On Zero-Shot Recognition of Unseen State-Object Composition

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in Computer Science

by

Aditya Panda

under the supervision of

Prof. Dipti Prasad Mukherjee

Electronics and Communication Sciences Unit
Indian Statistical Institute

September, 2024

To my parents.

Acknowledgements

At the onset, I owe my deepest gratitude to my supervisor Prof. Dipti Prasad Mukherjee, for his invaluable guidance and unwavering support throughout my doctoral study. With his patient supervision, tireless mentorship and depth of knowledge, Prof. Mukherjee has guided me to shape my nascent research ideas into the form of a mature manuscript. While always encouraging me to think independently, he with his incisive comments, has shaped my skills as a junior researcher. I shall continue to hold the values and the morales he instilled in me, throughout my career.

I am also grateful to all the other faculty members of the Electronics and Communication Sciences Unit (ECSU) as I was blessed to have insightful academic feedback through different academic interactions with them. I am also thankful to the non-teaching staff members of ECSU for their support in various forms.

I extend my gratitude to Dr. Partha Pratim Mohanta for his support. I am indebted to Dr. Bikash Santra who collaborated with me for a couple of publications. The clarity of vision of Dr. Santra has helped me to clear my doubts in different stages of our work. I am also grateful to my lab-mates Sankarsan Seal, Dr. Saikat Sarkar and Suman Ghosh for the cooperative and cordial relationship we shared.

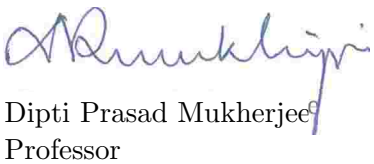
Finally, I shall like to thank my parents, who, all along have been a source of steadfast support and inspiration throughout the thick and thin of my journey towards my doctoral degree.

Aditya Panda,
Senior Research Fellow.

Indian Statistical Institute,
Kolkata.

Certificate

This is to certify that the thesis entitled "On Zero-Shot Recognition of Unseen State-Object Composition" submitted by Aditya Panda to Indian Statistical Institute, in partial fulfillment for the award of the Ph.D degree in Computer Science is a bonafide record of work carried out by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of this institute.



Dipti Prasad Mukherjee
Professor
Electronics and Communication Sciences Unit
Indian Statistical Institute

Abstract

Compositional Zero-Shot Learning (CZSL) attempts to recognise images of new (unseen) compositions of states and objects, when images of only a subset of state-object compositions are available as training data. Thus a CZSL model should recognise a *young dog* when the model has seen images of the state-object compositions *young bear*, *old bear* and *old dog*. There are multiple challenges to solve the CZSL problem. It is difficult to disentangle the visual features of object *dog* and its state *young* from its compositional image *young dog*. The features of a state are observed to have high variation in visual features across compositions. For example, the state *sliced* has different visual features in compositions *sliced apple* and *sliced tomato*. In the second chapter of the thesis, we attempt to disentangle the visual features of state and object using a two-stage sequential recognition approach. In next chapter of the thesis, we work on the open-world CZSL problem where no prior information about the feasibility of a state-object composition is available. We use a Graph Convolutional Network based architecture along with a frequency-based feasibility prediction approach for the open-world CZSL problem. Another challenge in CZSL lies in the fact that the extent of association between the features of a state and an object vary significantly in different images of the same composition. For example, in different images of *peeled orange*, the *oranges* may be *peeled* to a different extent. Thus the visual features of images of *peeled orange* may vary. In the fourth chapter, a novel Knowledge-guided Transformer Network is proposed to better process the partial association between the visual features of state and object. In the fifth chapter, we attempt the partially supervised CZSL (pCZSL) problem, where for each state-object compositional image, either the state or the object annotation is available. We propose a novel vision transformer based architecture with Locality Preserving Neighbourhood Aggregation approach in the fifth chapter. Effective identification of the discriminative features of state and object often depends on the scale of the object in the image. For example, in the images of the two compositions, *young bear* and *old bear*, the identification of the states *young* and *old* may depend on recognising the scale (or size) of the object *bear* in the image. In the sixth chapter, we leverage Vision Language Model (VLM) to estimate the scale-aware features in CZSL. Extensive experiments on C-GQA, MIT-States and UT-Zappos50k datasets demonstrate the effectiveness of the approaches in this thesis, when compared to the state-of-the-art in the closed-world CZSL, open-world CZSL and pCZSL settings. As concluding remarks, we discuss the future scope of research in CZSL.

Contents

List of Tables

List of Notations

S	Set of the states in the CZSL dataset
O	Set of the objects in the CZSL dataset
N_s	Number of states in the CZSL dataset
N_o	Number of objects in the CZSL dataset
N_t	Number of images in the training set
T	The training set of the CZSL dataset with images and the corresponding state-object compositional labels
$\mathcal{C}$	Set of all possible compositional state-object classes
$\mathcal{C}_{seen}$	The set of all *seen* state-object compositions or the set of state-object compositions for which images are present in the training set
$\mathcal{C}_{unseen}$	The set of all *unseen* state-object compositions or the set of state-object compositions for which images are only present in the test set
$H(p)$	Entropy of the probability distribution, p
$MI(\cdot)$	Mutual information
$\mathcal{M}$	Feasibility Matrix, predicting the feasibility of a particular state-object composition
$\mathcal{A}(\cdot)$	Temperature based annealing approach
$\hat{g}_s, \hat{g}_o$	Pseudo state and object tokens, respectively
Q_l, K_l and V_l	Query, key and value matrices in layer l of the transformer encoder.
E_{n_c}	Effective number of samples for compositional class c
$\mathcal{G}_{weak}(\cdot)$	Weak augmentation applied on the input image
$\mathcal{G}_{strong}(\cdot)$	Strong augmentation applied on the input image
$\hat{\mathcal{R}}(h)$	The *empirical risk* of the hypothesis h
$\hat{\mathfrak{R}}_{\mathrm{T}}(\cdot)$	The Rademacher complexity defined on the training set T

Chapter 1

Introduction

1.1 Compositional Zero-Shot Learning

A machine learning model typically recognizes patterns grouped into a number of (different) classes. A Zero-Shot Learning (ZSL) system recognizes a class that has not been seen during training. In the context of computer vision, a ZSL system (Larochelle et al., 2008, Lampert et al., 2009, Palatucci et al., 2009, Farhadi et al., 2009, Xian et al., 2017) is expected to recognize an object which is previously unseen by the ZSL system during its training. In Compositional Zero-Shot Learning (CZSL) (Nan et al., 2019, Xu et al., 2021b, Karthik et al., 2022) the input image is annotated by a label which is a composition of the name of the object and its state (for example, *cloudy sky*). CZSL should seamlessly combine familiar concepts like the *yellow* of a *yellow tulip* and *tiger* from a *white tiger* to identify novel compositions such as *yellow tiger*. This innate ability to recognise compositions from finite pre-conceived knowledge, without seeing every conceivable compositions is a defining trait of human intellect (Allen et al., 2020, Lake et al., 2017, Marcus, 2003). The ability to learn primitive concepts, their compositions and re-utilization of those learned concepts to recognise unseen compositions is broadly referred to as *compositional generalization* (Atzmon et al., 2016). CZSL requires learning systems to perform *compositional generalization*. In other words, CZSL requires to recognise unseen (or *new*) compositions of known objects and known states. Thus, an image classification model for CZSL problem may be trained on annotated images of *ripe orange*, *ripe apple* and *peeled orange* (see Fig. 1.1). The trained model is expected to recognise images of *peeled apple* during test time. There are multiple challenges in the CZSL problem, as discussed next.

1.2 Challenges

There are four major challenges in CZSL, as described next.

i) *Isolation of state and object features:* In a state-object composition, isolation of the features of state and features of object from the input image is an important requirement. The isolation (or disentanglement) of the state and object features helps to identify unseen state-object compositions during test. Due to entanglement between the visual features of state and object features, it is always challenging to effectively separate the corresponding visual features.

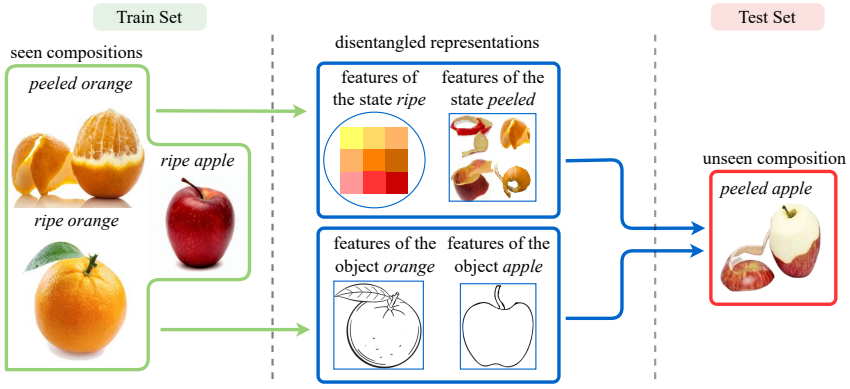


Figure 1.1: An example of CZSL: On being trained with *peeled orange*, *ripe orange* and *ripe apple*, an algorithm for CZSL is expected to learn the features of *ripe*, *apple*, *orange* and *peeled*. The trained features of states and objects are utilised by the CZSL algorithm to recognize unseen image of *peeled apple*.

ii) *Intra-class variation in the visual features of state:* In a state-object composition, object represents a physical entity. On the other hand, the state represents a collection of attributes (like color, texture etc.) of the object. The visual features of the state vary widely based on the context i.e. the particular state-object composition. For example, the state *ripe* has different features in compositions *ripe apple* and *ripe banana*. This *intra-class variation* in visual features of the state *ripe* creates ambiguity for the model to learn unique features for the state *ripe*. Hence, effective processing of the contextual relationship is a necessary requirement to solve the CZSL problem.

iii) *Partial association between visual features of state and object:* Third major challenge is observed in understanding the extent of association between the visual features of the primitives (state and object). For example, for different images of *peeled orange*, an *orange* may be *peeled* to different extents. Based on the extent to which the *orange* is *peeled*, the visual features of images of *peeled orange* vary.

iv) *Extracting the scale-aware visual features:* The fourth notable challenge in CZSL is that the discriminative features of state and object often depend on the scale of the object in the image. For example, in the images of two compositions, *young bear* and *old bear* the identification of the states *young* and *old* may depend on recognising the scale (or size) of an object *bear* in the image. Next, we discuss the CZSL problem from the perspective of data efficient learning approach.

1.3 CZSL: A Data-efficient Learning Approach

Most of the objects in real-life are compositions of different states. The number of images available in a dataset for state-object compositional classes often follow a long-tailed distribution (Salakhutdinov et al., 2011, Wang et al., 2017). It is always difficult to gather annotated images for all possible states of an object.

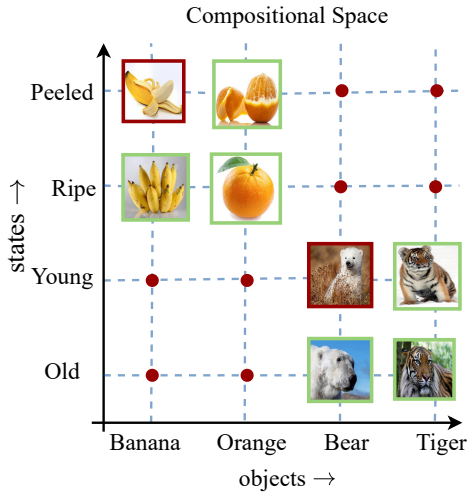


Figure 1.2: CZSL is a data efficient learning approach where only a subset of state-object compositional images are required to train the model. In this figure, the images inside the green coloured squares represent the images from the training set and the images inside the red coloured squares represent images from the test set. The red dot in the figure represents that the corresponding state-object composition is infeasible and thus the images corresponding to that compositional class does not exist.

A solution for CZSL requires annotated training images of only a subset of possible state-object compositional classes and can recognise a lot of new compositional classes during test. As shown in Fig. 1.2, the training set in a CZSL dataset may contain images of the compositional classes *young tiger*, *old tiger* and *old bear*. A CZSL approach should be able to recognise images of the unseen class *young bear* in addition to the aforementioned compositional classes in the training set. Thus a solution for CZSL is a data-efficient approach for state-object compositional image classification problem. The partially supervised CZSL problem requires even less annotation as discussed next.

1.3.1 Partially Supervised CZSL

In partially supervised Compositional Zero-Shot Learning (pCZSL) (Karthik et al., 2021), only the state annotation or the object annotation is available during training for each image. For example, corresponding to the image of *young bear*, only *young* annotation may be available in the training set of pCZSL (see Fig. 1.3). The pCZSL is a more realistic problem than the existing CZSL. This is due to the fact that pCZSL requires even less amount of labelled data than CZSL. Next, we discuss the objectives of the thesis.

1.4 Objectives of the Thesis

CZSL is a challenging problem in the paradigm of data-efficient learning approaches. In this thesis, we work towards addressing the challenges inherent to the CZSL problem. The objectives of this thesis are summarised as follows.

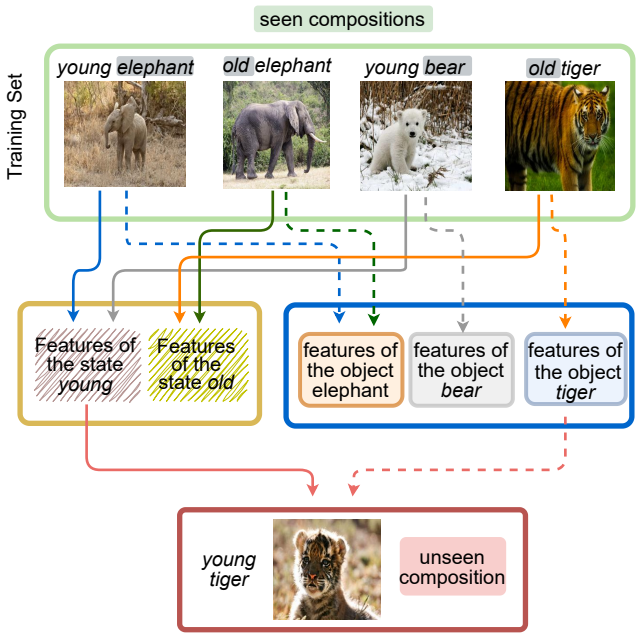


Figure 1.3: Example of partially supervised CZSL (pCZSL): The training set consists of images of *young elephant*, *old elephant*, *young bear* and *old tiger*. The test set contains images of the *unseen* composition *young tiger*. In pCZSL, either the state annotation or the object annotation is only available during training for each image. For example, corresponding to the image of *young elephant*, only *young* annotation is available in the training set of pCZSL. The labels inside the gray coloured boxes represent unavailability of the corresponding labels during the training phase in pCZSL.

i) As referred in Section 1.2, the disentanglement of state and object features is a key challenge in CZSL. So, we attempt to address the disentanglement using a sequential recognition approach.

ii) The conventional approaches for CZSL require the list of feasible state and object list even for unseen compositions. This as a drawback in proposing a more generalised solution for CZSL. Hence we work on the open-world CZSL (OW-CZSL) (Mancini et al., 2021) problem where the proposed approach do not require any information regarding feasibility for unseen compositions. More specifically, proposed approach considers all possible state-object compositions as a possible model output.

iii) Another objective of the thesis is to effectively address the partial association of state and the object features in CZSL. Existing methodologies often overlook the possibility of partial associations, focusing solely on the presence or absence of these features. Our approach however, not only considers the existence of state and object features but also the extent of their association.

iv) Although the CZSL problem is a data-efficient learning paradigm, still it requires state and object annotations for each state-object composition in the train set. In this thesis, we work on the pCZSL problem, where for each image either state or the object annotation is only available.

v) We also attempt to extract the scale-aware features in CZSL. In CZSL, in the images of two compositions *young bear* and *old bear*, the identification of the states *young* and *old* depends on recognising the scale (or size) of the object *bear* in the image. We attempt to address this issue by proposing scale-aware feature extraction approach.

Next we discuss the contributions of the thesis.

1.5 Contributions

There are five contributory chapters in our thesis, starting from the second chapter. The contributions of these five chapters are discussed next.

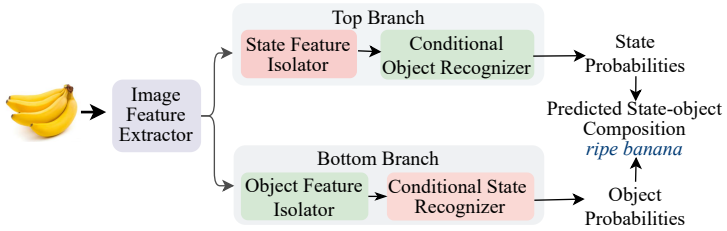


Figure 1.4: Block level representation of the proposed approach in Chapter 2.

Contributions of the second chapter: As already discussed, a model trained for CZSL should be able to recognise unseen compositions during test. To recognise unseen compositions, a model should be able to disentangle the features of constituent primitives. The disentangled features of state and object are then matched for compatibility with the features of the test image to predict the unseen compositional class. However the state and the object features are entangled in the visual features of the compositional image. Thus it is difficult to effectively isolate the features of the state and the object. We propose a two stage sequential recognition approach for better disentanglement of state and object features (Panda et al., 2023). However, the two stage sequential recognition approach often suffers from the drawback that the errors from the first stage affects the classification results of the second stage. To circumvent this issue, we propose a complementary branch approach. In our approach, there are two branches, each performing sequential recognition of state and object. The two branches are referred to as complementary branches as the first branch recognises the state first and followed by the object, the second branch recognises the object first, followed by the state.

For a CZSL dataset with α number of objects and β number of states, the number of possible state-object compositional classes are $\mathcal{O}(\alpha\beta)$. Hence, simultaneously recognising state and object requires identifying the correct state and object from a search space of dimension $\mathcal{O}(\alpha\beta)$. Whereas, in case of sequential approach, we recognize one visual primitive (state or object) at a time. As a result, the search space is of complexity $\mathcal{O}(\alpha+\beta)$. Assuming comparable number of states and objects in a dataset, i.e. $\alpha \approx \beta$, the sequential recognition approach has $\mathcal{O}(\alpha)$ search space. In comparison the simultaneous recognition of object and state has $\mathcal{O}(\alpha^2)$ search space. Thus proposed complementary branch sequential recognition approach has linear search space complexity for the CZSL problem (see Fig. 1.4).

Next, the visual features of state and the object in a state-object composition may be decomposed into three components, the unique visual features of state, the unique visual features of the object and the visual features of the state-object composition. For effective disentanglement of state and object features, we suppress the third component of features by proposing a gradient based penalization component in the final loss. Existing approaches (Purushwalkam et al., 2019, Nagarajan and Grauman, 2018, Misra et al., 2017, Xu et al., 2021b, Naeem et al., 2021, Nayak et al., 2022) have used simultaneous state-object recognition approach and thus suffer from quadratic search space complexity. Some approaches like (Saini et al., 2022, Li et al., 2022, Karthik et al., 2022) have used separate classifiers for state and object, respectively. However, these approaches fail to suppress the state-object joint features, as proposed using the gradient penalization loss our approach.

In CZSL state-object compositional image, the state and the object represent two distinct types of constituents. Thus for effective disentanglement of the state and object features from a compositional image, the isolated features of state and object should have minimum overlap between them. Besides, the disentangled state and object features together should be as much informative as possible with respect to the visual features of the input image (Do and Tran, 2019, Eastwood, Cian and Williams, Christopher KI, 2018). In Chapter 2, we have reported a lower bound of the mutual information between the isolated state and the isolated object features. Further details are discussed in Section 2.3.4, proposition 1.

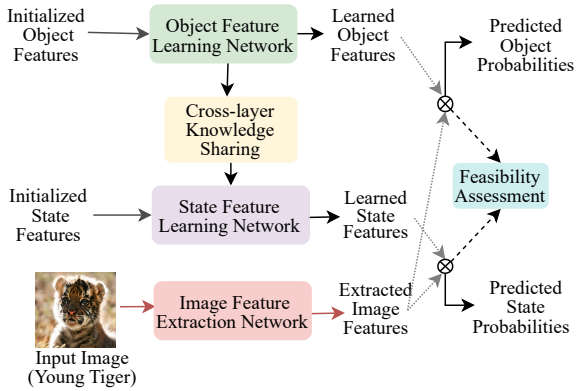


Figure 1.5: A simplified block diagram of the approach in Chapter 3 is shown.

Contributions of the third chapter: It can be observed that a group of similar objects like *apple*, *banana* or *orange* create compositions with a select group of states (for example, *rotten*, *ripe* or *peeled*). If we share the information that *apple* and *banana* belong to the same group in the context of state *peeled*, it may help to disambiguate the differences in visual features of *peeled* in the compositions *peeled banana* and *peeled apple*. This should help an algorithm for CZSL to identify the unique representation for the state *peeled*. Inspired by this observation we propose a two branch Graph Convolution Network (GCN) in the third chapter of the thesis. In the GCN in the first branch, there is one node for each state in the dataset. Similarly, in the GCN in the other branch, there is a node representing each object in the dataset. In the GCN in both the branches, we have kept the adjacency information among the state notes and among the object nodes learnable. Besides, we propose a cross layer knowledge sharing approach to share information among the state nodes in the state

branch to the object nodes in the other branch (see Fig. 1.5). We also work on the OW-CZSL (Mancini et al., 2021) problem. In OW-CZSL, no prior information about the feasibility of the compositions are allowed to be used during training. Here we propose a novel frequency based approach to estimate the feasibility of the unseen state-object compositions (Panda and Mukherjee, 2024a). Our approach is better than other GCN based CZSL approaches (Ruis et al., 2021, Naeem et al., 2021) in two aspects. First, they have not attempted to address the OW-CZSL problem. Besides, they also have used a fixed adjacency based approach in the GCN. We observe that a fixed adjacency based approach fails to learn the similarity in visual features of similar states and similar objects, respectively.

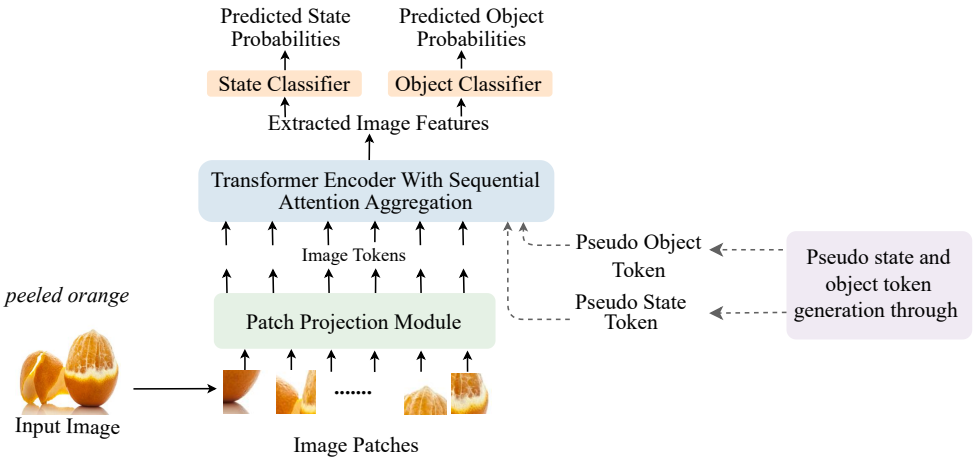


Figure 1.6: Block diagram of the proposed approach in Chapter 4.

