ren, and F. Hu, "Sr-gan: Semantic rectifying generative adversarial network for zero-shot learning," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2019, pp. 85–90.

[113] Z. Ye, F. Hu, F. Lyu, L. Li, and K. Huang, "Disentangling semantic-to-visual confusion for zero-shot learning," *IEEE Transactions on Multimedia*, vol. early access, p. doi:10.1109/TMM.2021.3089017, 2021.

[114] H. Huang, C. Wang, P. S. Yu, and C.-D. Wang, "Generative dual adversarial network for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 801–810.

[115] Y. Luo, X. Wang, and F. Pourpanah, "Dual VAEGAN: A generative model for generalized zero-shot learning," *Applied Soft Computing*, vol. 107, p. 107352, 2021.

[116] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 1126–1135.

[117] V. K. Verma, D. Brahma, and P. Rai, "Meta-learning for generalized zero-shot learning," in *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 6062–6069.

[118] V. K. Verma, A. Mishra, A. Pandey, H. A. Murthy, and P. Rai, "Towards zero-shot learning with fewer seen class examples," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2021, pp. 2240–2250.

[119] Z. Liu, Y. Li, L. Yao, X. Wang, and G. Long, "Task aligned generative meta-learning for zero-shot learning," in *Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

[120] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2751–2758.

[121] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1778–1785.

[122] M. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008, pp. 722–729.

[123] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *Science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

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

[125] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, “Zero-shot learning with semantic output codes,” *Advances in Neural Information Processing Systems*, vol. 22, pp. 1410–1418, 2009.

[126] Y. Liu, Q. Gao, J. Li, J. Han, L. Shao *et al.*, “Zero shot learning via low-rank embedded semantic autoencoder.” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 2490–2496.

[127] T. Xu, Y. Zhao, and X. Liu, “Dual generative network with discriminative information for generalized zero-shot learning,” *Complexity*, vol. 2021, pp. 6 656 797:1–6 656 797:11, 2021.

[128] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, “Ridge regression, hubness, and zero-shot learning,” in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2015, pp. 135–151.

[129] G. Dinu, A. Lazaridou, and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” in *Proceedings of 3rd International Conference on Learning Representations, Workshop on Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6568

[130] M. Radovanović, A. Nanopoulos, and M. Ivanović, “On the existence of obstinate results in vector space models,” in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010, pp. 186–193.

[131] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, “Hubs in space: Popular nearest neighbors in high-dimensional data,” *Journal of Machine Learning Research*, vol. 11, no. sept, pp. 2487–2531, 2010.

[132] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[133] C. Luo, Z. Li, K. Huang, J. Feng, and M. Wang, “Zero-shot learning via attribute regression and class prototype rectification,” *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 637–648, 2018.

[134] D. Eagleman, *Incognito: The Secret Lives of the Brain*. Vintage Books, 2013.

[135] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proceedings of 3rd International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[136] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *Computing Research Repository*, vol. abs/1610.05256, 2016. [Online]. Available: http://arxiv.org/abs/1610.05256

[137] Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," *Computing Research Repository*, vol. abs/1611.01599, 2016. [Online]. Available: http://arxiv.org/abs/1611.01599

[138] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of 2nd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[139] Z. Qian, K. Huang, Q.-F. Wang, and X.-Y. Zhang, "A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies," *Computing Research Repository*, vol. abs/2203.14046, 2022. [Online]. Available: http://arxiv.org/abs/2203.14046

[140] C. Lyu, K. Huang, and H.-N. Liang, "A unified gradient regularization family for adversarial examples," in *Proceedings of IEEE International Conference on Data Mining*, 2015, pp. 301–309.

[141] M. Suzuki, Y. Iwasawa, and Y. Matsuo, "Learning shared manifold representation of images and attributes for generalized zero-shot learning," 2018. [Online]. Available: https://openreview.net/pdf?id=Hkesr205t7

[142] Y. Zheng, R. Zhang, and Y. Mao, "Regularizing neural networks via adversarial model perturbation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8156–8165.

[143] R. Keshari, R. Singh, and M. Vatsa, "Generalized zero-shot learning via over-complete distribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 300–13 308.

[144] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang, "Counterfactual zero-shot and open-set visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 404–15 414.

[145] J. Zhang, H. Zhang, and B. Hu, "Dual prototype relaxation for generalized zero shot learning," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2021, pp. 1–6.