Contributions of the fourth chapter: A notable challenge in CZSL lies in the fact that the extent of association between the features of a state and an object vary significantly in different images of same composition. As reported in Section 1.2, in different images of *peeled orange*, the *oranges* may be *peeled* to different extents. As a consequence, the visual features of images of the class *peeled orange* may vary. Hence, there exists significant amount of intra-class variability among the visual features of different images of a composition. Existing approaches (Misra et al., 2017, Nagarajan and Grauman, 2018, Nan et al., 2019, Mancini et al., 2021) ignore the possibility of partial association between state and object features in a compositional image and only look for existence or absence of the features of a state or object in a composition. Hence, these approaches fail to tackle the significant amount of intra-class variability among the visual features of multiple images of a composition. On the contrary, our approach not only looks for the existence of the state features (and object features) but also the extent of association between state and object features. In the chapter 4, we propose a novel Knowledge guided Transformer Network (Panda and Mukherjee, 2024b) to model the partial association between the visual features of state and object.

We attempt to model the partial association between the features of state and object in a compositional image. For each state-object compositional image, a set of scalar values $\in [0, 1]$ are learnt. Each learnt scalar value quantifies the association of features of a particular state or a particular object in the compositional image features. We derive a mutual information based regularisation loss to learn the partial association between state and object features

in an unsupervised approach. The detailed approach is reported in Section 4.3.4. Besides to better process the image features through multiple layers of the *transformer encoder*, we also propose a Sequential Attention Aggregation approach (see Fig. 1.6).

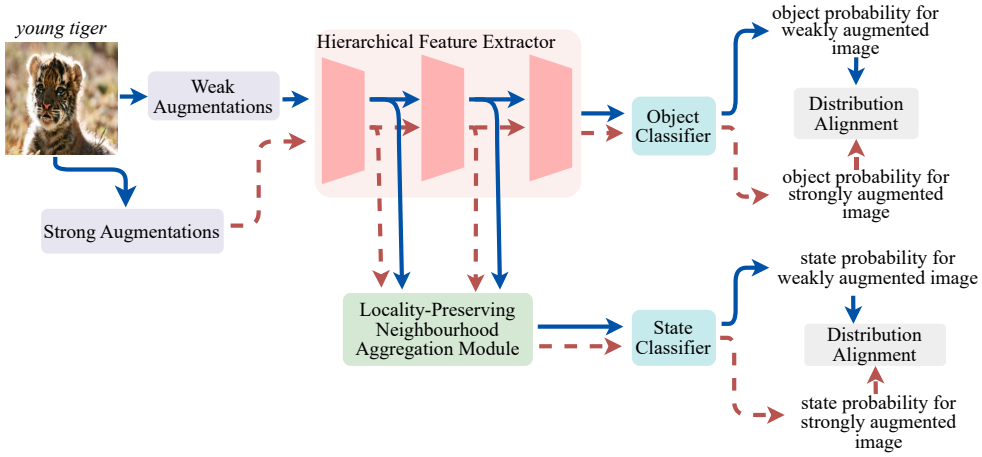


Figure 1.7: Block-level representation of our approach in Chapter 5.

Contributions of the fifth chapter: In the chapter 5 of the thesis, we work on the partially supervised CZSL (pCZSL) problem. In this work, we propose a novel architecture using a transformer based Shared Feature Extractor (SFE) and a Locality-Preserving Neighbourhood Aggregation (LoPNA) module. The SFE can better process the large range of semantic interpretation of state and object features compared to the existing convolutional network based feature extractors. So the SFE is utilised to process the contextual dependency that exists between the features of state and object. Proposed LoPNA utilises features from the intermediate stages of the SFE to better understand the features of object at their corresponding scale. To leverage the partially labelled data in pCZSL, we utilise a novel α divergence-based *distribution alignment* strategy. We pass a *strongly* perturbed (augmented) version of the input image along with the *weakly* augmented version of the input image to the proposed architecture. Next, we encourage the predicted class probability distributions for the *weakly* and *strongly* augmented images to be as close as possible using a Class-balanced Confidence-scaled Distribution Alignment (CCDA) loss (see Fig. 1.7). Proposed CCDA loss incorporates class specific loss re-weighting approach to alleviate the effect of data imbalance issue in CZSL datasets. Recently, KG-SP (Karthik et al., 2022) and Pro-CC (Huo et al., 2024) attempt to address the pCZSL problem. KG-SP utilised external knowledge (ConceptNet (Speer et al., 2017)) to estimate the feasibility of unseen state-object composition. Our approach on the contrary, does not require any external knowledge during training or inference stage. In Pro-CC (Huo et al., 2024), the intermediate response from the Object Classifier is added with the intermediate response in the State Classifier, in an attempt to process the context dependency between the state and object features. The classifiers in Pro-CC use the MLPs and are less capable to process the local structures in the image features. We argue that the context dependency between state and object visual features can be more effectively processed through sharing the feature response from intermediate layers of the image feature extractors. Besides, the existing CZSL algorithms (Karthik et al., 2022, Huo et al., 2024,

Naeem et al., 2021, Purushwalkam et al., 2019, Nagarajan and Grauman, 2018) have not attempted to address the imbalance problem in the benchmark CZSL datasets (Isola et al., 2015, Yu and Grauman, 2014, 2017, Naeem et al., 2021).

In the pCZSL problem, due to partial annotation, there is a large amount of unlabeled data. To leverage the unlabeled data, we pass a strongly and a weakly augmented versions of the input image to the proposed model. Next, we use a *distribution alignment* loss to obtain discriminative features of state and the object. The *distribution alignment* loss is specifically designed for the class-imbalanced CZSL datasets, where higher loss contributions from *minority classes* (classes with less number of images in the training set) are incorporated. We theoretically show the advantages of the proposed loss with respect to the usual KL-divergence loss. For further details, see proposition 1 in Section 5.3.2.3.

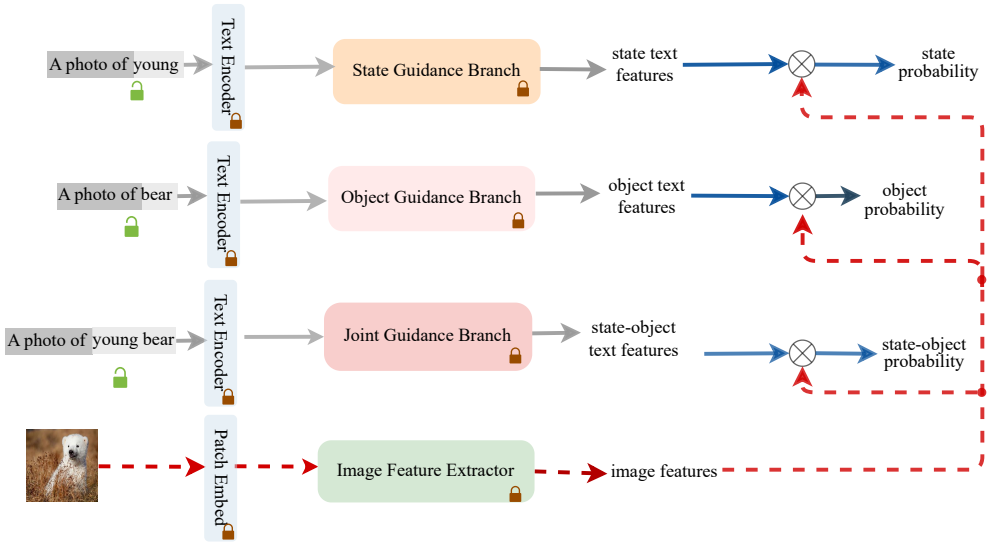


Figure 1.8: Block-level diagram of the proposed architecture in Chapter 6.

Contributions of the sixth chapter: We proposed a novel Vision Language Model (VLM) (Radford et al., 2021) based CZSL approach, as discussed next. To aid the disentanglement process in CZSL, three Guidance Branches (GBs) are proposed to extract features of state, object and the state-object composition (Panda and Mukherjee, 2024c). Besides, we propose Knowledge Coupling Module (KCM) to integrate the knowledge from the GBs to the image feature extractor of the proposed approach. The KCMs help to process the contextual dependency between the visual features of state and object through cross-modal interaction. We propose a novel variational prompt learning approach to incorporate diversity in textual features extracted from prompts and to better adapt to variations in image features. Extensive experiments on three benchmark CZSL datasets demonstrate the effectiveness of our approach in both open-world and closed-world CZSL evaluation protocols. The improvements of our approach over the aforementioned approaches (Xu et al., 2022, Lu et al., 2023, Nayak et al., 2022) are two-fold. First, we use three independent prompts and corresponding *text feature extractors* for effective disentanglement of state and object features from the features of the compositional image. Besides we incorporate cross-modal knowledge sharing through the proposed Knowledge Coupling Modules (KCMs), to better process the contextual dependency between state-object features (see Fig. 1.8). Three recently proposed

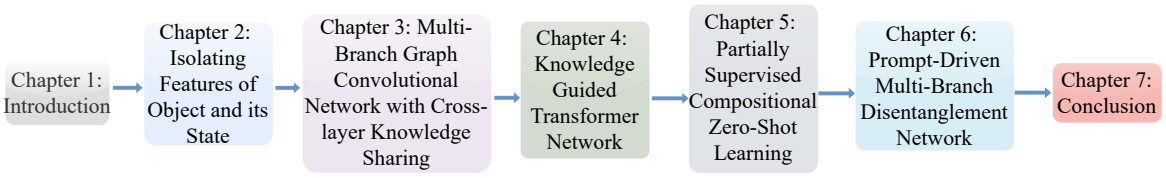


Figure 1.9: Chapter outline of the thesis.

works (Huang et al., 2023, Wang et al., 2023a, Li et al., 2024), also use multiple prompts for CZSL. However, aforementioned three CZSL approaches suffer from different weaknesses over our approach. In Troika Huang et al. (2023), intermediate layer response from the text feature extractor are share to the image feature extractor. On the contrary, in our approach using KCMs, we share knowledge from the *image feature extractor* to guide the text features. Hence, the approach in Troika (Huang et al., 2023) is less effective for CZSL problem than our approach in processing the intra-class feature variation in CZSL. Besides proposed multi-scale feature extraction approach in this work helps to better solve scale-aware feature aggregation problem. The works in (Wang et al., 2023a, Li et al., 2024) are less effective to process intra-class variation in image features than our approach due to absence of knowledge sharing between the *text* and *image feature extractors*.

In the sixth chapter, we use a Rademacher complexity (Bartlett and Mendelson, 2002) analysis to theoretically show that the risk associated in the state-object compositional class prediction is upper-bounded by the linear sum of the risk associated with the individual state and object predictions (see proposition 1 in Section 6.3.1.5). We also provide a novel variational inference based approach for estimating the contextual dependency in state-object compositions. Next we discuss the organisation of the thesis.

1.6 Organisation of the Thesis

This thesis is divided in seven chapters. The first chapter introduces the CZSL problem and summarises the contribution of the thesis. The problem specific reviews of the relevant approaches are reported in each of the next five chapters. The overall organisation of the thesis is represented in Fig. 1.9.

In chapter 2, we mainly focus on the disentanglement of the state and the object features from the visual features of state-object compositional image. For effective disentanglement of state and object features, we use two branch architecture in (Panda et al., 2022). Next we extend this work and propose a sequential recognition approach for effective disentanglement of the state and the object features (Panda and Mukherjee, 2023).

In chapter 3 of this thesis, we work on the OW-CZSL problem. In OW-CZSL, no prior information about the feasibility of the compositions are allowed to be used during training. Here we propose a novel frequency based approach to estimate the feasibility of the unseen state-object compositions (Panda and Mukherjee, 2024a). Besides we also propose a GCN based architecture for the OW-CZSL problem.

In chapter 4, we attempt another important challenge in Compositional Zero-Shot Learning (CZSL), the considerable variability in the association between the features of a state and an object across different images of the same composition. We introduce a novel Knowledge-Guided Transformer Network designed to model the partial association between the visual

features of state and object (Panda and Mukherjee, 2024b).

In chapter 5 of the thesis, we work on the partially supervised CZSL (pCZSL) problem. To better process the cross-scale features, we propose a novel multi-scale feature aggregation approach using a Locality Preserving Neighbourhood Aggregation approach. We also propose an α-divergence based distribution alignment loss to leverage the partially labelled data in pCZSL. Besides, we utilise a class-balanced scaling approach to address the data-imbalance issue prevalent in CZSL datasets.

In the sixth chapter of the thesis, we leverage the large scale pre-training of VLMs for the CZSL problem. Specifically we propose a multi scale feature aggregation approach and diversity preserving feature regularisation using variational inference approach (Panda and Mukherjee, 2024c).

Finally, in chapter 7, we summarise the contributions made by the previous chapters and point to the future direction of the research. We also report the new challenges that originated due to the work in this thesis.

Chapter 2

Isolating Features of Object and its State Using Sequential Approach

2.1 Introduction

A model trained for CZSL should be able to recognise unseen compositions during test. To recognise unseen compositions, a model should be able to deftly disentangle the features of constituent primitives. The disentangled features of state and object are then matched for compatibility with the features of the test image to predict the unseen compositional class. However the state and the object features are entangled in the visual features of the compositional image. Thus it is challenging to isolate the visual features of the state and the object. We propose a two-stage sequential recognition approach for better disentanglement of state and object features (Panda et al., 2023), as described next.[1]

Proposed model consists of two distinct branches: the *top* and the *bottom branches* as shown in Fig. 2.1. The *top branch* consists of two cascaded stages. The *top branch* takes the image features as input and performs sequential state and object recognition. Next, the second stage of the *top branch* recognizes the object given the state as determined in the first stage of the *top branch*. The *bottom branch* of the proposed architecture, as illustrated in Fig. 2.1 first recognizes the object present in the input image. Subsequently, the second stage predicts the state for the object recognized in the first stage. As both the branches of the proposed architecture sequentially recognize the state and the object, one after another, we refer our model as Sequential Compositional Learning Network (SeCoNet). Next we briefly justify the advantage of the sequential architecture.

As discussed in Section 1.5, for a CZSL dataset with α number of objects and β number of states, the number of possible state-object compositional classes are $\mathcal{O}(\alpha\beta)$. Hence, jointly recognising state and object requires identifying the correct state and object from a search

[1]Parts of the work done in this chapter is published as follows,

1. Aditya Panda, Bikash Santra and Dipti Prasad Mukherjee, "Bi-Modal Compositional Network For Feature Disentanglement", Published at IEEE International Conference on Image Processing (ICIP), 2022, pp. 3051-3055.
2. Aditya Panda, Bikash Santra and Dipti Prasad Mukherjee, "Isolating State and Object Features for Compositional Zero-shot Learning", Published in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7 (5), October 2023.

space of dimension $\mathcal{O}(\alpha\beta)$. In case of sequential approach, we recognize one visual primitive (state or object) at a time. As a result, the search space is reduced to $\mathcal{O}(\alpha + \beta)$. Thus we provide a two stage sequential learning approach with linear search space complexity for the CZSL problem.

Next, we discuss the advantages of two branch architecture. Let us consider an input image where the recognition of the state is more difficult than the recognition of the object. For example, in the case of the compositions *ripe apple*, *ripe banana* and *ripe pineapple*, the objects *apple*, *banana* and *pineapple* are easier to recognize. However the state *ripe* has distinct visual representations across different objects. In the *top branch*, if state recognition in the first stage is incorrect, the error will be propagated to the second stage. As a result, the corresponding object recognition in the second stage will be incorrect. However, in the *bottom branch*, the object is recognized first. If the object is recognized correctly then it will reduce the model's confusion in recognizing state in the second stage.

To help the object and the state recognition process, we propose a set of Weight Estimators (WEs). The WE takes the word embedding (Pennington et al., 2014, Bojanowski et al., 2017) of the object and state labels of an image as input and estimates a set of weights. The weight vector, as obtained from the WE, is multiplied with the intermediate features of the two stages of the *top* and *bottom branches*. Since the CZSL involves recognition of unseen state and object composition, we intend to use features from textual modality (by using word embedding of state and object labels) for training of our SeCoNet. The purpose of the WEs in the proposed architecture is to help SeCoNet in integrating the features from two distinct modalities of input data, the image and the text. The proposed SeCoNet is trained by optimizing a gradient based regularization loss component for better disentanglement of states and objects from the state-object compositions. The key contributions of the proposed work with respect to state-of-the-art approaches are four-fold:

(a) The proposed sequential approach SeCoNet has reduced the search space to $\mathcal{O}(\alpha)$ from $\mathcal{O}(\alpha^2)$.
(b) To avoid the limitations of a sequential approach, we build a two-branch network in addressing the CZSL problem.
(c) In Section 2.3.3.1, we propose a novel gradient regularization based loss component for isolating the object and state features from the visual cues of the input image.
(d) We theoretically analyse the capacity of the proposed network in disentanglement of state and object features as shown in Section 2.3.4.

Remainder of the chapter is organized as follows. Section 2.2 reviews the related works. Proposed architecture is explained in Section 2.3. Section 2.4 presents the experiments and their results. Finally, Section 2.5 summarises the chapter.

2.2 Works Related to Disentanglement of Visual Features in CZSL

Misra et al. (2017) train a binary SVM classifier for each objects and states in the dataset. The weights of SVM for each object and state are utilized to predict the weights of the SVM classifier for the state-object composite class. However training one SVM for each object and state is computationally expensive. Nagarajan *et al.* (Nagarajan and Grauman, 2018) represent the state feature as a linear transformation on the object feature. However, linear transformations are ineffective to represent the intricate interactions between visual

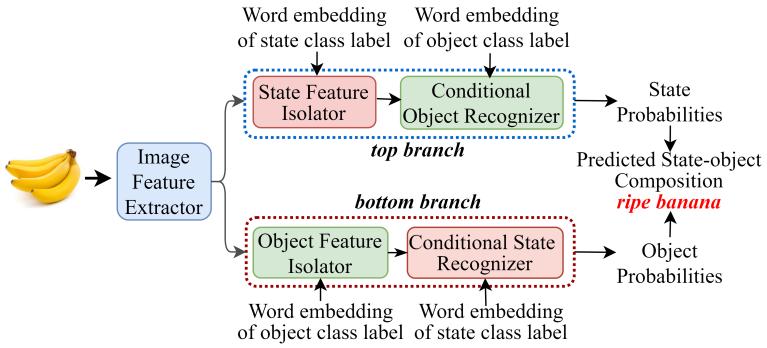


Figure 2.1: Block level representation of the proposed Sequential Compositional Learning Network (SeCoNet).

features of state and object. Wei et al. (2019) use Wasserstein GAN (WGAN) (Arjovsky et al., 2017) to generate the features of the images for unseen state-object compositions from the word embedding of class labels. However, word embedding and features of the images are sampled from two distinct distributions. Hence, the WGAN struggles to adapt between two different domains. Wang et al. (2019b) propose a model consisting of a meta-learner and a predictor network. The meta-learner takes the word embedding (Pennington et al., 2014) of object and state labels as input, and generates parameters (or weights) for modules in the predictor network. Finally, the predictor network predicts the final state-object labels of unseen images. However, in their approach, the weights of the main predictor network are not only learned with the back-propagation of errors but also predicted by the meta-learner network. Thus, this strategy of weight learning from two distinct sources affects the convergence of the predictor network and subsequently the end result. In (Nan et al., 2019, Xu et al., 2021b), the visual features of the state-object composition is extracted from the input image. Also the features of the state and object are separately extracted from the word embedding of the state and object labels, respectively. The final prediction is performed evaluating the similarity between extracted visual features and word features in the shared latent feature space. Li et al. (2020) propose a solution for CZSL using a group theoretic analogy that assumes state as a symmetric operator in object space. Atzmon et al. (2020a) present a causal view based approach for the CZSL problem. Mancini *et al.* (Mancini et al., 2021), measure cosine similarity between the word embedding of state and object class labels. The cosine similarity values are then utilized to predict the feasibility of unseen state-object compositions. A Graph Convolution Network (Kipf and Welling, 2016) based approach for solving the CZSL problem is proposed in (Naeem et al., 2021). The approach by Karthik et al. (2022) utilises external knowledge (Speer et al., 2017) to predict feasibility of unseen state-object compositional classes.

Purushwalkam et al. (2019) train a network composed of fully connected modules jointly with a gating network. Gating network multiplies trainable weights with the intermediate output of each module of fully connected layer. However, the network in (Purushwalkam et al., 2019) uses one single branch in their architecture to simultaneously predict the state and object in an unseen image. Similarly, other existing state-of-the-art algorithms (Atzmon and Chechik, 2018, Nagarajan and Grauman, 2018, Naeem et al., 2021) also perform simultaneous recognition of objects and states. As discussed in 2.1, the simultaneous state and

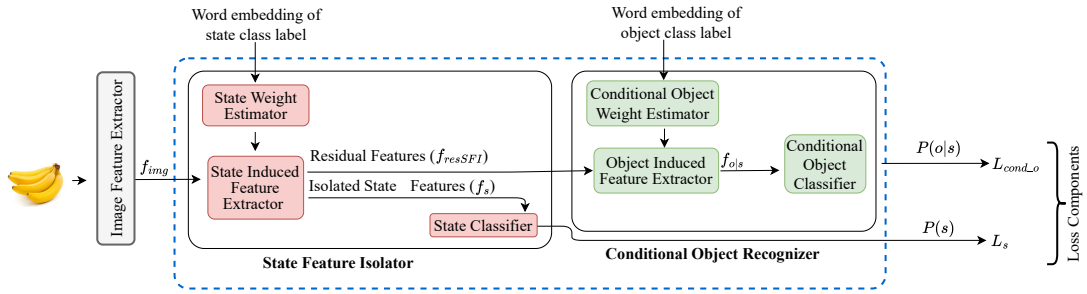


Figure 2.2: Flowchart of the *top branch* of SeCoNet

object recognition has quadratic computational cost wheres the sequential state and object recognition approach followed in our work has linear computational cost. Next we introduce the methodology of our approach.

2.3 Methodology

Before reporting the proposed approach, we briefly report the CZSL problem statement.

2.3.1 Problem Statement

In a CZSL dataset the set of states is represented as $S = \{s_1, s_2, \dots s_{N_s}\}$. Here N_s represents the number of states in the dataset. Similarly, the set of objects is represented as $O = \{o_1, o_2, \dots o_{N_o}\}$, N_o denoting the number of objects in the dataset. The set of all possible state-object compositions are represented as $\mathcal{C} = S \times O$. The training set is represented as $T = \{(I_1, c_1), \dots (I_{N_t}, c_{N_t})\}$. Here i^{th} image from the training set and corresponding state-object compositional label are represented as I_i and c_i, respectively. N_t represents the number of training images in the dataset. Also $c_k = (s_i, o_j)$ with s_i and o_j representing corresponding state and object labels, respectively ($i \in \{1, 2, \dots N_s\}$, $j \in \{1, 2, \dots N_o\}$, $k \in \{1, 2, \dots |\mathcal{C}|\}$). The set of all possible state-object compositions, $\mathcal{C}$ is split into two sets, the set of *seen* compositions, $\mathcal{C}_{seen}$ and the set of *unseen* compositions $\mathcal{C}_{unseen}$. Here $\mathcal{C}_{seen}, \mathcal{C}_{unseen} \subset \mathcal{C}$ and $\mathcal{C}_{seen} \cap \mathcal{C}_{unseen} = \phi$. For closed-world CZSL (CW-CZSL) evaluation proposed by (Purushwalkam et al., 2019), a predefined subset of $\mathcal{C}_{unseen}$, is only present in the test set. Thus for CW-CZSL, $\mathcal{C}_{test}^{CW-CZSL} \subset \mathcal{C}_{seen} \cup \mathcal{C}_{unseen}$.

2.3.2 SeCoNet: CZSL through Sequential Classification

The proposed SeCoNet, has an Image Feature Extractor block and two primary branches: *top branch* and *bottom branch* as shown in Fig. 2.1. Next, we describe the working principle of each building block of SeCoNet.

2.3.2.1 Image Feature Extractor

In the proposed SeCoNet, the input image I is first passed through an Image Feature Extractor module, $\mathcal{F}(\cdot)$. Let $f_{img} = \mathcal{F}(I)$ represent the extracted features of the input image I. In the subsequent stages, the extracted features, f_{img} is simultaneously fed to the first stages of

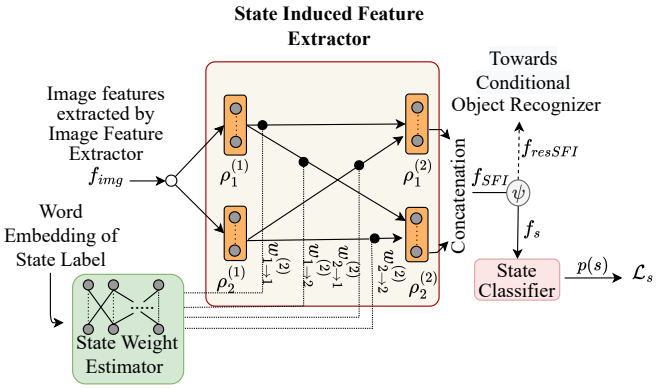


Figure 2.3: Block diagram of State Feature Isolator (i.e. first stage) of the *top branch* of SeCoNet

the *top branch* and the *bottom branch* for further processing. Next we explain the working principle of the *top branch*.

2.3.2.2 The top branch

As shown in Fig. 2.2, the *top branch* consists of two cascaded stages: State Feature Isolator (SFI) and Conditional Object Recognizer (COR). The first stage, SFI attempts to identify the state. In the second stage, COR tries to recognize the object, conditioned on the recognition results of the first stage. Next we describe the SFI.

First Stage State Feature Isolator (SFI): As shown in Fig. 2.2, the State Feature Isolator (SFI) of the proposed SeCoNet consists of three blocks: (a) State Weight Estimator (SWE), (b) State Induced Feature Extractor (SIFE) and (c) State Classifier (SC). Next we describe the detailed working of these components of SFI.

(a) State Weight Estimator: SWE component of SFI takes the word embedding of state label as input and generates a set of weights. The set of weights generated by SWE are multiplied with the intermediate features of each layer of SIFE (see Fig. 2.3). In other words, the weights (estimated by SWE) act as a filter and decide the priority of the intermediate features as extracted by each module of SIFE. Next we explain the SIFE module of SFI.

(b) State Induced Feature Extractor: The SIFE component of SFI is shown in Fig. 2.3. In the CZSL problem, at the test time, the model may experience unseen state-object compositions. Hence, to better guide the model for unseen compositions, SIFE utilizes both image features and word features, while extracting the state features. The SIFE is designed using a multi-layered architecture consisting of fully connected modules. The fully connected structure of SIFE (each module in each layer is connected with all the modules in the previous layer) specifically helps to integrate the word features with the intermediate image features. We choose a multi-layered architecture rather than a single-layered, to gradually (layer by layer) integrate the image features with word features. The input to each module in the first layer of SIFE is the image features f_{img} for the image I. For rest of the layers of SIFE, the weighted-sum of the outputs from the modules of the previous layer is input to the current layer. All the weights for the intermediate features in SIFE are derived by our SWE discussed in the previous paragraph.

Assume that the SIFE consists of η number of layers. The j^{th} layer of SIFE comprises of $\tau^{(j)}$ number of modules (see Fig. 2.3). For the k^{th} module of the j^{th} layer of SIFE, let $h_k^{(j)}$ represent module's input and $m_k^{(j)}$ represent module's output. Let $\rho_k^{(j)}$ denote the operations performed by the k^{th} module of the j^{th} layer, $m_k^{(j)} = \rho_k^{(j)}\big(h_k^{(j)}\big)$. Note that $h_k^{(j)}$, $m_k^{(j)}$ are vectors. Next, we multiply scalar weight (generated by the SWE) with $m_k^{(j)}$ for calculating the weighted sum of the outputs of all the modules in a layer. The input $h_k^{(j)}$ to k^{th} module of j^{th} layer of SIFE is determined as:

$$h_k^{(j)} = \sum_{t=1}^{\tau^{(j-1)}} w_{t\to k}^{(j)} m_t^{(j-1)}. \tag{2.1}$$

The weights $w_{t\to k}^{(j)} \in \mathbb{R}$ represent the weights of the edges between the module $m_t^{(j-1)}$ of the $(j-1)^{\text{th}}$ layer and $m_k^{(j)}$ of the j^{th} layer. Note that $m_1^0 = 1$, $\tau^{(0)} = 1$ and $\tau^{(\eta)} = 1$. These scalar weights are generated by SWE. These sets of scalar weights for all the layers in SIFE can be denoted by $\mathbb{W} = \{w_{t\to k}^{(j)} \mid j \in [1,\eta], t \in [1,\tau^{(j-1)}], k \in [1,\tau^{(j)}]\}$.

The word embedding (Bojanowski et al., 2017, Pennington et al., 2014) d_o of state label is the input to our SWE. SWE is a multi-layered fully connected neural network having $|\mathbb{W}|$ number of nodes in its output layer. SWE generates the outputs $r_{t\to k}^{(j)}$, $j \in [1,\eta]$, $t \in [1,\tau^{(j-1)}]$, $k \in [1,\tau^{(j)}]$. Subsequently, the scalar weights are obtained by normalizing the outputs $r_{t\to k}^{(j)}$ of SWE with softmax. The architectures of SIFE and SWE as implemented in the proposed SeCoNet, are inspired by the architecture of the feature extractor in (Purushwalkam et al., 2019). However, (Purushwalkam et al., 2019) presents a single stage, single branch architecture with simultaneous state-object recognition. In contrast SeCoNet utilizes a two-stage, two-branch approach for solving the CZSL problem. As mentioned in the second paragraph of Section 2.2, the two-stage, two-branch sequential approach has multiple advantages over the single stage, single branch approach of (Purushwalkam et al., 2019).

Finally, the output from all the modules of the final layer of SIFE is concatenated to obtain the features f_{SFI}. Next a scalar parameter $\psi \in [0,1]$ is used to obtain f_s and f_{resSFI} from f_{SFI}, as follows.

$$f_s = \psi f_{SFI}, \text{ and } f_{resSFI} = (1-\psi) f_{SFI}. \tag{2.2}$$

Next f_s is sent to the State Classifier for identifying the state in the image. Simultaneously, the f_{resSFI} is sent to the second stage of the *top branch* i.e. Conditional Object Recognizer. In our implementation, ψ is chosen experimentally as detailed in our ablation study in Section 2.4.6. Next, we explain the State Classifier.

(c) State Classifier: As shown in Fig. 2.3, the State Classifier takes the state features, f_s as input and predicts the state probability, p_s. The State Classifier is implemented using a fully connected layer. During training of the proposed SeCoNet, the loss $\mathcal{L}_s$ between the predicted p_s and ground truth state label for the input image is minimized as described in Section 2.3.3. COR is the second stage of *top branch* which aims to determine the probability of objects conditioned on the probability of a state identified in the first stage (SFI). The COR is explained next.

Second Stage Conditional Object Recognizer (COR): COR is composed of three blocks:

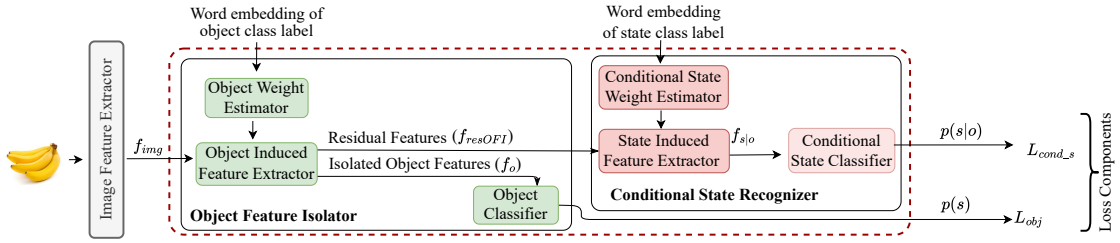


Figure 2.4: Flowchart of the *bottom branch* of SeCoNet

(a) Conditional Object Weight Estimator (COWE), (b) Object Induced Feature Extractor and (c) Conditional Object Classifier. These components of COR are explained briefly next. The block level diagram of COR is shown in Fig. 2.2.

The Object Induced Feature Extractor of COR is a multi-layered network similar to the architecture of SIFE in SFI. Like SWE in SFI, COR also has a corresponding Conditional Object Weight Estimator which takes the word embedding of object label as input and generates a set of weights. These weights are multiplied with the intermediate outputs of the COR network. Object Induced Feature Extractor processes f_{resSFI} and returns the conditional object features, $f_{o|s}$. The working principles of Object Induced Feature Extractor and Conditional Object Weight Estimator are similar to the working principles of SIFE and SWE as given in (2.1). The internal diagram of COR and COWE are similar to the diagram shown in Fig. 2.3.

However, $f_{o|s}$ is sent to the Conditional Object Classifier to obtain the conditional object probabilities, $p(o|s)$. Next we explain the *bottom branch* of the proposed SeCoNet.

2.3.2.3 The bottom branch

In the *top branch*, the first stage estimates the probability of state and the second stage recognizes the probability of the object conditioned on the state recognized in the first stage. While in the *bottom branch*, the first stage estimates the probability of object and the second stage evaluates the probability of state conditioned on the object recognized in the first stage. However, the architecture of the *bottom branch* is completely identical to the architecture of the *top branch*.

The first stage of *bottom branch* is the Object Feature Isolator (OFI) which consists of Object Induced Feature Extractor (OIFE), Object Weight Estimator and Object Classifier (see Fig. 2.4). Similar to the process flow of *top branch*, the output from the Image Feature Extractor, f_{img} is sent to the OIFE in OFI of *bottom branch*. OIFE returns a features f_{OFI} for the object present in the input image. Subsequently, the object features f_o and the residual features f_{resOFI} are determined from f_{OFI} as we have done in case of SFI using (2.2). Next f_o is fed to the Object Classifier, which provides the object probability p_o. In the second stage, the Conditional State Recognizer (CSR) is implemented. The CSR takes residual features, f_{resOFI} as input. CSR determines the conditional state probability $p(s|o)$ based on the prior recognition of object in the first stage.

The loss $\mathcal{L}_o$ between p_o and the respective object ground truth labels for the input image is minimized during training SeCoNet. Similarly, the loss $\mathcal{L}_{cond\_s}$ between $p(s|o)$ and the respective state ground truth labels for the input image is also minimized at the time of

training SeCoNet as explained in Section 2.3.3.

Thus the *top* and *bottom branches* compensate the errors of each other in correctly recognizing state-object compositions. Hence we have used a two-branch architecture. Next we present the learning strategy of SeCoNet.

2.3.3 Learning Strategy

All of the State, Conditional State, Object and Conditional Object Classifiers calculate a confidence score for being a state and an object, respectively. During training, computing the scores of all possible classes i.e. valid state-object pairs (or compositions) for each image is a computationally expensive procedure. Thus, we follow a negative sampling based strategy (Bengio and Senécal, 2003, Purushwalkam et al., 2019, Li et al., 2020) for generating the scores and subsequently for training the network. Assume $c_k = (s_i, o_j)$ be the class label of the i^{th} training image I_i from the training set T. For the image I_i, we first collect n_1 number of negative state-object labels $\tilde{c}_k = (\tilde{s}_i, \tilde{o}_j)$, $i \in \{1, 2, \cdots, N_s\}, j \in \{1, 2, \cdots, N_o\}, k \in \{1, 2, \cdots, |\mathcal{C}|\}$ i.e. s_i and $\tilde{s}_i$ are not identical and o_j and $\tilde{o}_j$ are different. For each input data, negative samples are independently sampled from the dataset. Subsequently, for each of the four classifiers in SeCoNet, we create an array of $n_1 + 1$ elements, where first index of the array represents the positive label and rest of the indices denote negative labels respectively. For training the SeCoNet, we use one-hot label vectors (where first element of each vector is 1 and rests are 0) as ground-truth corresponding to the state, object, object given the state, and state given the object labels for the image I_i. During training, the state features f_s from the SIFE and the object features f_o from the OIFE are combined, and sent to state-object joint classifier to obtain the predicted state-object composition probabilities. Given the above setting the complete loss $\mathcal{L}$ is represented as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_o + \lambda_3 \mathcal{L}_{cond\_s} + \lambda_4 \mathcal{L}_{cond\_o} + \lambda_5 \mathcal{L}_{s\_o} + \lambda_6 \mathcal{L}_{dis\_ent}. \tag{2.3}$$

where $\mathcal{L}_s$, $\mathcal{L}_o$, $\mathcal{L}_{cond\_s}$, $\mathcal{L}_{cond\_o}$, and $\mathcal{L}_{s\_o}$ are the losses in recognizing states, objects, states given objects, objects given states, and state-object compositions; $\mathcal{L}_{dis\_ent}$ is the disentanglement loss; λ_1, λ_2, λ_3, λ_4, λ_5 and $\lambda_6 \in [0, 1]$ are the scalar weights for the loss components involved in $\mathcal{L}$. Here $\mathcal{L}_s(\cdot)$, $\mathcal{L}_o(\cdot)$, $\mathcal{L}_{cond\_s}(\cdot)$, $\mathcal{L}_{cond\_o}(\cdot)$, and $\mathcal{L}_{s\_o}(\cdot)$ are the *cross-entropy* losses. The disentanglement loss $\mathcal{L}_{dis\_ent}$, which is a novel contribution in proposed SeCoNet, is explained next.

2.3.3.1 Gradient Penalization for Feature Isolation

There are three components of the visual features for an image of state-object composition. These three components are: (a) features representing the state uniquely, (b) features representing the object uniquely, and (c) state-object joint features. Disentanglement of object and state features is a key task for the recognition of object and state in a compositional image. However, the existence of the third kind of features (which represents state-object composition) is the main obstacle for effective disentanglement of the features for the states and objects. As the third kind of features i.e. state-object joint features are unique to each composition, it cannot help to recognise any other state-object composition.

For any image I, let us assume that f_s, f_o, and $f_{<s,\, o>}$ be the visual features representing states, objects and state-object compositions respectively. Note that $< s,\ o >$ denotes the

state-object composition. However, f_{img}, which is obtained from the output of Image Feature Extractor, is the overall feature representing the input image I. Thus, we can write,

$$\begin{aligned} f_{img} &= f_s + f_o + f_{<s,\,o>}, \\ \text{or, } \mathbb{I} &= \frac{\partial f_s}{\partial f_{img}} + \frac{\partial f_o}{\partial f_{img}} + \frac{\partial f_{<s,\,o>}}{\partial f_{img}}, \\ \text{or, } \frac{\partial f_{<s,\,o>}}{\partial f_{img}} &= \mathbb{I} - \left(\frac{\partial f_s}{\partial f_{img}} + \frac{\partial f_o}{\partial f_{img}}\right). \end{aligned} \tag{2.4}$$

Here $\mathbb{I}$ represents the identity matrix. f_s and f_o are obtained from the first stages of the *top branch* and *bottom branch* of our SeCoNet respectively. The feature interactions between the state and the object are modeled by prior CZSL approaches in different ways. For example, prior works by (Nagarajan and Grauman, 2018, Li et al., 2020, 2021) considered state features as an operator based transformation over the object features. Specifically, the work in (Nagarajan and Grauman, 2018) considered the state features as linear transformation of the object features. In a similar line of thought, we have considered the visual features of the input image, f_{img} as linear addition of state, object and state-object features, (f_s, f_o, $f_{<s,\,o>}$). Given the above setting, we introduce a new loss component, $\mathcal{L}_{dis\_ent}$ for disentanglement of state and object features from the input image I as:

$$\mathcal{L}_{dis\_ent} = \left\| \frac{\partial f_{<s,\,o>}}{\partial f_{img}} \right\|_2^2 = \left\| \mathbb{I} - \left(\frac{\partial f_s}{\partial f_{img}} + \frac{\partial f_o}{\partial f_{img}}\right) \right\|_2^2 = \left\| \frac{\partial f_s}{\partial f_{img}} + \frac{\partial f_o}{\partial f_{img}} \right\|_2^2, \tag{2.5}$$

where $\mathbb{I}$ is an additive constant and hence, it is ignored in the final expression.

However, the loss term in (2.5) is not sufficient for feature disentanglement as it has some challenges of its own. The primary challenge is the difficulty in isolating the multi-dimensional features for the states and objects. Hence, for accurate isolation of the features f_s and f_o, we introduce a gating function $\phi(x)$ to prioritize the important features representing the states or objects, where x is a feature vector.

Therefore, we define two separate gating functions $\phi_s(x) = W_s^T x$ and $\phi_o(x) = W_o^T x$ for state and object features respectively, where W_s and W_o are trainable parameters in SeCoNet. Thus, instead of using the features f_s and f_o in formulation of $\mathcal{L}_{dis\_ent}$ in (2.5), we use $\phi_s(f_s)$ and $\phi_o(f_o)$. Then (2.5) becomes

$$\mathcal{L}_{dis\_ent} = \left\| \frac{\partial \phi_s(f_s)}{\partial f_{img}} + \frac{\partial \phi_o(f_o)}{\partial f_{img}} \right\|_2^2. \tag{2.6}$$

Moreover, in order to regularize the disentanglement loss $\mathcal{L}_{dis\_ent}$, we add an l_1 norm gradient penalization term in (2.6) for each of the two gating functions $\phi_s(f_s)$ and $\phi_o(f_o)$ as follows,

$$\mathcal{L}_{dis\_ent} = \left\| \frac{\partial \phi_s(f_s)}{\partial f_{img}} + \frac{\partial \phi_o(f_o)}{f_{img}} \right\|_2^2 + \left\| \frac{\partial \phi_s(f_s)}{\partial f_s} \right\|_1^1 + \left\| \frac{\partial \phi_o(f_o)}{\partial f_o} \right\|_1^1. \tag{2.7}$$

In the next section we propose a theoretical analysis of the disentanglement ability of SeCoNet.

2.3.4 Analysis of Disentanglement Ability of SeCoNet

We argue that the following two criteria need to be satisfied for effective disentanglement of f_s and f_o (Do and Tran, 2019, Eastwood, Cian and Williams, Christopher KI, 2018): (a) f_s should be separable from f_o and vice-versa, (b) each of the disentangled features, f_s and f_o must be individually informative with respect to f_{img}.

One of the approaches to quantify (a) and (b) above is by analyzing the conditional mutual information ($MI(f_{img}, f_s|f_o)$) between f_{img} and f_s given f_o. The $MI(f_{img}, f_s|f_o)$ represents the amount of information in f_{img}, which is contained in f_s but not in f_o. Thus, higher the $MI(\cdot)$ is, better is the disentanglement.

Note that $MI(f_{img}, f_s|f_o)$ measures the isolation of state features from the object features. Similarly, $MI(f_{img}, f_o|f_s)$ can be calculated for measuring the isolation of object features from the state features. Rest of the theoretical analysis in this section is carried out considering $MI(f_{img}, f_s|f_o)$. Without loss of generality, the similar analysis can also be extended considering $MI(f_{img}, f_o|f_s)$.

Proposition 1. *If N_t is the number of images in the dataset and k is the minimum number of images in the training dataset in which a particular state is present, then $MI(f_{img}, f_s|f_o) \geq \log\left(\frac{N_t}{k}\right)$, $k > 0$, typically $k \ll N_t$.*

This proposition provides a lower bound on the conditional mutual information $MI(f_{img}, f_s|f_o)$. As a result, this proposition presents a theoretical analysis of the extent to which the proposed SeCoNet is successful in disentanglement.

Proof: Using the chain rule of mutual information (Do and Tran, 2019),

$$MI(f_s, f_{img}|f_o) = MI(f_s, f_{img}) - (MI(f_s, f_o) - MI(f_s, f_o|f_{img})). \tag{2.8}$$

As SeCoNet uses distinct branches to obtain f_s and f_o, we assume mutual information between f_s and f_o given f_{img} to be negligible (Do and Tran, 2019). Assuming $MI(f_s, f_o|f_{img}) \approx 0$, we simplify (2.8) as follows,

$$MI(f_s, f_{img}|f_o) \approx MI(f_s, f_{img}) - MI(f_s, f_o). \tag{2.9}$$

Following Appendix A.1, $MI(f_{img}, f_s|f_o)$ is simplified as,

$$MI(f_{img}, f_s|f_o) = H(f_{img}) - H(f_s, f_{img}) - H(f_o) + H(f_s, f_o), \tag{2.10}$$

where $H(\cdot)$ represents the entropy (Cover, Thomas M, 1999). Since the architecture of SeCoNet has two distinct branches to separately find out the state and object features,

$$H(f_s, f_o) = H(f_s) + H(f_o). \tag{2.11}$$

Using the above result, (2.10) can be further simplified as,

$$\begin{aligned} MI(f_{img}, f_s|f_o) &= H(f_{img}) - H(f_s, f_{img}) - H(f_o) + H(f_s) + H(f_o) \\ &= H(f_{img}) + H(f_s) - H(f_s, f_{img}). \end{aligned} \tag{2.12}$$

From the definition of entropy, $H(f_s)$ can be expressed as,

$$H(f_s) = -\mathbb{E}_{p(f_s)} \log \big(p(f_s)\big), \tag{2.13}$$

where $p(f_s)$ represents the probability distribution of the state features and $\mathbb{E}_{p(f_s)}$ represents expectation over the probability distribution of state features. Note that state features are generated from the image features. In other words f_s is always dependent on f_{img}. So instead of evaluating the expectation in (2.13) only over the state features, we evaluate it over the state features conditioned on the image features. Similarly, instead of evaluating the expected value of the state features, we evaluate the expectation of state features conditioned on the image features. However, the state features conditioned on image features are dependent on the image feature distribution. Hence the weights of the conditional probability distribution are assigned using the corresponding image feature probabilities. Thus (2.13) becomes,

$$H(f_s) \equiv -\mathbb{E}_{p(f_s|f_{img})} \log \big(\mathbb{E}_{p(f_{img})} p(f_s|f_{img})\big) \tag{2.14}$$

$$= -\sum_r p(f_s^r|f_{img}^r) \log \big(\sum_u p(f_{img}^u) p(f_s^r|f_{img}^u)\big), \tag{2.15}$$

where $p(f_{img}^u)$ denotes the probability of features of the u^{th} image. Without loss of generality, we assume all the images in the dataset have the same probability. So, $p(f_{img}^u) = \frac{1}{N_t}$. Thus we can simplify (2.14) as follows,

$$H(f_s) \equiv -\sum_r p(f_s^r|f_{img}^r) \log \left(\sum_u \frac{1}{N_t} p(f_s^r|f_{img}^u)\right). \tag{2.16}$$

Each state can have feasible compositions with many objects. The dataset contains multiple images of a particular state-object composition. Thus, many images may have same state and thus same state features. Let k represent the minimum number of images in which features of a particular state can be present. Evidently, $k \geq 1$. Thus, $\sum_s p(f_s^r|f_{img}^s) \geq kp(f_s^r|f_{img}^r)$. Following Appendix A.1, (2.16) becomes,

$$H(f_s) \geq \log\left(\frac{N_t}{k}\right) - \sum_r p(f_s^r|f_{img}^r) \log \big(p(f_s^r|f_{img}^r)\big). \tag{2.17}$$

Simplifying the expression for $H(f_s, f_{img})$ in (2.12), we get

$$H(f_s, f_{img}) = H(f_{img}) + H(f_s|f_{img}). \tag{2.18}$$

For detailed derivation of (2.18), refer Appendix A.1. Again $H(f_s|f_{img})$ can be expressed as,

$$H(f_s|f_{img}) = -\mathbb{E}_{q(f_s|f_{img})}[\log(q(f_s|f_{img}))]$$

$$= -\sum_r p(f_s^r|f_{img}^r) \log[p(f_s^r|f_{img}^r)]. \tag{2.19}$$

Split →			Train		Validation			Test		
Dataset ↓	State	Obj	Seen Class	Img	Seen Class	Unseen Class	Img	Seen Class	Unseen Class	Img
MIT-States	115	245	1262	30k	300	300	10k	400	400	13k
UT-Zappos50k	16	12	83	23k	15	15	3k	18	18	3k
C-GQA	453	870	5592	27k	1252	1040	7k	888	923	5k

Table 2.1: Brief summary for the datasets.

Finally substituting the expression for $H(f_s)$ from (2.17) in (2.12),

$$\begin{aligned} MI(f_{img}, f_s|f_o) &= H(f_{img}) + H(f_s) - H(f_s, f_{img}) \\ &\geq H(f_{img}) + \log\left(\frac{N_t}{k}\right) - \sum_r p(f_s^r|f_{img}^r)\log\left(p(f_s^r|f_{img}^r)\right)) \\ &- \left(-\sum_r p(f_s^r|f_{img}^r)\log\left(p(f_s^r|f_{img}^r)\right)\right) - H(f_{img}) \\ &\geq \log\left(\frac{N_t}{k}\right). \qquad \blacksquare \end{aligned}$$

Next we present the inference strategy.

2.3.5 Inference Strategy

A test image I_T is first passed through the Image Feature Extractor of the network to obtain the image features. Subsequently, the image features along with the word embedding of all valid possible state labels are fed to the first stage of the *top branch* comprising of SFI (refer Fig. 2.1). The State Classifier of the first stage returns confidence scores for the states. These confidence scores are normalized to obtain the state probabilities p_s using the *softmax* function. Simultaneously, the image features along with the word embedding of all valid possible object labels are sent to the first stage of the *bottom branch* comprising of OFI (refer Fig. 2.1). The object classifier of the *bottom branch* determines the confidence scores for the objects and these scores are normalized to determine the object probabilities p_o. The state having the highest probability in p_s and the object presenting highest probability in p_o define the predicted state-object composition for the test image I_T. Note that the first stage of the *top* and *bottom branches* of SeCoNet are only utilized during test. Our ablation study in Section 2.4.6 suggests that the state-object composition inference results generated from the first stage is much more accurate than the results generated from the second stage. Next we present the experimental details of the proposed approach.

2.4 Experiments

2.4.1 Datasets

The proposed approach is evaluated on three publicly available CZSL benchmark datasets, namely MIT-States (Isola et al., 2015), C-GQA (Naeem et al., 2021) and UT-Zappos50K (Yu and Grauman, 2014, 2017). Table 2.1 reports different statistics of the three datasets. MIT-

States is a real-life CZSL dataset containing images of diverse set of state-object compositions e.g *ripe apple*, *young bear* etc. Similar to MIT-States, the C-GQA also consists of images of real-life state-object compositions. C-GQA contains images from more number of state-object compositions in comparison to MIT-States (see Table 2.1). We also evaluate our approach on another dataset, UT-Zappos50k (Yu and Grauman, 2014, 2017). UT-Zappos50k has approximately 33,000 shoe images. The states in UT-Zappos50k represent the materials of the shoes (e.g. *cotton* and *wool*) while the objects are the types of shoes (e.g. *sandals* and *slippers*). For MIT-States and UT-Zappos50k datasets, the train-test splits proposed in (Purushwalkam et al., 2019) are used. For C-GQA dataset we use the train-test split proposed by Naeem *et al.*(Naeem et al., 2021). For the test-train split of all three datasets, the test set includes images of both unseen and seen state-object compositions.

2.4.2 Implementation Details

Image Feature Extractor utilizes ResNet18 (He et al., 2016) pre-trained with ImageNet (Russakovsky et al., 2015) *conv_5_4* layer as the backbone to extract the image level features. In order to find out the word level representations of each state label and object label, we use 300-dimensional FastText (Bojanowski et al., 2017) word embedding. The architectures of SFI, COR, OFI and CSR stages (see Section 2.3.2) in our SeCoNet are identical. Both SIFE and OIFE in SFI, COR, OFI and CSR consist of 3 layers. Each of the 3 layers consists of 28 modules for MIT-States and 10 modules for UT-Zappos50k. Each module consists of a fully connected layer with 16 nodes followed by one ReLU layer. Besides, each module in the last two layers includes an additional batch-norm layer after the ReLU layer. The architectures of the SFI and OFI are identical. Thus, the dimensions of f_s and f_o are also same. As mentioned in Section 2.3.3, for each input image, the number of negative samples, n_1 is taken to be 500. The parameters of the proposed architecture is experimentally chosen. As mentioned in Section 2.3.2, the classifiers of our SeCoNet are constructed using a fully connected layer. The Weight Estimators (WEs) are also built using fully connected layer with 300 input nodes and 64 hidden nodes. In (2.2), the scalar parameter ψ is set to 0.5. The proposed network is optimized by Adam (Kingma and Ba, 2015) optimizer. Since UT-Zappos50k has fewer number of images than the number of images in MIT-States, the mini-batch size is set to 32 for UT-Zappos50k and 128 for MIT-States. The weights of different loss components in (2.3) are experimentally set to $\lambda_1 = 0.05$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$, $\lambda_4 = 0.05$, $\lambda_5 = 0.75$ and $\lambda_6 = 0.05$. The proposed model is trained for 30 epochs with the learning rate $1e^{-4}$. The proposed approach is implemented in python 3 on a Ubuntu 18.04 system with NVIDIA RTX GPU (24 GB VRAM), Intel i9 CPU and 64 GB RAM. Next we mention the competing approaches followed by the description of the metrics for evaluation.

2.4.3 Competing Approaches

We compare the proposed solution for CZSL against several recent state-of-the-art approaches: LE (Misra et al., 2017), AAO (Nagarajan and Grauman, 2018), SymNet (Li et al., 2020) and TMN (Purushwalkam et al., 2019). The results for all these methods are reproduced from (Naeem et al., 2021) and the corresponding open source implementations.

Dataset →	UT-Zappos50k							MIT-States							C-GQA						
Algorithm↓	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*
AAO	21.5	25.9	59.8	54.2	40.8	38.9	69.6	2.5	1.6	14.3	17.4	9.9	21.1	23.6	0.9	0.4	11.8	4.4	4.6	17.7	19.9
LE	26.4	25.7	53.0	**61.9**	41.0	41.2	69.2	3.0	2.0	15.0	20.1	10.7	23.5	26.3	0.8	0.4	11.4	4.8	4.5	18.0	20.5
TMN	36.8	29.3	58.7	60.0	45.0	40.8	69.9	3.5	2.9	20.2	20.1	13.0	23.3	26.5	1.7	0.8	18.8	6.1	6.4	16.5	25.6
SymNet	25.9	23.9	53.3	57.9	39.2	40.5	71.2	4.3	3.0	24.4	**25.2**	16.1	26.3	28.3	**2.9**	1.0	20.6	7.0	7.3	21.4	25.1
SeCoNet	**37.5**	**31.2**	**61.1**	60.8	**46.1**	**42.4**	**72.9**	**4.9**	**4.0**	**26.0**	24.8	**17.4**	**27.5**	**30.0**	2.0	**1.3**	**21.8**	**7.8**	**7.7**	**22.5**	**27.0**

Table 2.2: Closed-world results.

2.4.4 Evaluation Metrics

All the methods are evaluated using the experimental protocol explained in (Purushwalkam et al., 2019, Xu et al., 2021b, Li et al., 2020, Atzmon et al., 2020a). The following paragraphs describe the evaluation protocol.

Note that in CZSL, both seen and unseen state-object compositions may occur during test. Evaluating the prediction performance for both seen and unseen compositions using a model trained only on seen compositions, has some flaws of its own (Chao et al., 2016). During test, the model is already familiar with the seen compositions. Hence, seen compositions are predicted much better than unseen compositions due to inherent bias of the model towards the seen compositions. To compensate the effects of this bias in the model, a scalar value (referred to as *calibration bias*) is added to the predicted scores of all unseen compositions. For a given value of the *calibration bias*, we compute the accuracy for both seen and unseen compositions. On a seen w.r.t. unseen accuracy graph, we plot the seen and unseen accuracy of a model obtained by varying the *calibration bias*. Subsequently, we obtain a curve and the *AUC* reports the combined performance of the algorithm (see Chao et al. (2016), Purushwalkam et al. (2019) for further details).

The AUC metrics are reported on both the test set and the validation set for the three datasets (Purushwalkam et al., 2019, Yu and Grauman, 2014, 2017, Naeem et al., 2021). The *AUC* values obtained on the validation set and the test set are referred to as *val AUC* and *test AUC*, respectively.

For fair comparison with competing approaches, beside *AUC*, we separately compute the accuracy for test images of seen state-object pairs (referred to as *seen* accuracy) and for images of unseen state-object compositions (referred to as *unseen* accuracy). We also calculate *harmonic mean* (*HM*) that quantifies the overall performance of both *seen* and *unseen* accuracy in a single metric, where $HM = \frac{2\,(seen\ accuracy)\,(unseen\ accuracy)}{(seen\ accuracy)+(unseen\ accuracy)}$. Here *seen* and *unseen* represents the average classification accuracy of seen and unseen classes, respectively. We also report the top-1 state and object recognition accuracy (referred to as *state* and *object* accuracy). Next we present the results and analysis.

2.4.5 Results and Analysis

Table 2.2 tabulates the results of our proposal and other competing methods on MIT-States, UT-Zappos50k and C-GQA datasets. Table 2.2 lists the recognition results in terms of seven accuracy measures: *val AUC*, *test AUC*, *seen*, *unseen*, *HM*, *object* and *state* recognition accuracies.

The Table 2.2 shows that our algorithm outperforms all competing approaches for *val*

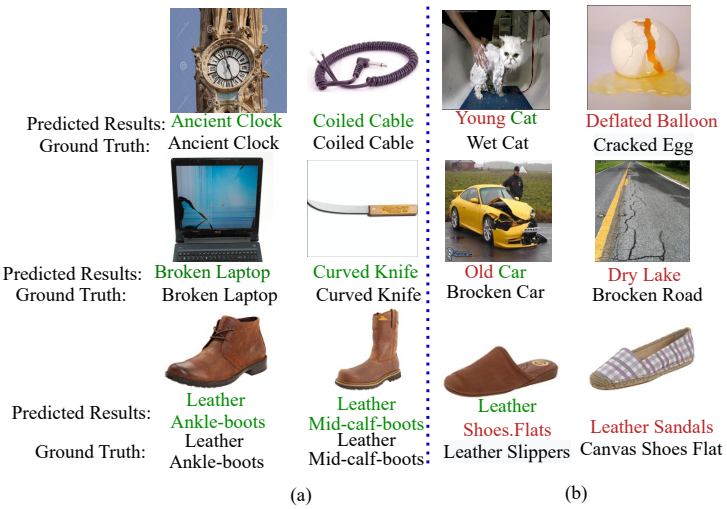


Figure 2.5: A few examples of some (a) successful (see first two columns) and (b) unsuccessful (see second two columns) image classification performances of our proposal on MIT-States (see first and second rows) and UT-Zappos50k (see third row) datasets. Texts in green represent correct predictions while texts in red represent incorrect predictions.

AUC and *test AUC* metrics on both MIT-States and UT-Zappos50k datasets. SeCoNet performs either similar or better than other competing approaches in case of *seen*, *HM*, *state* and *object* accuracy on MIT-States.

For UT-Zappos50k dataset, the proposed algorithm is better compared to other baselines considering all performance metrics discussed in this chapter. On UT-Zappos50k, the improvement in *test AUC* over the closest competitor, SymNet (Li et al., 2020), is nearly 6.8%. For other competitive algorithms, the improvement in *test AUC* ranges from 1.9% to 5%. Similar improvement is also achieved in case of *val AUC* on UT-Zappos50k. Further, on both MIT-States and UT-Zappos50k, we can see an improvement on state recognition (see Table 2.2) by at least 1.2% and improvement in object recognition accuracy by at least 1.7%.

HM is a balanced metric as it incorporates the performance of the algorithm on both *unseen* and *seen* classes. In case of UT-Zappos50k dataset, our algorithm shows 6.9%, 1.1%, 5.3%, and 5.1% performance (in *HM*) improvements over SymNet, TMN, AAO and LE respectively.

As shown in Table 2.1, for both the datasets, numbers of seen and unseen classes in the validation set are lower than the number of seen and unseen classes in the test set. Hence, almost all the competing methods including ours achieve better *AUC* on the validation set, in comparison to the *test AUC*.

As reported in Table 2.2, on C-GQA dataset, the proposed SeCoNet outperforms all other algorithms in terms of *test AUC, seen, unseen, HM, state* and *object* metrics. However on *val AUC*, SymNet has reported a marginally better result. C-GQA has much more feasible state-object compositions than the other two datasets (see Tab. 2.1). Hence C-GQA is a more challenging CZSL dataset. Consequently, *AUC* scores for C-GQA for all the algorithms are lower than that for the MIT-States dataset.

We can see in Table 2.2 that the reported *AUC*s are quite low, particularly on the MIT-

States dataset for all the baseline algorithms, in comparison to UT-Zappos50k. The low AUC on MIT-States dataset can be attributed to multiple issues. First, this dataset includes a large number (around 1962) of feasible state-object pairs and each object is associated with 9 different states, on an average. Naturally, the dataset presents a large number of potentially valid pairs for each given image. Also, the study in (Atzmon et al., 2020a, Naeem et al., 2021) reports that the MIT-States dataset contains a lot of noisy class labels. The noise in class label is mainly due to incorrect annotation of images in MIT-States dataset. This creates a challenge for achieving higher performance on MIT-States dataset.

Fig. 2.5 illustrates some example CZSL image classification results of the proposed approach. First two columns show the successful cases while the later two columns highlight the unsuccessful cases. First two rows of Fig. 2.5 highlight the example results of our SeCoNet for the images from test set of MIT-States dataset, while the last row demonstrates the example results of the proposed approach for the images from test set of UT-Zappos50k dataset. For each of the datasets, the SeCoNet is trained with the train set of the respective dataset. Next we carry out the experiment for establishing the qualitative image classification performance of our SeCoNet.

Following (Nagarajan and Grauman, 2018), we investigate the qualitative image classification performance of the proposed SeCoNet on ImageNet (Russakovsky et al., 2015). The SeCoNet model, which is trained with the MIT-States dataset, is tested with a set of images from the validation set of ImageNet. We first select a query label of the state-object composition and we then pass all the images from the validation set of ImageNet to SeCoNet. Next, images for which our model predicts highest confidence score corresponding to the query compositional label are noted. These images are reported as qualitative classification results corresponding to the query compositional label. Fig. 2.6 illustrates a few successful image classification results for a few labels of the state-object compositions. Hence proposed algorithm is able to generalize well in other datasets (in this case ImageNet) over which training is not performed. This establishes the efficacy of the proposed SeCoNet for cross-dataset qualitative image classification. Next we present our ablation study.

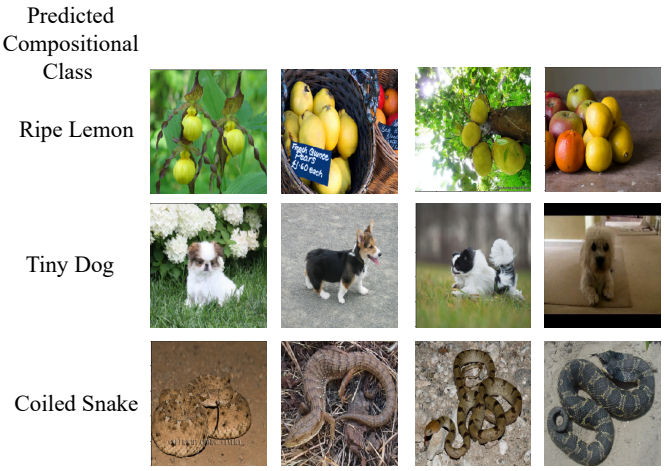


Figure 2.6: Examples of qualitative results images from the ImageNet dataset (Russakovsky et al., 2015).

2.4.6 Ablation Study

Proposed SeCoNet consists of a number of important components. In the following paragraphs, we establish the significance of each of the components. First, we examine the importance of different branches of our SeCoNet followed by the importance of different stages of the branches in the end result. The results of this experiment are tabulated in Table 2.3.

Justification for using Two Branches in SeCoNet: The second row of Table 2.3 reports the AUC on the val set of the datasets, if we exclude the *bottom branch* from SeCoNet (i.e. when the SeCoNet is configured with only the *top branch*). The prediction results for states and objects are taken from the first stage and second stage of the *top branch* of SeCoNet, respectively. This experiment reports 1.1% and 3.5% lower *val* AUC in comparison to the proposed SeCoNet (see fourth row in Table 2.3) on MIT-States and UT-Zappos50k datasets, respectively. The inferior performance of this configuration of the network can be justified as follows.

In the sequential recognition approach, recognition results of the second stage are dependent on the recognition performances of the first stage. Thus, the recognition error of first stage must affect the recognition performance of the second stage. Hence, we build a two-stage architecture with the two branches (*top branch* and *bottom branch*) recognizing state and object in reverse order. The two-branch architecture somewhat addresses the highlighted weakness of the sequential learning approach by compensating the errors of the *top branch* in the *bottom branch* and vice-versa. The results in the second row of Table 2.3 justifies the use of two-branch architecture. Note that, while excluding a branch or a stage for considering different configurations of SeCoNet in this ablation study, we also exclude the corresponding loss components in the proposed loss as computed in (2.3).

Justification for using Two Stages in Each Branch of SeCoNet: The second stage of *top* and *bottom branch* aims to recognize the conditional state and conditional object probabilities. Now, we validate the utility of incorporating the conditional recognition in second stage of *top* and *bottom branches*. In Table 2.3, the third row corresponds to the performances of the SeCoNet on validation set of the datasets after discarding the second stages from both the *top* and *bottom branches*. Therefore, in this case, the SeCoNet is configured with only the first stage in both *top* and *bottom branches* i.e. with SFI and OFI. This experiment shows 0.8%

Various Configurations of the Proposed Network	MIT-States	UT-Zappos50k
SeCoNet with single branch and two stages	3.8	34.0
SeCoNet with two branches, each branch has one stage	4.1	35.1
SeCoNet with two branches and two stages in each branch, where output of the first stage determine the unseen state-object composition	**4.9**	**37.5**
SeCoNet with two branches and two stages in each branch, where outputs of the second stage determine the unseen state-object composition	3.6	33.8
SeCoNet trained excluding $\mathcal{L}_{dis\_ent}$	3.7	35.3

Table 2.3: Performances (in terms of *val* AUC) of the different configurations of SeCoNet

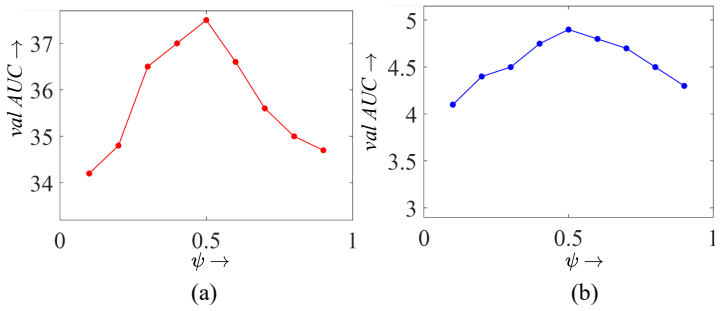


Figure 2.7: Plot of *val AUC* with changing values of ψ for (a) UT-Zappos50k and (b) MIT-States

and 2.4% lower *val AUC* than the *val AUC* for SeCoNet on MIT-States and UT-Zappos50k, respectively. Hence, this is evident that the conditional recognition in the second stage of each branch plays a vital role in achieving better prediction performance. The results in fourth row in Table 2.3 imply the effectiveness of two-branch two-stage architecture of SeCoNet.

Justification for Performing Inference from First Stage of Each Branch in SeCoNet: As explained in Section 2.3.5, during inference, the state and object probabilities are calculated from the first stage of the *top* and *bottom branch*, respectively. Here we discuss the effectiveness of this inference strategy. The fourth row of Table 2.3 shows the result of our algorithm if the inference is conducted from the first stage of the *top* and *bottom branches*. The fifth row of Table 2.3 shows the result of our algorithm if the inference is done from the second stage of the *top branch*. Evidently, this experiment suggests that the SeCoNet achieves better result if the inference about the state and object probabilities are obtained from the first stage of each branch.

Importance of the Disentanglement Loss: The theoretical motivation (or the intuitive explanation) for the proposed disentanglement loss of (2.7) is provided in Section 2.3.3.1. Here we experimentally assess the significance of our novel disentanglement loss term by removing $\mathcal{L}_{dis\_ent}$ from the loss function (2.3) during training of the SeCoNet. The results of this experiment on the validation set of the datasets are tabulated in the last row of Table 2.3. The results of SeCoNet considering the disentanglement loss can be found in the fourth row of the table. Table 2.3 shows that the performance of SeCoNet drops 1.2% and 2.2% in terms of *val AUC* in case of MIT-States and UT-Zappos50k, respectively, if we ignore the disentanglement loss. This clearly indicates the efficacy of the proposed disentanglement loss. Next we carry out the experiment for choosing the near optimal value of ψ.

Choice of the Parameter ψ: In this section, we discuss the effect of the parameter ψ (refer to (2.2)) on the performance of our SeCoNet for the validation set of the MIT-States and UT-Zappos50k datasets. The *AUC* w.r.t. ψ plot in Fig. 2.7 illustrates the fluctuations of *AUC* for different values of ψ. As shown in Fig. 2.7, both higher and lower values of ψ result in lower *AUC*. Fig. 2.7(a) and (b) show that the highest *AUC* on both the datasets are obtained at $\psi = 0.5$. This observation can be intuitively justified as follows.

For an input image, the IFE returns the image level features f_{img} which essentially represent both the object and its state. The SIFE module inside the SFI stage of the *top branch* (refer to Fig. 2.2) separates the state features f_s and residual features f_{resSFI} from the input

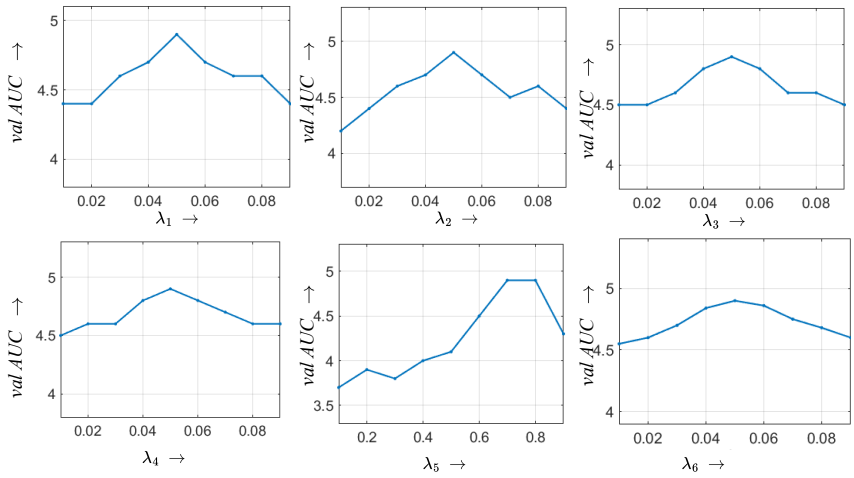


Figure 2.8: Variation of AUC for different values of $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ and λ_6

image features f_{img}. The residual features, which predominantly contain the object features, are then sent to the second stage (i.e. COR) of *top branch*. As defined in (2.2), setting a higher value of ψ, results in higher proportion of f_{SFI} being assigned as f_s, and lower proportion of f_{SFI} being assigned as f_{resSFI}. Hence, higher value of ψ results a better state recognition performance but lower object recognition accuracy. Conversely, if we set the lower values to ψ, major part of the f_{SFI} is assigned to f_{resSFI} and passed to the second stage of *top branch*. As a result, the state recognition probability is reduced. In the CZSL problem, the joint state-object recognition AUC depends on the accurate recognition of both object and state. Therefore, we need to set ψ in a way that both the state and object recognizer perform equally. The performances of state and object recognizer is balanced if $\psi = 0.5$ and hence, yields a better AUC for recognition of state-object composition. Next we present a discussion on the effect of the weights on loss components of (2.3) of the SeCoNet.

Ablation Study on the Weights of the Loss Components: The final composite loss for the SeCoNet is shown in (2.3). The weights are $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ and λ_6. In this section we analyse the sensitivity of the results of SeCoNet on the MIT-States (Isola et al., 2015) with respect to the above mentioned weights. The results of the experiments are plotted in Fig. 2.8. While varying one weight, all other weights are kept constant at their respective reported values (see Section 2.4.2).

It can be seen that the final *val* AUC is more sensitive with respect to the variations in λ_1 and λ_2 in comparison to λ_3 and λ_4. This can be justified as follows. The first stage of the *top branch* recognises the state present in the image. The second stage recognises the object present in the image conditioned on the state recognised in the first stage. Evidently the conditional object recognition of the second stage and the final state-object recognition is largely dependent on the first stage's state recognition. Hence the final AUC of the SeCoNet shows higher dependence on the first stage loss coefficients. Similar argument applies for the *bottom branch* i.e. for λ_2 and λ_4. Significant variability for the final AUC is shown for the loss coefficient λ_5. This can be justified as the final AUC metric represents the joint state-object prediction. Hence, the loss coefficient λ_5 has produced strong variability on the final AUC. Next we report the computational complexity of the SeCoNet.

2.4.6.1 Computational Complexity of SeCoNet

To measure the computational complexity, we measure the number of floating point operations (FLOPs) for a single forward pass during training. To calculate the number of FLOPs of all the algorithms, we have used the open source library *fvcore*[2]. To maintain the fairness of evaluation, the number of FLOPs for an algorithm is determined only during training of the network, when both the batch size and the number of negative samples are set to 1.

One single forward pass through the SymNet, LE, AAO, and TMN algorithms require approximately $\sim 2.4 \times 10^8$, $\sim 1.1 \times 10^9$, $\sim 1.8 \times 10^8$, and $\sim 1.3 \times 10^5$ FLOPs respectively. On the contrary, proposed SeCoNet requires $\sim 3.7 \times 10^6$ FLOPs. Thus, the required number of FLOPs for SeCoNet is much lower than that of SymNet, LE, AAO and higher than that of TMN. Hence, we conclude that even though SeCoNet is comprised of two stages and two branches, it has lower computational cost than almost all other existing methods except TMN.

Additionally we have compared the training time of SeCoNet and other state-of-the-art approaches. On MIT-States dataset the training time of AAO, LE, SymNet and TMN algorithms are 3 hours, 3.5 hours, 6 hours and 5 hours respectively. The un-optimized code of SeCoNet requires approximately 5.5 hours of training time. It can be noted that TMN, SymNet and the proposed SeCoNet are the three algorithms with better results as reported in Table 2.2.

Although TMN requires less computational time, SeCoNet reports better AUC than TMN. Each of the two branches (*top* and *bottom branches*) in SeCoNet has two distinct stages. In comparison, the TMN has only one branch with only one stage (Purushwalkam et al., 2019). The individual stages in the two branches have comparable computational cost in comparison to the computational cost of the TMN. The overall architecture in our approach is observed to have higher computational cost due to multiple stages and branches. However, despite having a higher computational complexity, the proposed approach has achieved higher AUC for UT-Zappos50k, MIT-States, and C-GQA datasets compared to the TMN (see Table 2.2). Thus, our approach, despite having higher computational cost, achieves better results over TMN. The relevant details regarding the implementation details and machine configurations are reported in Section 2.4.2. Next, we summarise the work done in this chapter.

2.5 Summary

Here, we introduce a sequential recognition framework for identification of state-object compositions present in an image. The working principle of our novel SeCoNet architecture is primarily based on a two stage sequential recognition approach. Specifically we use the two primitives (the state and object) in a compositional image one after another. Thus Our approach has a linear search space complexity over the joint stare object recognition approach. To the best of our knowledge, the sequential approach of state and object recognition is never emphasized in prior art of CZSL. Next for effective disentanglement of state and the object we proposed a gradient penalization based regularization loss. We also provide a theoretical guarantee of disentanglement between object and image features in the proposed approach.

For the C-GQA dataset, there are 870 object classes and 453 state classes, resulting in a total of 394, 110 state-object compositional classes. However, as shown in Table 2.1, the dataset

[2] https://github.com/facebookresearch/fvcore accessed as on Oct 29, 2022.

contains images of $1,962$ compositional classes. Among the $394,110$ state-object compositional classes, many are impractical in real-world scenarios (e.g., a *rotten chair*). Therefore, a key piece of information implicitly used by the algorithms is that only feasible state-object compositions are included in the model's prediction space. A more generalised solution for CZSL entails that no specific information about feasibility of the compositions should be used by a CZSL algorithm. Thus a more general solution for CZSL should work on the open-world CZSL (OW-CZSL), where no prior information about the feasibility of state-object compositions is available. Consequently, in OW-CZSL, all $394,110$ state-object compositional classes are potential model predictions. To address the OW-CZSL, an algorithm should be able to assess the feasible compositions and weed out the infeasible compositions of itself. In this chapter, our approach does not have any such approach to address the OW-CZSL problem.

In the next chapter, we propose a more generalized solution to the CZSL problem, which should be capable of handling open-world CZSL (OW-CZSL), where no prior information about the feasibility of state-object compositions is available.

Chapter 3

Multi-Branch Graph Convolutional Network with Cross-layer Knowledge Sharing

3.1 Introduction

As discussed in the summary of the last chapter (section 2.5), a generalised solution for CZSL entails to work on the Open-World CZSL (OW-CZSL) problem, where no prior information about the feasibility of a particular state and object is available during training. Thus to work on OW-CZSL problem, we propose a Graph Convolutional Network (GCN) based architecture with frequency based approach for predicting feasibility of unseen state-object composition. Next we report a brief overview of the proposed approach.[1]

In a state-object composition, the object is a physical entity, whereas the state represents a semantic description of the object. For example, the visual features of the state *wet* in the compositions *wet cloth* and *wet glass* vary widely. In order to recognise novel state-object composition, the object and state features need to be disentangled from the composite state-object features. However, the context dependency of the state makes it difficult to disentangle the object feature from the state feature.

Existing works (Nagarajan and Grauman, 2018, Misra et al., 2017) mainly learn features of each object independent of other objects (and so, is for states). However, it can be noted that a select group of objects, say, *apple*, *banana*, *orange* etc., create feasible compositions with a select group of states (for example *rotten*, *ripe*, *peeled*). Hence, we argue that there exists a rich dependency among features of different objects (similarly among features of different states). We therefore propose to exploit this dependency structure by constructing a graph-based architecture to better solve the CZSL problem.

In this chapter, we have proposed a two-branch Graph Convolutional Network (GCN) (Kipf and Welling, 2016) to address the CZSL problem. Each of the two branches consists of a multi-layer GCN architecture. Each branch performs the specific task of object and state recognition separately. The two-branch architecture gives our algorithm the flexibility

[1] A part of the work done in this chapter is published as follows,
Aditya Panda, Dipti Prasad Mukherjee, "Compositional Zero-Shot Learning using Multi-Branch Graph Convolution and Cross Layer Knowledge Sharing", Pattern Recognition, 145 (2024): 109916.

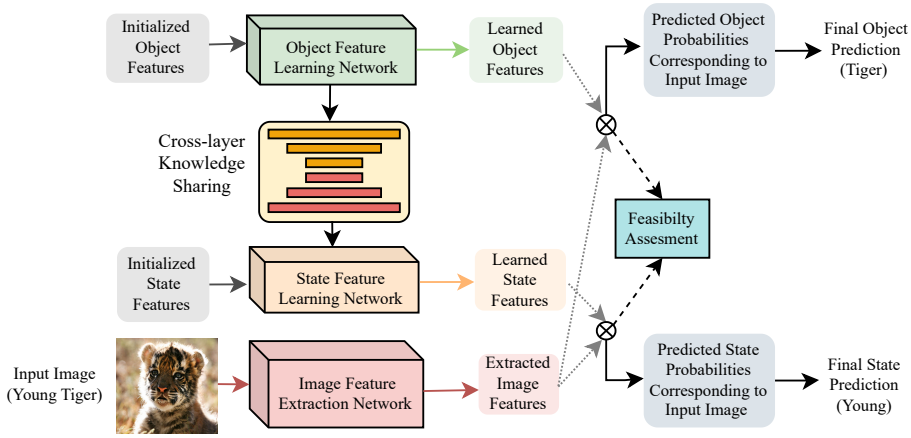


Figure 3.1: A simplified block diagram of our approach: The Object Feature Learning Network (OFLN) and the State Feature Learning Network (SFLN) learn the object and state features, respectively

to independently fine-tune the parameters of each branch of the proposed architecture. This capacity to independently fine tune the state and object branches, helps better disentangle the object and state features.

We have also proposed a novel knowledge sharing strategy between the layer of the object branch and the layer of the state branch. This strategy helps the model to better capture the intricacies of context dependency between object and state features in a composition. In chapter 2, we also propose a two branch approach in our SeCoNet. However SeCoNet is based on the idea of sequential learning, whereas the approach in this chapter is designed to leverage the intricate relationship between visual features of states and objects. Besides we propose a novel cross-layer knowledge sharing strategy in this chapter which was not explored in our prior work.

For the OW-CZSL approach, we utilise the frequency distribution of the state-object compositional classes in the training set to assess the feasibility of the unseen state-object compositions. For each state-object pair, our feasibility assessment strategy assigns a feasibility score $\in [0, 1]$. Feasible pairs are assigned a higher score and infeasible compositions are assigned a lower score. Our model uses a penalization, inversely proportional to the feasibility score as regularizer. The proposed feasibility regularization loss assists the network to weed-out less feasible compositions from the state-object composition prediction space of the network. Finally, our algorithm is tested on the benchmark datasets MIT-States-CL (Isola et al., 2015), MIT-States (Isola et al., 2015), UT-Zappos50k (Yu and Grauman, 2014, 2017) and C-GQA (Naeem et al., 2021). Our approach reports competitive results across the datasets. The key contributions of this chapter can be summarized as,

- We have introduced a novel two-branch GCN based architecture to better exploit the rich dependencies of the visual features between objects and states.
- To better resolve the ambiguity arising out of the context dependency of the visual features of the state, we propose a between-branch Knowledge Sharing Network based on encoder-decoder architecture.

- We have also proposed a novel strategy to assess the feasibility of each unseen object and state composition in the dataset.
- We have performed extensive experiments on the popular benchmark datasets. The proposed algorithm reports competitive results.

Section 3.2 reviews the relevant literature. Section 3.3 discusses proposed algorithm in detail and Section 3.4 reports our results and compares with other state-of-the-art algorithms. Finally, Section 3.5 summarises the chapter.

3.2 Related Works

In this section, we review existing approaches relevant to our approach.

Graph Convolutional Network (GCN): GCNs are a member of a broader family of Graph Neural Networks (GNNs) (Bruna et al., 2014, Defferrard et al., 2016, Kipf and Welling, 2016). Bruna et al. (2014) introduced GNN, and Defferrard et al. (2016) further extended it with an efficient filtering strategy, reducing its computational complexity equivalent to that of the commonly used Convolutional Neural Networks. Kipf *et al.* (Kipf and Welling, 2016) proposed further simplifications to improve scalability and robustness of GNN and applied their approach to semi-supervised learning on graphs. Their approach is commonly referred as GCN and provides the foundation for the model proposed in this chapter. As reported by Monti et al. (2017), CNN struggles to learn and generalize features of non-Euclidean data. However, GCN, instead of performing conventional convolution operation, performs similarity based aggregation of features of the neighbouring nodes and can better learn features from the non-Euclidean data structures (Monti et al., 2017, Bronstein et al., 2017).

Relevant CZSL approaches are already discussed in Section 2.2. Here we review some additional CZSL approaches, relevant to our work in this chapter.

Compositional Zero-shot Learning: Xu et al. (2021b) proposed a network consisting of a concept module and a visual module. The concept module takes state and object label as input query and generates the corresponding state and object features. The visual module extracts the state and object's visual features from an input image. The joint recognition of state and object is performed by evaluating compatibility between state and object features (as obtained from the concept module) and visual features (as obtained from the visual module). Mancini *et al.* (Mancini et al., 2021), measured cosine similarity between word embedding of object and word embedding of states. The obtained cosine similarity is used to predict feasibility of unseen state-object compositions. Recently, Naeem et al. (2021) proposed a GCN based model for the CZSL problem, named CGE. Although CGE also uses a GCN based architecture like the proposed approach, still our approach has many advantages. The GCN in requires nodes for each object, each state, and each feasible state-object composition. In contrast, the GCN in our model does not require nodes for state-object composition. For n number of states and m number of objects, number of nodes in CGE is $\mathcal{O}(m+n+mn) \approx \mathcal{O}(mn)$. In our case number of nodes required in the GCN are $\mathcal{O}(m+n)$. Assuming comparable number of states and objects ($m \approx n$), CGE needs $\mathcal{O}(m^2)$ nodes whereas our approach requires only $\mathcal{O}(m)$ nodes. Further, the approaches in (Ruis et al., 2021, Naeem et al., 2021) have considered a fixed similarity relationship between nodes and there is no scope for updating the similarity relationship based on unseen state-object

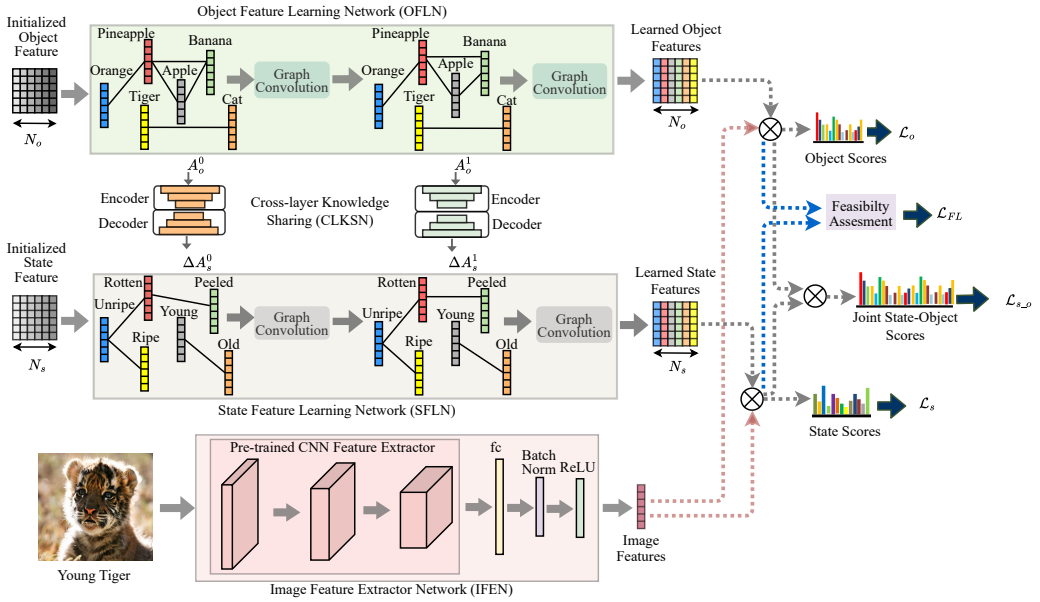


Figure 3.2: The proposed CZSL architecture consists of four major components, the Object Feature Learning Network (OFLN), State Feature Learning Network (SFLN), Image Feature Extractor Network (IFEN) and Cross-layer Knowledge Sharing Network (CLKSN). The implementation details of each of the modules are reported in Section 3.4.1.

compositions. We, instead of taking a fixed similarity relationship based approach, update the similarity based on the between-branch knowledge sharing strategy. Next, we present our approach in detail.

3.3 Methodology

The proposed architecture consists of four major components, namely the Object Feature Learning Network (OFLN), State Feature Learning Network (SFLN), Image Feature Extractor Network (IFEN) and Cross-layer Knowledge Sharing Network (CLKSN) (see Fig. 3.2). Before explaining the detailed working principle of each component of our architecture, we first formally state the problem in the next section.

3.3.1 Problem Statement

The Closed-World CZSL (CW-CZSL) problem is reported in in Section 2.3.1. Here we additionally report the OW-CZSL problem.

Open-world CZSL (*OW-CZSL*): In a CZSL dataset the set of states is represented as $S = \{s_1, s_2, \dots s_{N_s}\}$. Here N_s represents the number of states in the dataset. Similarly, the set of objects is represented as $O = \{o_1, o_2, \dots o_{N_o}\}$, N_o denoting the number of objects in the dataset. Proposed by (Mancini et al., 2021), OW-CZSL assumes no prior information about the feasibility of a particular state-object composition. The number of possible predictions in OW-CZSL is also $N_s \times N_o$. However, in OW-CZSL, for each image in the training set both state and object annotations are provided. Next, we present the proposed approach.

3.3.2 Object Feature and State Feature Learning Networks

The main purpose of the proposed OFLN is to learn the disentangled features for object present in the given training images of state-object compositions. The output of OFLN is the object feature matrix and it is represented as $H_o \in \mathbb{R}^{N_o \times d}$. The number of objects in the dataset is N_o. The feature vector of each object is d dimensional. However, the major challenge in learning this object feature is that the object feature matrix does not necessarily satisfy the Euclidean distance based neighbourhood properties.

Thus the spatial allocation of features of individual objects in H_o may not be explainable using the Euclidean distance between feature vector. Hence, the object feature matrix is considered as a non-Euclidean data structure. CNNs struggle to learn and generalize features from non-Euclidean data (Bronstein et al., 2017, Monti et al., 2017). In this model, we have tried to overcome this limitation and used Graph Convolutional Network (GCN) (Kipf and Welling, 2016) to obtain the object feature matrix, H_o.

OFLN is constructed using L layers of GCN. Each layer of OFLN consists of an undirected graph $G = (V, E)$. The vertex set V contains N_o number of nodes. Each node corresponds to an object in the dataset. Each node in the OFLN has a d dimensional feature vector. The object feature matrix of the l^{th} layer of OFLN is denoted by $H_o^l \in \mathbb{R}^{N_o \times d}$, where $l \in \{1, L\}$. H_o^l is obtained by aggregation of feature vectors from layer $(l-1)$ of neighbouring object nodes. The neighbourhood of an object node in layer l is defined by the adjacency matrix $A_o^l \in [0,1]^{N_o \times N_o}$. The feature matrix at layer $(l+1)$ is evaluated by multiplying the adjacency matrix (A_o^l) with the object feature matrix (H_o^l) at layer l. The resulting graph propagation rule (Kipf and Welling, 2016) is expressed as follows,

$$H_o^l = \sigma(A_o^l H_o^{l-1} W_o^l). \tag{3.1}$$

Here $W_o^l \in \mathbb{R}^{d \times d}$ is the learnable weight parameter matrix for layer l of the OFLN module. The sigmoid operation (Pennington et al., 2017) is represented by $\sigma(\cdot)$.

However, this strategy suffers from a major limitation (Kipf and Welling, 2016). The nodes with large degrees will make feature aggregation from a larger neighbourhood. Accordingly, those nodes will have higher values in their feature representations. Nodes with smaller degrees will have smaller values in feature representation. This may cause vanishing and exploding gradient (Kipf and Welling, 2016) problems.

Thus, in an attempt to solve this limitation, we first obtain the degree matrix (D_o^l) as follows. The degree matrix (D_o^l) is an $N_o \times N_o$ dimensional matrix. The degree of the i^{th} node is placed on the corresponding diagonal cell of the i^{th} row in the matrix D_o^l, $i \in \{1, 2, \ldots N_o\}$. The nodes with larger neighbourhood and higher degree, will have smaller value in the $[D_o^l]^{-1}$ matrix. Hence, multiplying $[D_o^l]^{-1}$ to A_o^l corresponds to taking numerical average of neighbourhood features. Hence the feature vector corresponding to each node is brought to comparable scale, independent of degree of the node, by using $[D_o^l]^{-1} A_o^l$ instead of A_o^l in (3.1). So, the final updated propagation rule is obtained by modifying (3.1) as follows,

$$H_o^l = \sigma([D_o^l]^{-1} A_o^l H_o^{l-1} W_o^l). \tag{3.2}$$

Similar to the OFLN, the SFLN, also, has L number of layers of GCN. Each vertex in each of the L layers of SFLN corresponds to one state in the dataset. The state feature matrix of

the l^{th} layer of SFLN is represented as $H_s^l \in \mathbb{R}^{N_s \times d}$,

$$H_s^l = \sigma([D_s^l]^{-1} A_s^l H_s^{l-1} W_s^l). \tag{3.3}$$

N_s is the number of states in the dataset. A_s^l is the adjacency matrix for layer l in SFLN. $D_s^l \in \mathbb{R}^{N_s \times N_s}$ is the degree matrix of layer l of SFLN.

The last layer feature matrices of the OFLN and SFLN (H_o^L and H_s^L, respectively) are used as the learned object and state features, and used in subsequent stages. In Fig. 3.2, the upper and middle branches represent the OFLN and SFLN, respectively. The OFLN is shown inside the green box in the upper branch. The SFLN is shown inside the yellow box on the middle branch. Next, we explain the Image Feature Extractor Network.

3.3.3 Image Feature Extractor Network

In this section, we extract the image features using a pre-trained CNN (He et al., 2016). A combination of fully connected layers and batch-norm layer (Ioffe and Szegedy, 2015) is used to map the extracted image features to a scale identical to H_o^L and H_s^L. Let the projection function along with the pre-trained CNN be represented as $\mathcal{F} : \mathbb{R}^{3 \times r \times c} \rightarrow \mathbb{R}^d$. For an RGB image $I \in \mathbb{R}^{3 \times r \times c}$, the extracted image features are

$$f_{img} = \mathcal{F}(I), \tag{3.4}$$

where f_{img} represents the latent representation of the image features, $f_{img} \in \mathbb{R}^d$. Here r and c represent the height and the width of the input image. The image feature extractor is shown in the bottom-most branch of the architecture in Fig. 3.2. Next, we explain the Cross-layer Knowledge Sharing Network.

3.3.4 Cross-layer Knowledge Sharing Network

As reported in Section 1.2, in a state-object composition the object represents a physical entity and the state is a semantic description of the object. Hence, the visual features of the state in a state-object composition vary depending on the composition. For example, as shown in Fig. 3.3, the visual features of the state *peeled* in *peeled apple* is not exactly same as the visual features of *peeled* in the compositions *peeled banana* and *peeled orange*. Due to this variation in the features of states, the network struggles to recognise unique features of the state *peeled*. To capture these context dependencies of states corresponding to an object, we propose to introduce knowledge sharing between the object branch and the state branch of our network.

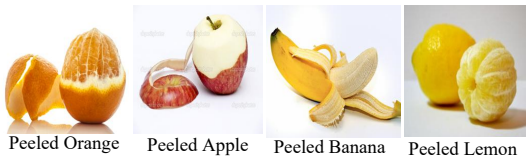


Figure 3.3: Sample images from four classes with state *peeled* from the MIT-States dataset. The state *peeled* causes different visual senses in four different compositions.

It can be observed that a group of similar objects like *apple*, *banana* or *orange* create compositions with a select group of states (for example, *rotten*, *ripe* or *peeled*). The information about objects sharing common states is contained in the adjacency matrix A_o^l of each layer of OFLN.

We propose that object grouping information is useful during learning of the state features in the SFLN. If we share the information that *apple* and *banana* belong to the same group in the context of state *peeled*, it may help to disambiguate the differences in visual features of *peeled* in the compositions *peeled banana* and *peeled apple*. This should help the network to identify the unique representation for the state *peeled*.

Thus, the information of groups within objects must be shared to SFLN. Hence, we define a set of Encoder-Decoder based modules, one for each of the L layers of GCN in OFLN. The proposed knowledge sharing architecture is formally represented as,

$$\left[(Enc_o^1, Dec_o^1), (Enc_o^2, Dec_o^2), \dots (Enc_o^L, Dec_o^L)\right], \tag{3.5}$$

where *Enc* and *Dec* represent encoder and decoder, respectively, each being implement with a single fully-connected layer linked through a bottleneck (refer Section 3.4.1 for details regarding architecture of the encoder-decoder module). The update scheme of the state adjacency matrix A_s^l of layer l is,

$$\begin{aligned} \Delta A_s^l &= Dec_o^l(Enc_o^l(A_o^l)) \\ A_s^l &\leftarrow A_s^l + \Delta A_s^l. \end{aligned} \tag{3.6}$$

With appropriately selecting the output dimensions of decoder module, we can ensure that $\Delta A_s^l \in [0,1]^{N_s \times N_s}$. Hence, A_s^l and ΔA_s^l can be added.

The adjacency matrix may have noise and redundancies. Besides, some objects may not form any group with any other object. Hence, they are typically outlier with respect to any other group. For example, in the C-GQA dataset (Naeem et al., 2021), the state *analog* has only one feasible compositional class in the dataset, *analog clock*. This outlier information acts like noise to the *Knowledge Sharing Network*. Hence, first this noise and redundancies should be minimized before sharing the information from OFLN to SFLN. The purpose of encoder is to minimize redundancy and encode only that information in the adjacency matrix that contains possible grouping information. The de-noised information is up-sampled by decoder network. In Fig. 3.2, the Cross-layer Knowledge Sharing Network is shown. However, we shall like to highlight here that, the knowledge sharing takes place in one way i.e. only from OFLN to SFLN.

Next we briefly describe the architecture of the CLKSN. There is one encoder-decoder pair of CLKSN for sharing knowledge from each layer of OFLN to corresponding layer of SFLN. The encoder is created using one fully connected layer with $N_o \times N_o$ number of input nodes and N_{hid} number of output nodes. The decoder network is created using a similar fully connected layer with N_{hid} number of input nodes and $N_s \times N_s$ number of output nodes. The parameter N_{hid} is investigated in detail in the ablation study (Section 3.4.6). Next, we explain the feasibility prediction strategy for the unseen composition.

3.3.5 Feasibility Prediction Strategy

As explained previously, the final state-object composite class prediction of the model is created as joint prediction of the object branch and the state branch. In case of the MIT-States dataset (Isola et al., 2015), there are 245 object classes and 115 state classes, creating 28175 number of state-object classes, in total. Among the 28175 compositions, most of them are infeasible compositions (ex: *ripe chair*). However, the most widely used train-test split has only 1962 feasible state-object classes (Purushwalkam et al., 2019). Similar argument exists for C-GQA (Naeem et al., 2021) dataset with 453 states and 870 objects and 9378 feasible state-object classes in the split proposed by (Naeem et al., 2021). Thus, the two-branch architecture may predict many unnecessary infeasible compositions initially. Hence, to aid the model in solving the problem, we propose a non-parametric strategy to evaluate a feasibility score ($\in [0,1]$) for each state-object composition. First, we create a *feasibility matrix* $\mathcal{M}_F \in \{0,1\}^{N_o \times N_s}$, as follows,

$$\mathcal{M}_F[i][j] = \begin{cases} 1, & \text{if at least one image with annotation of } i^{th} \text{ object and} \\ & j^{th} \text{ state exists in the training set,} \\ 0, & \text{otherwise,} \end{cases} \tag{3.7}$$

where $i \in \{1, 2, \ldots, N_o\}$ and $j \in \{1, 2, \ldots, N_s\}$. Next, we find the *object similarity matrix* $\mathcal{S}_{obj}$ and *state similarity matrix* $\mathcal{S}_{state}$ as follows,

$$\mathcal{S}_{obj} = \mathcal{M}_F(\mathcal{M}_F)^T \text{ and } \mathcal{S}_{state} = (\mathcal{M}_F)^T\mathcal{M}_F, \tag{3.8}$$

where $\mathcal{S}_{obj} \in \mathbb{R}^{N_o \times N_o}$ and $\mathcal{S}_{state} \in \mathbb{R}^{N_s \times N_s}$. The content of a cell at the intersection of u^{th} column and v^{th} row of the matrix $\mathcal{S}_{obj}$ represents the number of compositions with common states between u^{th} and v^{th} objects.

We suggest that similar objects should always create feasible composition with nearly same set of states. For example, the objects from fruits and vegetable group, like *orange*, *apple*, *banana* etc. share similar set of states like *peeled*, *rotten*, *unripe* etc. Hence, the value in the cell at the intersection of u^{th} column and v^{th} row of the matrix $\mathcal{S}_{obj}$ can be interpreted as the similarity score between u^{th} and v^{th} objects.

Each object has the highest number of common states with itself (if there are images of *peeled apple* and *peeled banana* in the training set, then *peeled* is a common state between the objects *apple* and *banana*). Hence, the diagonal cells of the matrix $\mathcal{S}_{obj}$ have the highest values. This may create the matrix to be imbalanced. Commonly occurring objects have a greater number of instances in the dataset. Naturally, state-object compositions of commonly occurring objects are higher in number in the training set. Hence, these common objects will obtain high similarity values and will dominate the similarity matrix. Whereas less common objects will have less number of compositions and hence smaller values in the similarity matrix.

Hence, to offset such biases, we normalize the $\mathcal{S}_{obj}$ matrix by dividing each of its rows by the corresponding diagonal element in that row, to obtain normalized version of $\mathcal{S}_{obj}$ matrix. Similar process is followed for the matrix $\mathcal{S}_{state}$. Finally, after normalization of the $\mathcal{S}_{obj}$ and $\mathcal{S}_{state}$ matrices, we have $\mathcal{S}_{obj} \in [0,1]^{N_o \times N_o}$ and $\mathcal{S}_{state} \in [0,1]^{N_s \times N_s}$.

The initialized $\mathcal{M}_F$ has 1 corresponding to all compositions available in the training set

and 0 corresponding to all compositions not available in the training set. Thus, the initialised $\mathcal{M}_F$ matrix cannot be considered as a feasibility matrix for test set as $\mathcal{M}_F$ does not include feasibility value for unseen compositions.

Feasibility in case of test set: Let us consider an example where the training set has at least one annotated image of the compositions {*rotten apple*, *rotten orange*, *unripe apple*, *unripe orange*, *unripe banana*, *ripe apple*, *ripe orange*, *ripe banana*, *cracked vessel* and *cracked egg*}. These compositions have feasibility values 1 in $\mathcal{M}_F$, as these compositions belong to the training set.

Let the test set contain images of unseen composition *rotten banana*. The feasibility value of unseen test composition *rotten banana* in $\mathcal{M}_F$ is 0. However, compositions like *rotten banana* should have high feasibility value, as close as possible to the maximum value 1. We utilise the following strategy to update the values in $\mathcal{M}_F$ for unseen compositions like *rotten banana*.

First, we find out the states which have produced feasible compositions with the object *banana*. We define it *state_affordance*. The group of similar states creating feasible compositions with the object *banana* is referred to as *state_affordance*{banana}.

From the example above, *state_affordance*{banana}={unripe, ripe}. Next, utilising the $\mathcal{S}_{state}$ matrix, we find the similarity between states from the set *state_affordance*{banana} and the test state *rotten*. Let us denote the set of similarity values as

$$score_{banana} = \{\mathcal{S}_{state}[j_{rotten}][j_{unripe}], \mathcal{S}_{state}[j_{rotten}][j_{ripe}]\}, \tag{3.9}$$

where j_{rotten}, j_{unripe} and j_{ripe} indicate the indices of states *rotten*, *unripe* and *ripe* in the set of states respectively.

Similarly, we find out which objects have produced feasible composition with the state *rotten*. We define it as *object_affordance*{rotten}={apple, orange}. Next, from the $\mathcal{S}_{obj}$ matrix, we find the similarity between states from the set *object_affordance*{rotten} and the test object *banana*. Let us denote the set of similarity values as

$$score_{rotten} = \{\mathcal{S}_{obj}[i_{banana}][i_{apple}], \mathcal{S}_{obj}[i_{banana}][i_{orange}]\}, \tag{3.10}$$

where i_{banana}, i_{apple} and i_{orange} indicate the indices of objects *banana*, *apple* and *orange* in set of objects respectively. The feasibility of the unseen composition *rotten banana* is then represented as follows.

$$\mathcal{M}_F[i_{banana}][j_{rotten}] = \frac{max\{score_{banana}\} + max\{score_{rotten}\}}{2}. \tag{3.11}$$

Here $max\{score_{banana}\}$ represents the element with maximum value from the set $score_{banana}$. $max\{score_{banana}\}$ and $max\{score_{rotten}\}$ represent the feasibility of *banana* and *rotten* being present in the unseen composition *rotten banana*, respectively. The scores $max\{score_{banana}\}$ and $max\{score_{rotten}\}$ are averaged to obtain the final feasibility score of the composition *rotten banana*. For generalised case, this process of feasibility estimation is repeated till all the unseen state-object compositions are covered. This approach is formally written in the form of pseudo-code in Algorithm 1. Next, we explain the loss components for the network.

Algorithm 1 Feasibility Assessment Algorithm

Input : Feasibility Matrix $\mathcal{M}_F \in \mathcal{R}^{N_o \times N_s}$
Output : Updated Feasibility Matrix
Initialization : Initialize the $\mathcal{M}_F$ for seen compositions using (3.7)
Create the object and state similarity matrices $\mathcal{S}_{obj}$ and $\mathcal{S}_{state}$ using (3.8);
Obtain the normalized object and state similarity matrices by dividing each row by the corresponding diagonal element;
for $i \leftarrow 0$ **to** $N_o - 1$ **do**
  for $j \leftarrow 0$ **to** $N_s - 1$ **do**
    if $\mathcal{M}_F[i][j] == 1$ **then**
      skip
    else
      For the object i, find the set of states such that object i has formed feasible composition with each such state in the training set. Represent the indices of the set of states as *state_affordance*$\{i\} = \{j_{\alpha_s}, j_{\beta_s}, \ldots, j_{\gamma_s}\}$;
      Evaluate $score_{object[i]} = \{\mathcal{S}_{state}[j][j_{\alpha_s}], \mathcal{S}_{state}[j][j_{\beta_s}], \ldots, \mathcal{S}_{state}[j][j_{\gamma_s}]\}$
      For the state j, find the set of objects such that state j has formed feasible composition with each such object in the training set. Represent the indices of the set of states as *object_affordance*$\{j\} = \{i_{\alpha_o}, i_{\beta_o}, \ldots, i_{\gamma_o}\}$;
      Evaluate $score_{state[j]} = \{\mathcal{S}_{obj}[i][i_{\alpha_o}], \mathcal{S}_{obj}[i][i_{\beta_o}], \ldots, \mathcal{S}_{obj}[i][i_{\gamma_o}]\}$
      Evaluate $\mathcal{M}_F[i][j] = \frac{max\{score_{obj[i]}\} + max\{score_{state[j]}\}}{2}$
    end
  end
end
return Updated Feasibility Matrix ($\mathcal{M}_F$)

3.3.6 Loss Components

We define loss function as a combination of four components, object cross-entropy, state cross-entropy, state-object joint cross-entropy and the feasibility loss. Each of these components is detailed below.

Object cross-entropy loss: For each image, during training, image feature (f_{img}) is extracted from the Image Feature Extractor Network. We multiply the object feature matrix (H_o^L) with image feature (f_{img}) to get the object score vector $E_{o\_I} = H_o^L f_{img}$, ($E_{o\_I} \in \mathbb{R}^{N_o}$). The object score vector is used to train the model through cross entropy loss with respect to object ground truth (GT_o) annotation.

$$\mathcal{L}_o = \mathcal{L}_{CE}(E_{o\_I}, GT_o). \tag{3.12}$$

State cross-entropy loss: Following a similar approach as above, we evaluate the state score vector as $E_{s\_I} = H_s^L f_{img}$, ($E_{s\_I} \in \mathbb{R}^{N_s}$). The loss component for state prediction is evaluated using state ground truth annotation (GT_s) and cross entropy loss is evaluated as follows,

$$\mathcal{L}_s = \mathcal{L}_{CE}(E_{s\_I}, GT_s). \tag{3.13}$$

State-object joint cross-entropy loss: The visual features of images of each state-object composition consist of three components, features specific to object, features specific to state and finally, features created due to the state-object composition.

Training a model with the object and state losses, helps the model to better recognise the discriminative state and discriminative object features. Identifying the discriminative state and the object features help the model to better recognise unseen compositions. The third component of features are unique to each state-object composition and helps the model to better recognise the seen compositions (note that in CZSL the test set comprises of both

images from seen as well as unseen compositions). Using a state-object joint loss helps to better recognise the third component of the features. So, we add the joint state-object loss.

The object score vector, E_{o_I} is passed through a softmax layer to obtain the object probability vector ($p(o) \in \{0,1\}^{N_o}$) corresponding to the input image. The object probabilities are evaluated as follows,

$$p(o)[i] = \frac{\exp\Big(E_{o\_I}[i]\Big)}{\sum_{i'=1}^{N_o} \exp\Big(E_{o\_I}[i']\Big)}, i \in \{1, 2, \ldots, N_o\}. \tag{3.14}$$

Similarly, the state scores E_{s_I} are passed to a softmax layer to obtain the state probability vector ($p(s) \in \{0,1\}^{N_s}$). Next we multiply the state probability and object probability to get the corresponding state-object probability and in this way we create the state-object probability vector ($p(s\_o) \in \{0,1\}^{N'}$), where N' is the number of state-object compositions present in the dataset. Using the ground truth of the state-object composition (GT_{s_o}), we evaluate the state-object joint loss as follows,

$$\mathcal{L}_{s\_o} = \mathcal{L}_{CE}(p(s\_o), GT_{s\_o}). \tag{3.15}$$

Feasibility loss: For a given image, the model's final prediction of the state-object compositional class is taken from the predictions of individual state and individual object. These predictions may give rise to a prediction of infeasible combination of state-object pair. Hence, we add a feasibility penalization to the loss term to restrict the model's prediction space to only feasible predictions.

For a given image, let the object prediction be i^{th} object and state prediction be j^{th} state. The feasibility score is obtained from the feasibility matrix $\mathcal{M}_F[i][j]$. The feasibility penalization is inversely proportional to the feasibility score. Thus, we define the feasibility loss as,

$$\mathcal{L}_{FL} = \log_2\left(\frac{1}{\mathcal{M}_F[i][j]}\right). \tag{3.16}$$

The values in $\mathcal{M}_F$ are calculated using Algorithm 1 as a pre-processing step before the training of the proposed model. The obtained $\mathcal{M}_F$ serves as ground truth scores for feasibility evaluation of the predictions. The $\mathcal{M}_F$ is calculated using dataset statistics. Since dataset does not change during training, $\mathcal{M}_F$ is also not updated during training. Using this $\mathcal{L}_{FL}$ during training, the model is expected to learn the prediction of feasible compositions. The total loss is computed as follows,

$$\mathcal{L} = \lambda_1\mathcal{L}_s \ + \lambda_2\mathcal{L}_o + \lambda_3\mathcal{L}_{s\_o} + (1 - \lambda_1 - \lambda_2 - \lambda_3)\mathcal{L}_{FL}, \tag{3.17}$$

where λ_1, λ_2 and λ_3 are scalar parameters within [0,1] and $\lambda_1 + \lambda_2 + \lambda_3 < 1$. Next, we explain the inference strategy for the proposed algorithm.

3.3.7 Inference Strategy

During inference, the OFLN and SFLN are not used. Instead, the learned object feature matrix (H_o^L) and state feature matrix (H_s^L) are used. The features of the test image are extracted by the Image Feature Extractor Network. The extracted features are multiplied

with the H_o^L and H_s^L to obtain the object score vector (E_{o_I}) and the state score vector (E_{s_I}). The object prediction in the test image is taken as the object having maximum score in the object score vector. Similarly, the final state in the test image is predicted as the state having maximum score in the state score vector. Next, we present the experimentation details.

3.4 Experiments

3.4.1 Implementation Details

The object and state feature vectors are initialized with 300 dimensional FastText (Bojanowski et al., 2017) word embedding of the corresponding object and state labels. Inside the OFLN we initialize the object adjacency matrix of all the layers (A_o^l) using $\mathcal{S}_{obj}$ of (3.8). Similarly, inside the SFLN we initialize the state adjacency matrix of all the layers (A_s^l) using $\mathcal{S}_{state}$ of (3.8). The maximum value of the loss component $\mathcal{L}_{FL}$ in (3.16) is fixed to be a constant value of 10. We use Adam optimizer (Kingma and Ba, 2015) with learning rate 0.00005. The scalar weights for loss components are $\lambda_1 = 0.4, \lambda_2 = 0.2$ and $\lambda_3 = 0.3$.

For all the datasets, we train the model for 60 epochs. We have used early stopping, based on the performance of the model on the validation set. The Image Feature Extractor Network consists of ResNet-18 (He et al., 2016) (pre-trained on ImageNet (Russakovsky et al., 2015) dataset) upto *conv_5_4* layer. The remaining layers consist of a fully connected layer, a batch-norm layer (Ioffe and Szegedy, 2015) and a ReLU layer. Next we describe the architecture of the encoder-decoder based Cross-layer Knowledge Sharing Network (CLKSN). There is one encoder-decoder pair of CLKSN for sharing knowledge from each layer of OFLN to corresponding layer of SFLN. As both OFLN and SFLN have two layers of GCN, we have two sets of encoder-decoder pair in CLKSN. The encoder is created using one fully connected layer with $N_o \times N_o$ number of input nodes and N_{hid} number of output nodes. The decoder network is created using a similar fully connected layer with N_{hid} number of input nodes and $N_s \times N_s$ number of output nodes. The parameter N_{hid} is chosen through ablation study described later. For UT-Zappos50k (Yu and Grauman, 2014, 2017), $N_{hid} = 4$. For MIT-States (Isola et al., 2015), MIT-States-CL and C-GQA (Naeem et al., 2021) datasets, we set $N_{hid} = 250$. We initialise the weights of the fully connected layers in the encoder and decoder of CLKSN using Kaiming initialization (He et al., 2015). The batch size of 256 is used for MIT-States-CL, MIT-States and C-GQA, whereas for UT-Zappos50k dataset batch size of 32 is used. The parameters of the models are selected based on cross-validation in the validation set.

The model is implemented using PyTorch on a NVIDIA RTX Titan GPU with 24 GB memory, CUDA 10.1 and cuDNN 7.6. The training of the proposed algorithm takes nearly 6, 16, 15 and 20 hours for UT-Zappos50k, MIT-States, MIT-States-CL and C-GQA datasets, respectively, with an un-optimized Python 3.6 code on a PC with Intel i9 3.3 GHz processor, 64 GB RAM and Linux operating system. Next, we present the details of the dataset used for evaluation.

3.4.2 Datasets

Our experiments are carried out on three popular benchmark datasets: MIT-States (Isola et al., 2015), UT-Zappos50k (Yu and Grauman, 2014, 2017) and C-GQA (Naeem et al., 2021).

Split →			Train		Validation			Test		
Dataset ↓	State	Obj	Seen Class	Img	Seen Class	Unseen Class	Img	Seen Class	Unseen Class	Img
MIT-States	115	245	1262	30k	300	300	10k	400	400	13k
MIT-States-CL	95	242	709	19k	176	159	6k	194	141	7.5k

Table 3.1: Statistics for the newly proposed MIT-States-CL dataset in comparison to the MIT-States dataset.

Details of these datasets are referred in Section 2.1. MIT-States (Isola et al., 2015) dataset has been extensively used in relevant literature (Purushwalkam et al., 2019, Li et al., 2020, Misra et al., 2017, Nagarajan and Grauman, 2018). But, there are many challenges in using MIT-States dataset for training a model. The images of this dataset are labeled automatically. As a result, there are a number of incorrect annotations in the MIT-States dataset. The noise issues of MIT-States dataset due to incorrect annotation have been extensively explored (Atzmon et al., 2020a, Naeem et al., 2021). The authors in (Atzmon et al., 2020a) have used Amazon Mechanical Turk to quantify correctness of the labels of the images using human raters. Only 32% of the raters selected the correct state for their first choice (top-1 accuracy), and only 47% of the raters had the correct state in one of their choices (top-2 accuracy). Interested readers are referred to (Atzmon et al., 2020a) for further details.

To facilitate fair evaluation of our as well as existing algorithms, we propose a cleaner subset of MIT-States dataset and refer it as MIT-States with Clean Labels or MIT-States-CL. The MIT-States-CL is produced from MIT-States after removing images of a few classes with higher number of wrong annotations. We also propose a modified train-test split on the MIT-States-CL dataset. The incorrect annotations are mainly in terms of state labels. Brief description of MIT-States-CL dataset in comparison to MIT-States is reported in Table 3.1.

3.4.3 Compared Algorithms

In this section, we compare our algorithm against the following state-of-the-art algorithms: LE (Misra et al., 2017), AAO (Nagarajan and Grauman, 2018), TMN (Purushwalkam et al., 2019), SymNet (Li et al., 2020), CGE (Naeem et al., 2021). The brief review of these algorithms have been done in Sections 2.2 and 3.2. The results for all these methods are reproduced from the work in (Naeem et al., 2021) and the corresponding open source implementations. The evaluation protocol of the proposed approach is reported in Section 2.4.4. Next we report the results of our approach.

3.4.4 Results

The results for the proposed algorithm on UT-Zappos50k, MIT-States, MIT-States-CL and C-GQA are reported in Tables 3.2 and 3.3. The results corresponding to 'Prop. Algo' represent performance of our algorithm. Results corresponding to 'Prop. Algo. w/o FT' represent performance of our algorithm without fine-tuning the pre-trained feature extractor. Our algorithm reports better AUC metric over state-of-the-art algorithms on UT-Zappos50k, C-GQA and MIT-States-CL datasets. We report improvements in top-1 *test* AUC of 3.2% on UT-Zappos50k, 1.3% on C-GQA, and 2.0% on MIT-States-CL dataset over the existing state-of-the-art approaches. Similarly, on unseen classes of UT-Zappos50k, our algorithm achieves

Dataset →	UT-Zappos50k											MIT-States										
	val AUC			*test AUC*								*val AUC*			*test AUC*							
Algorithms ↓	*top-1*	*top-2*	*top-3*	*top-1*	*top-2*	*top-3*	*seen*	*unseen*	*HM*	*state*	*obj*	*top-1*	*top-2*	*top-3*	*top-1*	*top-2*	*top-3*	*seen*	*unseen*	*HM*	*state*	*obj*
LE	26.4	49.0	66.1	25.7	52.1	67.8	53.0	61.9	41.0	41.2	69.2	3.0	7.6	12.2	2.0	5.6	9.4	15.0	20.1	10.7	23.5	26.3
AAO	21.5	44.2	61.6	25.9	51.3	67.6	59.8	54.2	40.8	38.9	69.6	2.5	6.2	10.1	1.6	4.7	7.6	14.3	17.4	9.9	21.1	23.6
TMN	36.8	57.1	69.2	29.3	55.3	69.8	58.7	60.0	45.0	40.8	69.9	3.5	8.1	12.4	2.9	7.1	11.5	20.2	20.1	13.0	23.3	26.5
SymNet	25.9	50.9	64.5	23.9	48.2	64.4	53.3	57.9	39.2	40.5	71.2	4.3	9.8	14.8	3.0	7.6	12.3	24.4	25.2	16.1	26.3	28.3
CGE	43.2	64.5	77.7	33.5	64.2	77.5	64.5	71.5	60.5	48.7	76.2	**8.6**	**17.6**	**24.9**	**6.5**	**14.7**	**21.3**	**32.8**	**28.0**	**21.4**	**30.1**	**34.7**
Prop. Algo. w/o FT	44.8	65.8	79.0	35.1	65.7	78.8	65.8	73.0	61.3	50.3	77.8	8.0	14.9	21.7	4.9	10.1	14.6	28.6	27.8	17.4	26.4	29.6
Prop. Algo.	**45.9**	**66.9**	**80.8**	**36.7**	**67.0**	**80.1**	**66.1**	**74.5**	**63.2**	**51.9**	**79.4**	8.4	16.8	24.8	5.7	12.8	16.9	30.5	26.4	19.1	30.0	32.9

Table 3.2: CW-CZSL results on the UT-Zappos50k and MIT-States

Dataset →	MIT-States-CL											C-GQA										
	val AUC			*test AUC*								*val AUC*			*test AUC*							
Algorithms ↓	*top-1*	*top-2*	*top-3*	*top-1*	*top-2*	*top-3*	*seen*	*unseen*	*HM*	*state*	*obj*	*top-1*	*top-2*	*top-3*	*top-1*	*top-2*	*top-3*	*seen*	*unseen*	*HM*	*state*	*obj*
LE	4.6	9.3	15.6	5.3	8.8	12.9	17.4	24.0	13.1	26.1	28.9	0.9	2.2	3.2	0.4	1.0	1.5	11.8	4.4	4.6	17.7	19.9
AAO	4.1	8.8	14.9	4.4	8.0	12.0	18.3	19.0	12.2	23.0	25.8	0.8	2.2	3.4	0.4	1.1	1.7	11.4	4.8	4.5	18.0	20.5
TMN	4.8	10.0	16.2	6.0	10.1	15.6	22.4	23.0	16.4	25.1	28.4	1.7	4.2	6.4	0.8	1.8	2.9	18.8	6.1	6.4	16.5	25.6
SymNet	5.6	11.2	18.6	6.4	12.2	18.8	26.1	27.4	18.0	28.5	30.9	2.9	5.1	7.4	1.0	2.3	3.3	20.6	7.0	7.3	21.4	25.1
CGE	8.6	11.9	18.4	7.8	15.9	19.8	34.1	30.8	23.9	32.8	36.4	5.3	10.7	14.6	2.5	4.6	6.4	23.4	7.9	8.9	18.7	29.7
Prop. Algo. w/o FT	9.8	12.7	19.0	8.6	16.0	23.6	34.4	31.4	24.0	33.2	34.7	5.4	11.0	14.9	2.8	4.9	6.7	24.0	9.0	9.0	19.4	28.4
Prop. Algo.	**11.5**	**13.8**	**19.9**	**9.8**	**17.8**	**26.7**	**36.0**	**32.9**	**25.8**	**35.4**	**38.3**	**5.6**	**12.0**	**15.4**	**3.8**	**5.9**	**7.4**	**24.8**	**9.4**	**9.9**	**22.1**	**31.0**

Table 3.3: CW-CZSL results on MIT-States-CL and C-GQA

at least 10.5% better accuracy over the earlier algorithms (LE, AAO, TMN, SymNet). In comparison to the most recent algorithm CGE, our algorithm reports better result on UT-Zappos50k. C-GQA dataset comprises of three times more objects and states than MIT-States dataset (see the dataset statistics reported in Table 2.1). Consequently, the number of possible state-object compositional classes of C-GQA is increased multiple times over MIT-States. Hence, earlier algorithms (SymNet, LE, AAO and TMN) perform poorly on C-GQA dataset. However, our algorithm achieves better AUC as well as better seen and unseen class recognition accuracy over all the other algorithms on C-GQA dataset. On MIT-States, we have achieved second best AUC value.

As reported earlier, one of the major components of our algorithm is the CLKSN (Section 3.3.4). The CLKSN attempts to discover groups among objects in the dataset. This information about group is used for reducing ambiguity in state recognition. We have achieved better accuracy in state recognition with at least 2.2% improvement on all the three datasets, excluding MIT-States. The improvement in state recognition accuracy justifies that the proposed CLKSN is helpful in resolving ambiguity of state features. The CLKSN fails due to label noise in MIT-States (refer to Section 3.4.6 for detailed experiment on CLKSN). Hence proposed algorithm could not report better results on MIT-States. However, proposed algorithm is able to produce better results than all other competitive algorithms on the cleaner MIT-States-CL dataset.

The validation set in all the datasets are prepared following the train-test split proposed by Purushwalkam et al. (2019) and Naeem et al. (2021). The validations set consists of less number of seen and unseen state-object compositional classes in comparison to the test set (see Table 2.1). Hence, almost all the algorithms including ours have achieved better

Dataset →	MIT-States-CL						C-GQA					
Algorithms ↓	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*
LE	0.8	15.6	3.9	2.0	16.4	20.3	0.1	10.2	1.7	1.0	16.4	18.1
AAO	1.2	17.2	4.7	2.1	18.3	22.8	0.2	11.1	1.8	2.5	17.6	18.8
TMN	1.3	20.0	5.2	3.4	21.5	24.3	0.3	16.3	2.0	2.7	18.2	21.1
SymNet	1.4	22.2	6.8	5.0	23.5	26.8	0.4	19.7	2.2	3.2	18.6	23.9
CGE	1.6	23.1	7.1	6.2	24.2	27.4	0.4	23.8	2.4	3.3	19.7	27.1
Prop. Algo.	**1.9**	**26.2**	**8.9**	**7.0**	**25.4**	**28.3**	**0.5**	**25.4**	**2.5**	**3.4**	**20.9**	**28.4**

Table 3.4: OW-CZSL results of our approach

performance on the validation set in comparison to the test set.

The results in Tables 3.2 and 3.3 show that our algorithm achieves better performance if the pre-trained feature extractor is re-trained or fine-tuned on the dataset under consideration. Even without fine-tuning the pre-trained feature extractor, proposed model achieves very competitive results.

Evidently, the OW-CZSL is much more challenging than the CW-CZSL due to large number of possible outcomes. However, as shown in Table 3.4, proposed approach performs well in OW-CZSL. This can be justified as follows. The main challenge in OW-CZSL over the CW-CZSL is the unavailability of feasibility information of state-object compositions. However, proposed approach has specific feasibility assessment strategy (see Section 3.3.5) which can better tackle the unavailability of feasibility information in OW-CZSL.

We have shown a few qualitative classification results of proposed approach on the MIT-States, UT-Zappos50k and C-GQA datasets in Fig 3.4. Fig 3.4 shows both successful and unsuccessful classification results on the three datasets, as mentioned above.

Further, we also investigate the out-of-domain qualitative classification performance of proposed model on ImageNet dataset (Russakovsky et al., 2015) in Fig. 3.5. The proposed model, after being trained on the MIT-States dataset, is evaluated on a set of images from the validation set of ImageNet dataset. We first select a query state-object compositional label (for example *coiled snake*) from the MIT-States dataset. We then pass all the images from the validation set of ImageNet dataset to proposed model. Next, images for which our model predicts highest confidence score corresponding to the query compositional label are noted. These images are reported as qualitative classification results corresponding to the query compositional label. Fig. 3.5 illustrates a few qualitative image classification results for a few state-object compositional labels. Evidently, excluding a failure case in the *ripe fruit* class, all the other top-5 qualitative classification results are successful. Hence proposed algorithm is able to generalize well in other datasets (in this case ImageNet (Russakovsky et al., 2015)) over which training is not performed.

The training time required by LE, AAO, TMN, SymNet and CGE on MIT-States dataset are approximately 4 hours, 5.5 hours, 6.25 hours, 15.25 hours and 17 hours, respectively. Our approach requires approximate training time of 16 hours. The algorithms reporting better results, require additional computation time (ex: SymNet, CGE and our approach). Relevant information about the machine configuration is reported in Section. Next we present the experimental study on the CLKSN.

Figure 3.4: First, second and third rows from top correspond to successful and unsuccessful qualitative classification results on MIT-States, UT-Zappos50k and C-GQA, respectively. The ground truth class label for each image is mentioned below the image. The green and red text represent successful and unsuccessful class label prediction, respectively.

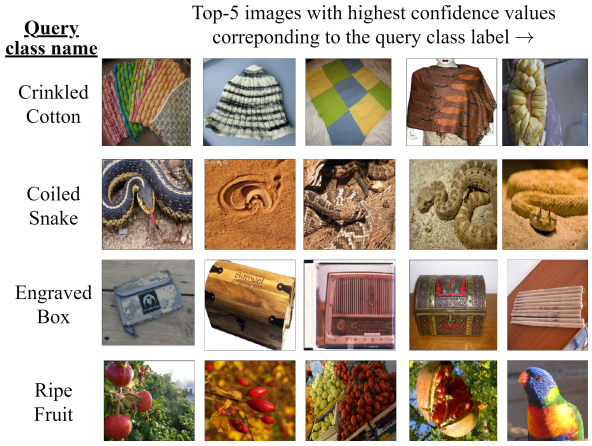


Figure 3.5: Qualitative image classification results from ImageNet (Russakovsky et al., 2015)

3.4.5 Group Information

One of the main contributions of proposed algorithm is the Cross-layer Knowledge Sharing Network (CLKSN). The CLKSN module helps to utilise the presence of groups amongst the objects in the dataset, and thereby, to reduce ambiguity in state recognition. In this section, we try to assess whether the proposed strategy is successful in improving the state recognition capability of our model.

For each state present in C-GQA dataset, we count the objects with which the state has formed feasible compositions. For C-GQA dataset there are states like *analog* which have only one feasible state-object composition in the dataset viz. *analog clock*. Similarly the state *boiled* has only one feasible composition in C-GQA dataset viz. *boiled egg*. On the contrary, there are some states like *white* which has formed 435 number of state-object

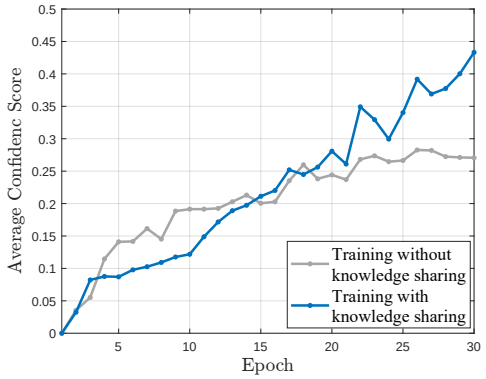


Figure 3.6(a): The average confidence score of the model w.r.t. epochs on C-GQA

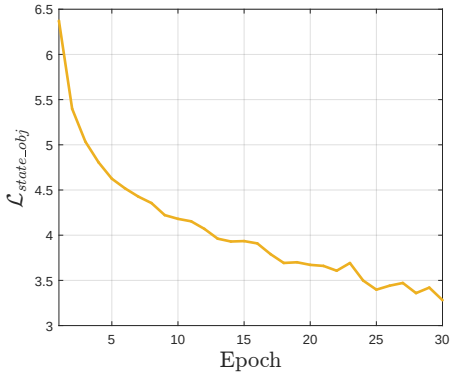


Figure 3.6(b): The variation of $\mathcal{L}_{s\_o}$ (see (3.15)) over the epochs for MIT-States

feasible compositions like *white snow, white saucer, white cup, white toilet, white wall, white backpack, white building, white shirt, white train, white shirt, white watch, white sweater* etc. It is more difficult for an CZSL algorithm to recognise states that create feasible compositions with multiple objects. In this context we calculate the median value of the number of objects associated with a state in the C-GQA dataset. The median value is observed to be 6. In the next step, we create a list of states with at least 6 feasible state-object compositions and this list is referred to as *commonly_occurring_state_list*.

After each epoch of training, we evaluate the model on the images whose states belong to *commonly_occurring_state_list*. We denote the above mentioned set of images as I'. For each image from set I', we calculate the model's confidence score which is the final softmax score corresponding to the ground truth state. We plot the model's confidence score averaged over all the images from the set I' against training epochs. The plots are shown in Fig. 3.6a. The graph in gray colour represents the performance of the model by excluding the knowledge sharing strategy. The graph in blue colour represents the performance of the model including the knowledge sharing strategy. Fig. 3.6a shows that without CLKSN, the model

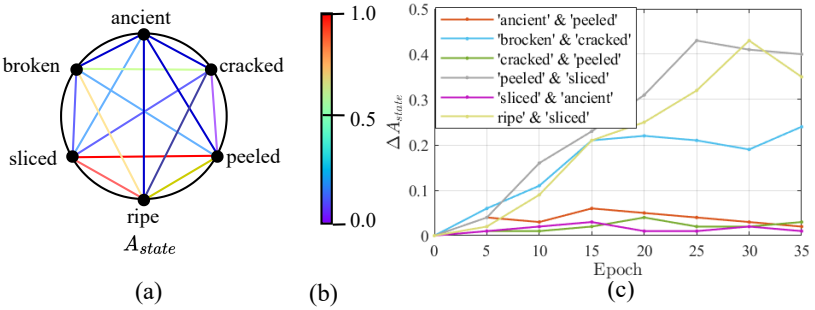


Figure 3.7: The visualisation of the effect of CLKSN and the corresponding adjacency values A_s as updated in (3.6). All A_s and ΔA_s values are obtained from the last layer i.e. second layer of the SFLN for MIT-States dataset. Diagram on the left represents the A_s. Fig. 3.7(c) shows variation of the ΔA_s values between states with epoch

is less confident on recognising states of images. The gray graph in Fig. 3.6a (corresponds to training the model without CLKSN) not only achieves lower confidence score, but also saturates the model's performance early. Hence, proposed CLKSN is helpful in reducing

Model	AUC on UT-Zappos50k	AUC on MIT-States-CL	AUC on MIT-States	AUC on C-GQA
with Object Knowledge and without State Knowledge	**45.9**	**11.5**	**8.4**	**5.6**
without Object Knowledge and with State Knowledge	39.4	7.2	8.2	3.9
without State Knowledge and without Object Knowledge	39.2	8.5	8.1	4.6
with State Knowledge and with Object Knowledge	42.8	10.5	7.3	4.7
without State Knowledge and with Object Knowledge and without $\mathcal{L}_{FL}$	39.3	9.9	7.6	4.5

Table 3.5: Ablation study on the use of CLKSN.

state ambiguity.

In order to justify the efficacy of our CLKSN module, we visualize the adjacency values between different states in Fig. 3.7(a). For the sake of simplicity we have shown the adjacency relationships between six states from MIT-States dataset (Isola et al., 2015), namely, *ancient*, *cracked*, *peeled*, *ripe*, *sliced* and *broken*. Assuming these six states are placed on the circumference of a circle, the adjacency values A_s between a pair of states out of these six states are shown using a line whose colour is given by the colour bar shown in Fig. 3.6a. It can be seen that *ancient* and *peeled* are loosely connected or loosely adjacent as there is no object that forms feasible compositions with both *ancient* and *peeled*. But *sliced* and *peeled* are strongly connected as there are multiple objects (like *apple* or *banana*) that form feasible compositions with states *sliced* and *peeled*. Fig. 3.7(c) shows the variation of the ΔA_s values with respect to epoch. The ΔA_s value between states like *peeled* and *sliced* has reached high values. That is, strongly adjacent state pairs are incremented significantly compared to loosely adjacent state pairs. Hence the results in Fig. 3.7 confirm that the proposed CLKSN module is successful in capturing the adjacency between states which form feasible compositions with similar set of objects. Next we present the ablation study.

3.4.6 Ablation Study

The proposed algorithm comprises of multiple modules. In this section, we report the effectiveness of each module in the final model.

The result in Table 3.5 represents the effect of different variations of the Cross-layer Knowledge Sharing Network (CLKSN) on the model's AUC performance. In Table 3.5, 'Model with Object Knowledge' corresponds to the strategy mentioned in Section 3.3.4 (strategy of sharing the group information available inside the adjacency matrix of OFLN to the corresponding layer of SFLN.). 'Model with State Knowledge' corresponds to the strategy of sharing the group information available inside the adjacency matrix of SFLN to the corresponding layer of the OFLN. The model performs best across all four datasets only when the knowledge is shared from OFLN to SFLN.

As mentioned earlier, in a state-object composition, the object represents a physical entity whereas the state represents a semantic description of the object. Hence, recognising an object is easier than recognising the state in the state-object compositional image. Thus, sharing the information about group of states forming feasible compositions with objects to OFLN, does not help much to reduce the ambiguity of the object recognition.

Comparison between MIT-States and MIT-States-CL dataset shows that higher improve-

Model	AUC on UT-Zappos50k	AUC on MIT-States-CL	AUC on MIT-States	AUC on C-GQA
with all loss components	**45.9**	**11.5**	**8.4**	**5.6**
without $\mathcal{L}_o$	34.4	9.7	7.4	5.4
without $\mathcal{L}_s$	36.6	9.5	7.2	4.9
without $\mathcal{L}_{s\_o}$	29.6	8.7	5.9	4.6
without $\mathcal{L}_{FL}$	36.3	9.7	8.1	5.5
without $\mathcal{L}_{FL}$, $\mathcal{L}_o$ and $\mathcal{L}_s$	33.1	8.9	6.4	4.7
without $\mathcal{L}_{FL}$ and $\mathcal{L}_{s\_o}$	29.4	8.6	5.9	4.5
without $\mathcal{L}_o$ and $\mathcal{L}_s$	33.4	9.1	6.5	4.7

Table 3.6: Ablation study on different loss components

ment in the final AUC is reported on MIT-States-CL when CLKSN is present during training. As already mentioned, there are many incorrect labels in MIT-States dataset. These incorrect labels in MIT-States fail to provide any useful information to the CLKSN. However, while preparing MIT-States-CL dataset, we have manually checked and removed the wrongly annotated images. Due to less amount of noise in MIT-States-CL dataset CLKSN introduces improvement in final AUC values for MIT-States CL dataset.

The bottom-most row of Table 3.5 reports the performance of the model on validation set while trained excluding the CLKSN and $\mathcal{L}_{FL}$. The proposed model reports low AUC in this case. The recognition capability of our algorithm depends largely on learning the adjacency relationship between states and objects. Excluding CLKSN during training in our model is equivalent to keeping a fixed adjacency relationship between states and objects. This fixed adjacency relationship results in inferior results of our algorithm.

The result of Table 3.6 compares the effect of the loss components used in (3.17). The strongest improvement on the results of the proposal is shown by the state-object loss ($\mathcal{L}_{s\_o}$). The sixth row in Table 3.6 shows the performance of the model without $\mathcal{L}_{FL}$ while the object and state knowledge sharing are included during training. It can be seen that including $\mathcal{L}_{FL}$ during training helps the model achieve better AUC on validation set of all the four datasets. This justifies the utility of $\mathcal{L}_{FL}$ loss term in (3.17).

The convergence graph of $\mathcal{L}_{s\_o}$ is shown in Fig. 3.6b. For an input image, the $\mathcal{L}_{FL}$ value is evaluated as the reciprocal of the feasibility of the predicted state-object composition as obtained from $\mathcal{M}_F$ (see (3.16)). To see the effectiveness of the evaluated $\mathcal{L}_{FL}$, we report the feasibility of a few *unseen* state-object compositions as obtained from $\mathcal{M}_F$. Using algorithm 1, for unseen state-object compositions (from MIT-States dataset) *cooked chicken*, *dull granite* and *blunt sword*, the feasibility values obtained from feasibility matrix $\mathcal{M}_F$ are $0.64, 0.65$ and 0.75, respectively. For infeasible compositions like *rusty ocean*, the predicted feasibility value from $\mathcal{M}_F$ is 0.05, which is quite low. Evidently, $\mathcal{M}_F$ predicts the feasibility of unseen state-object compositions effectively.

Next we also evaluate the utility of the $max(\cdot)$ operation as used in (3.11) and compare it with the effectiveness of $min(\cdot)$ and $avg(\cdot)$ operations. For an array of real numbers arr, $max(arr)$ and $min(arr)$ operations represent the value of the maximum and minimum elements of the array arr, respectively. Similarly, $avg(arr)$ operation represents the average value of all the elements of the array arr. On MIT-States dataset, for $max(\cdot)$, $min(\cdot)$ and $avg(\cdot)$ operations on (3.11), the model achieves AUC of 8.4, 8.1 and 8.2, respectively (on the validation set). This can be justified from the following facts. In MIT-States dataset, there are 1262 compositional classes in the train set (i.e. *seen* classes) and 300 and 400 *unseen* classes in the validation and test set, respectively (see Table 2.1). In view of the high

Model	AUC on UT-Zappos50k	AUC on MIT-States-CL	AUC on MIT-States	AUC on C-GQA
with 1 graph layer	33.5	9.4	7.3	3.9
with 2 graph layers	**45.9**	**11.5**	**8.4**	**5.6**
with 3 graph layers	41.5	8.3	5.8	4.7
with 4 graph layers	38.1	6.8	5.1	3.6
with 5 graph layers	36.8	6.3	4.5	3.5

Table 3.7: Ablation study on effects of depth of graph in GCN of proposed model

amount of *unseen* compositional classes ($1262 : 700 \approx 1.8 : 1$ ratio between the number of *seen* and *unseen* classes), there are insufficient number of compositional classes in the train set to decide the feasibility of unseen compositions using feasibility assessment strategy (see Section 3.3.5). In this context the $max(\cdot)$ operation helps to extract the highest feasibility score corresponding to the state and the object.

Additionally, we analyse the effect of inclusion of the feasibility assessment strategy during training of the state-of-the-art algorithms in OW-CZSL and CW-CZSL. In CW-CZSL the possible state-object prediction consists of only the feasible state-object compositions. Also LE, CGE, TMN, AAO are designed to predict state and object jointly. So there is no possibility of prediction of infeasible composition by LE, CGE, TMN, AAO in CW-CZSL. Thus there is no scope of using the feasibility prediction strategy for the above mentioned algorithms in CW-CZSL. However, SymNet has the provision of independently predicting the state and object. There may be prediction of infeasible state-object composition in SymNet. Next we train SymNet with the inclusion of $\mathcal{L}_{FL}$. We observe an improvement of 0.9% on *test AUC* for SymNet on MIT-States-CL dataset after inclusion of $\mathcal{L}_{FL}$. Next on OW-CZSL, we include $\mathcal{L}_{FL}$ during training of the compared algorithms, i.e. LE, AAO, SymNet, TMN and CGE on MIT-States-CL dataset. The improvement in *test AUC* observed is 0.8%, 1.1%, 1.3%, 1.3% and 1.0% for LE, AAO, SymNet, TMN and CGE, respectively. Evidently, for all the algorithms, better *test AUC* is achieved due to the presence of $\mathcal{L}_{FL}$.

The result of Table 3.7 reports the effects of different number of graph layers in the model's performance. We have experimented with $1, 2, 3, 4$ and 5 layers of GCN to construct both OFLN and SFLN. In order to use knowledge sharing between SFLN and OFLN (Section 3.3.4), we have always used same number of graph layers in both SFLN and OFLN. In Table 3.7, increment of the graph layers shows a negative effect on the algorithm's performance. This can be justified using over-fitting problem that occurs with the increase in number of layers in GCN (Li et al., 2018, Cai and Wang, 2020, Oono and Suzuki, 2020). Due to increase in the number of layers in GCN, the number of neighbours of each node increases. As a result, the nodes can have a larger receptive field. The resulting deeper GCN model, after training for some epochs, treats all the nodes identically and the feature vectors of the nodes converge to indistinguishable vectors (Oono and Suzuki, 2020, Li et al., 2018, Cai and Wang, 2020)). This reduces diversity of the model. The optimum result is obtained with two layers of GCN in both OFLN and SFLN.

The encoder in CLKSN is created using one fully connected layer with $N_o \times N_o$ number of input nodes and N_{hid} number of output nodes. The decoder network is created using a similar fully connected layer with N_{hid} number of input nodes and $N_s \times N_s$ number of output nodes. For MIT-States and UT-Zappos50k values of $N_{hid} = 250$ and 4 have been used in our algorithm (see last row of Fig. 3.8). The encoder in the CLKSN encodes the adjacency information in the intermediate layer of the OFLN into the latent features. The latent

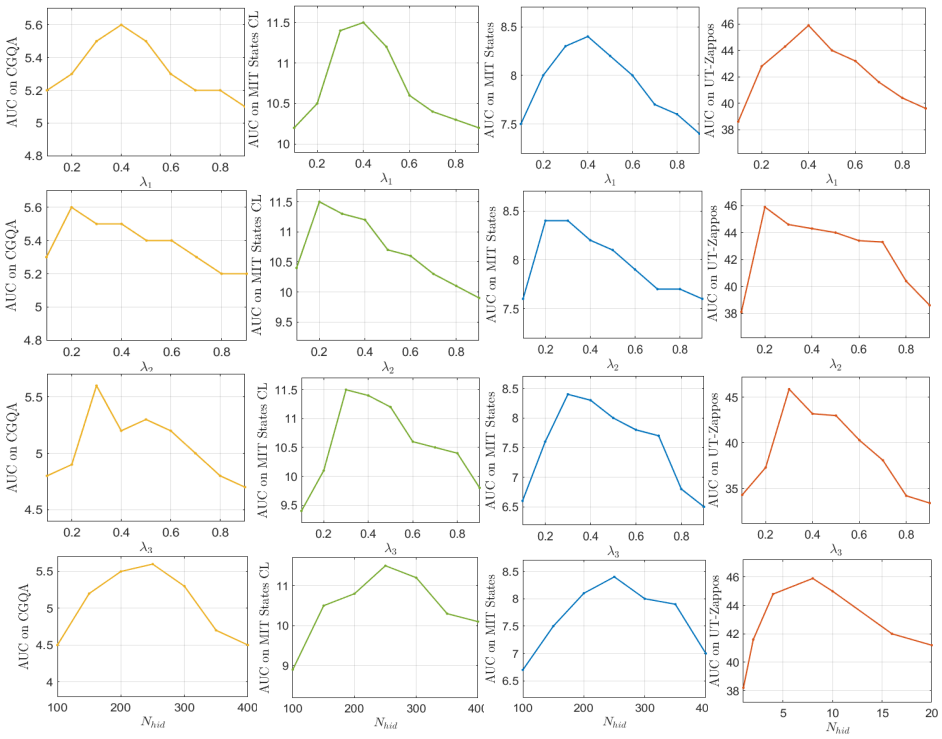


Figure 3.8: The first three rows show the variation of AUC due to change in the parameters λ_1, λ_2 and λ_3 of (3.17). Also we show the variation of AUC due to variation in the parameter N_{hid} in the CLKSN. The first, second, third and fourth columns represent the variations of AUC on C-GQA, MIT-States-CL, MIT-States and UT-Zappos50k datasets, respectively

features are next passed to the decoder. Next the resulting decoded output is added with the adjacency matrix in the intermediate layer of the SFLN. The adjacency matrix consists of information like the groups of objects that form feasible composition with same group of states. The CLKSN attempts to remove any redundant information from the adjacency matrix of the OFLN and share the useful information to the SFLN. In MIT-States dataset, there are 115 states and 245 objects in the dataset. UT-Zappos50k, on the contrary has only 12 states and 16 objects. Due to small number of states in UT-Zappos50k, the encoder of CLKSN can encode the useful information of the adjacency matrix in OFLN in a much smaller dimension. Thus we have used $N_{hid} = 4$ for UT-Zappos50k and $N_{hid} = 250$ for MIT-states dataset.

We perform ablation study to select the values for the parameters λ_1, λ_2 and λ_3 as used in (3.17) and the parameter N_{hid} in the encoder-decoder architecture of CLKSN. The result of these ablation studies are shown in Fig. 3.8. The nature of the graphs in Fig. 3.8, indicates desirable range of parameter values. We have selected the values of the parameters $\lambda_1, \lambda_2, \lambda_3$ and N_{hid} corresponding to the highest AUC on the four data sets. Next we summarise the contributions of the chapter.

A general observation can be made regarding the graphs in Fig. 3.8 is that, the performance of our algorithm depends heavily on the optimal value of the parameters. Thus it is crucial to find out the optimal value of the hyper-parameters for better performance of the SeCoNet.

As shown in Table 3.3, the closest competitor to our approach are the SymNet (Li et al., 2020) and the CGE Naeem et al. (2021) algorithms. We measure the number of floating point operations (FLOPs) required for a single forward pass through the model, to process a batch of 2 images. Using this experimental setting, the number of FLOPs required for CGE is 7.1×10^{10}, on the MIT-States dataset. The FLOPs required for SymNet is 9.2×10^{8}. The approach proposed by us requires 9.9×10^{9} FLOPs. Among the two compared algorithms, CGE reports results closest to our approach. CGE follows a similar backbone like our approach, using Graph convolutional Network. However, CGE requires more than 7 times the FLOPs as required by our approach. This may be justified as follows. The architecture in CGE consists of a GCN modules with a node for each possible state-object composition in the dataset. On the contrary, our approach does not require node for each state-object composition. Instead, a node for each state in the dataset and one node for each object in the dataset is required. The number of states and number of objects in CZSL datasets, when combined are much lower than number of possible state-object compositions in the dataset. As shown in Table 2.1, the number of states, objects and state-object compositions in MIT-States dataset are 115, 245 and 1962, respectively. Thus proposed approach is computationally efficient than the existing approaches. Besides our approach also reported better results than CGE. SymNet, although requires less number of FLOPs than our approach, our approach outperforms SymNet across all the metrics as reported in Table 3.3.

The feasibility loss $\mathcal{L}_{FL}$ penalizes the model if, corresponding to the input image, an infeasible state-object composition is predicted. Thus, the use of $\mathcal{L}_{FL}$ discourages largely incorrect predictions by the proposed model. In other words, due to the use of $\mathcal{L}_{FL}$, the gradients from back-propagation are less likely to experience sharp changes and model parameters are less prone to experience large updates. So, the use of $\mathcal{L}_{FL}$ promotes stable and controlled parameter updates during training. In addition, $\mathcal{L}_{FL}$ is useful for less complex models by restricting the prediction space of the CZSL model to only feasible predictions. This reduces the chances of overfitting the model.

3.5 Summary

In this chapter, a multi-branch graph convolutional network based architecture to solve the problem of CZSL is proposed. An important contribution is the introduction of the cross-layer knowledge sharing between object and state feature networks. In an image of a particular state-object composition, the state can affect the object's visual features in different ways. There exists different levels of association between a state and an object in a state-object composition. This variation in association gives rise to high intra-class variance in the visual features of different images belonging to a state-object compositional class. We attempt to address this variation in association between visual features of state and object in the next chapter.

Chapter 4

Knowledge Guided Transformer Network

4.1 Introduction

As discussed in Section 3.5, a major challenge in CZSL is understanding the extent of association between the basic visual primitives (state and object). For example in different images of *peeled orange*, an *orange* may be *peeled* to different extents. Based on the extent to which the *orange* is *peeled*, the visual features of images of *peeled orange* will vary. Hence, there exists significant amount of intra-class variability among the visual features of different images of same composition.[1]

Existing approaches (Purushwalkam et al., 2019, Nagarajan and Grauman, 2018, Naeem et al., 2021, Mancini et al., 2021) ignore the possibility of partial association between state and object features in a compositional image and only look for existence or absence of the features of a state or object in a composition. Hence, these approaches fail to tackle the significant amount of intra-class variability among the visual features of multiple images of a composition. Ideally, a CZSL approach should look for the existence of the state features (and object features) and also the extent of association between state and object features. In this work we have proposed a *Knowledge Guided Transformer* network (referred as KGT-Net) as shown in Fig. 4.1, in an attempt to solve the above mentioned two problems of CZSL.

As discussed in Section 1.2, the visual features of the state vary widely based on the particular state-object composition. For example the state *ripe* has distinct visual properties in the compositions *ripe apple* and *ripe banana*. This creates ambiguity for the model to learn unique features for the state *ripe*. Thus the visual features of the state in state-object composition depends largely on the context. Hence, effective processing of the contextual relationship is a necessary requirement to solve the CZSL problem.

Earlier proposed CZSL algorithms (Misra et al., 2017, Nagarajan and Grauman, 2018, Purushwalkam et al., 2019, Li et al., 2020, Naeem et al., 2021) mainly rely on Convolutional Neural Networks (CNNs) (He et al., 2016) to extract the image features. CNNs, due to convolution and pooling, have a small receptive field. Hence, CNNs can only extract locally dominant visual features (Yuan et al., 2021b). CNNs are inefficient to process the contextual relationship over larger region in an image, as needed to solve the CZSL problem (Dosovitskiy

[1] A part of the work done in this chapter is published as follows,
Aditya Panda and Dipti Prasad Mukherjee, "Knowledge Guided Transformer Using Partial Association of Features of State and Object for Compositional Zero-shot Learning", accepted in ACM Transactions on Multimedia Computing Communications and Applications, 2024.

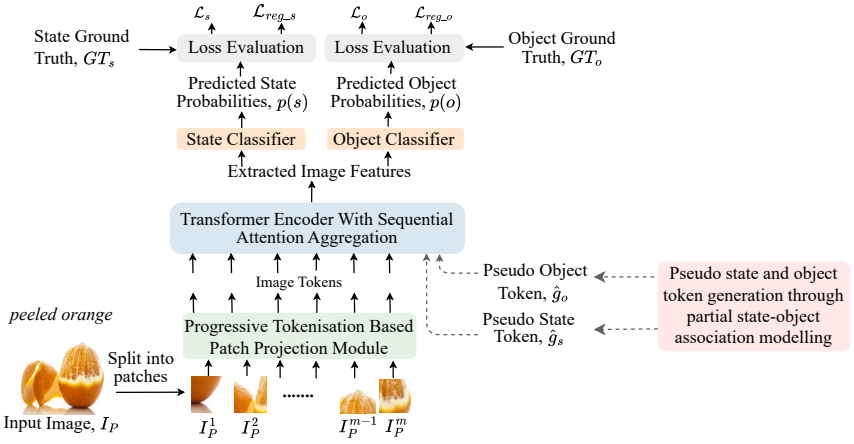


Figure 4.1: Block diagram of the proposed approach. Here $\mathcal{L}_s$, $\mathcal{L}_o$, $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$ represent different loss functions used to train the model.

et al., 2020). Vision Transformer (ViT) (Dosovitskiy et al., 2020, Yuan et al., 2021b) based approaches split an image into a set of patches and then process the interaction between all possible pairs of patches. This patch based processing approach allows the ViT to have a larger field of view over the input image. Hence, ViT can process context dependencies over a larger area in images. As explained earlier, the major challenge of CZSL is the ambiguity of state features due to its context dependency. Inspired by the effectiveness of ViTs, in processing context dependency, we propose a novel Knowledge Guided Transformer. We integrate the intermediate output from the KGT-Net with prior knowledge about the states and objects. This prior knowledge integration helps the proposed KGT-Net to better tackle the intra-class variability challenge of CZSL.

While integrating the prior knowledge about states and objects, we have attempted to model the partial association between state and object features. The prior knowledge about the states and objects is generated from external knowledge (word features (Mikolov et al., 2013, Pennington et al., 2014) of state and object labels). The obtained prior knowledge is integrated with the KGT-Net in the form of two feature vectors referred to as pseudo state token and pseudo object token.

The proposed KGT-Net, consists of two modules, State Weight Proposal Gate (SWPG) and Object Weight Proposal Gate (OWPG). SWPG and OWPG generate a set of scalar weights (referred as gating weights). These gating weights represent the extent of association between state features and object features. Simultaneously we use deep neural network to learn the state knowledge vectors from the word embeddings of the state labels (Mikolov et al., 2013, Pennington et al., 2014). Next the knowledge about each state in the dataset are multiplied by the corresponding state weights, σ_s^i, $\sigma_s^i \in [0, 1]$, $i \in \{1, 2, \ldots, N_s\}$. The weighted state knowledge vectors are combined to obtain the final pseudo state token. Similar strategy is followed for generating the pseudo object token. This state and object tokens are finally integrated within the KGT-Net to help the model in final state-object recognition.

Since there is no ground truth to supervise the partial association of features of state and object, we utilise a novel regularisation approach to learn the σ_s^i values. To the best of our knowledge, our work is the first in the domain of CZSL to address the problem of partial

association between features of state and object.

Additionally, we propose a novel Layer-Wise Adaptive Attention Aggregation module. Proposed approach generates layer specific adaptive weights and multiplies with the output of the intermediate layers of the *Transformer Encoders*. The intermediate layer outputs of the *Transformer Encoders*, after being multiplied with the adaptive weights, are aggregated and added with the final layer features from the *transformer encoder*. In CZSL, recognising a state is more difficult than recognising an object due to the variability observed in the state features. It can be observed that the fine-grained feature of the image (like textures) are specifically useful for state features recognition. *transformer encoder* processes image features sequentially through a number of layers. This sequential processing of visual features often reduces the feature diversity, specifically the fine-grained visual cues (Yuan et al., 2021b). Hence, to preserve feature diversity through layers, we propose Layer-Wise Adaptive Attention Aggregation module. The proposed multi-layer feature aggregation approach specifically helps to better preserve the fine-grained features recognised in different layers of the *transformer encoder* and subsequently helps to recognise the state in the state-object compositions better. Besides, we have utilised the progressive tokenisation framework for the image patches as proposed in (Yuan et al., 2021b). Next we briefly summarise the contributions of the proposed work.

1. The context dependency of state features is efficiently addressed using a novel *Knowledge Guided Transformer*.

2. For the first time, a knowledge integration strategy is introduced to better incorporate partial association between state features and object features.

3. We also propose a novel layer-wise adaptive attention aggregation approach to better process the feature diversity in a compositional image.

4. Our approach outperforms the existing state-of-the-art approaches for CZSL problem in both open-world and closed-world CZSL evaluation protocols.

Remaining part of this chapter is organised into four sections. First we briefly summarise the recent works relevant to our approach in Section 4.2. Next we discuss the proposed approach in Section 4.3. We present the results and experimental setup in Section 4.4 followed by conclusions in Section 4.5.

4.2 Related Works

Compositional Zero-Shot Learning from perspective of partial association between state and object features: The literature review of relevant CZSL approaches are reported in Sections 2.2 and 3.2, respectively. Here we discuss the capability of relevant approaches to model the partial association between state and object features.

Existing CZSL approaches (Nan et al., 2019, Yang et al., 2020, Atzmon et al., 2020a, Mancini et al., 2021, Ruis et al., 2021, Yang et al., 2022, Saini et al., 2022, Wang et al., 2023b, Li et al., 2023b) find unique features for each state and each object and correspondingly unique features are obtained for the state-object composition. Thus these approaches fail to

incorporate the variability in different images of same state-object composition due to partial association between state and object features.

In the approach by Purushwalkam et al. (2019) the text features of the state and object labels are multiplied with intermediate image features from a multi-layered image feature extractor, leading to partial integration of the image features with state and object features. However, in (Purushwalkam et al., 2019) the weights generated from word labels of each state and object are always same irrespective of the input image. On the contrary, in our approach, the gating weights represent the extent of association between state features and object features. The gating weights are functionally dependent on the features of the input image which helps to better process the intra-class variability caused by partial association between state and object features.

Naeem et al. (2021) have considered the Graph Convolutional Network (GCN) (Kipf and Welling, 2016) architecture with nodes for each state, each object and each feasible state-object composition in the dataset. However, the adjacency between state and object nodes are fixed binary values i.e. $\in \{0, 1\}$ with only provision for either the inclusion or exclusion of features of a particular state and object in the image features, with no scope for partial association between state and object feature. Li et al. (2022) have attempted to better recognise the unseen compositions by using a generative network to create visual features of unseen compositions. However, the generative network in (Li et al., 2022) takes state and object embeddings as input without any control on the extent of association between the features of state and object.

Vision Transformer: Transformer was originally introduced in an attempt to solve machine translation (Vaswani et al., 2017). More recently, Dosovitskiy et al. (2020) proposed Vision Transformer (ViT) which modifies the original transformer to use it in vision based problems. In recent years, transformer has shown great success in diverse high-level vision tasks, such as visual question answering (Man et al., 2022), image generation (Zhang et al., 2022a), video captioning (Man et al., 2022) etc. Our work utilises the progressive tokenisation strategy of the T2T-ViT (Yuan et al., 2021b). The tokenisation strategy in (Yuan et al., 2021b) helps in making the tokens (embeddings of patches generated from input image) more informative by incorporating the neighbourhood information in each token. However, we have additionally generated a pseudo state and a pseudo object token and their novel regularisation approach to better solve the CZSL problem. Besides we have proposed a layer wise attention aggregation module for better preserving of feature diversity as required in CZSL. Next we present the proposed *Knowledge Guided Transformer*.

4.3 Knowledge Guided Transformer Network

The proposed approach is shown in Fig. 4.2. The approach consists of the following modules, State Weight Proposal Gate (SWPG), Object Weight Proposal Gate (OWPG), Progressive Tokenisation Based Patch Projection Module, *transformer encoder* and Layer-wise Adaptive Attention Aggregation Module. The detailed working principle of these modules are reported in the subsequent sections.

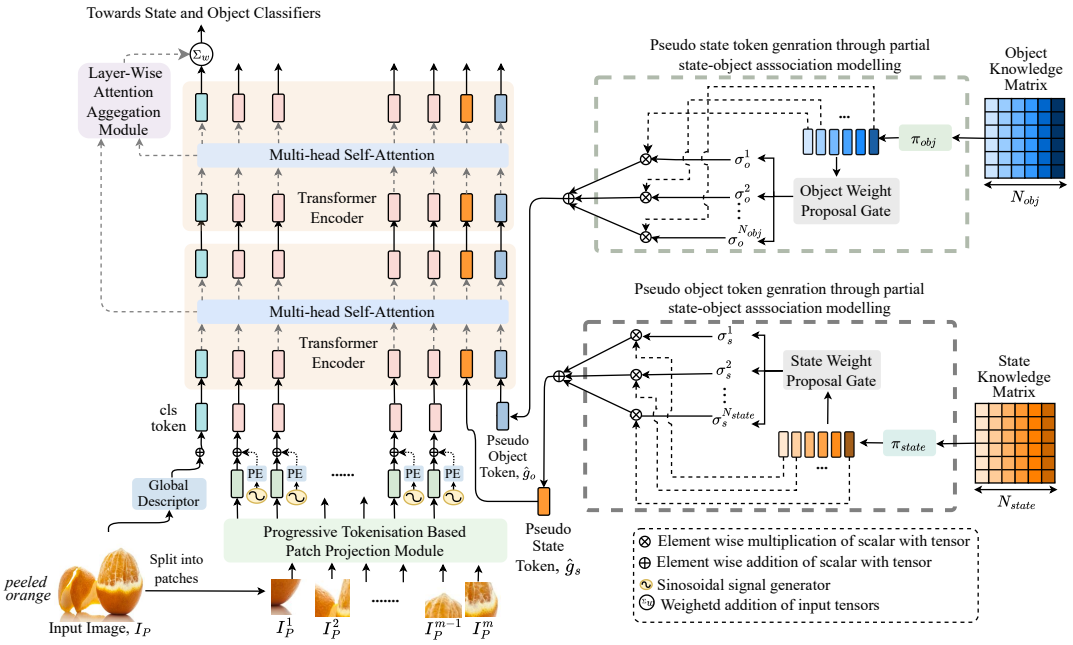


Figure 4.2: *Knowledge Guided Transformer Network* (KGT-Net): PE represents the Sinusoidal Positional Embedding (Dosovitskiy et al., 2020).

4.3.1 Motivation

We argue that the approach in the existing ViT (Dosovitskiy et al., 2020) is not sufficient to solve the CZSL problem. In CZSL, an object's visual features happen to be entangled with the state's visual features. This intricate dependency between features of state and object makes CZSL problem inherently difficult to solve.

To adapt the existing ViT (Dosovitskiy et al., 2020) for CZSL problem, we create two additional tokens alongside the existing tokens generated from image patches. ViT (Dosovitskiy et al., 2020) splits images into patches and a token (or patch embedding) is obtained from each patch. An input image $I \in \mathbb{R}^{3\times H\times W}$ is sequentially split into m number of $P \times P$ dimensional patches. The resulting patches are represented as $\{I_P^j\}_{j=1}^m$, $I_P^j \in \mathbb{R}^{3\times P\times P}$, where $m = \lfloor \frac{H}{P} \rfloor \times \lfloor \frac{W}{P} \rfloor$. Each patch I_P^j is passed through the *Projection Module*, constructed using a set of neural networks. For each patch, I_P^j, the output of the projection module is d dimensional vector, also referred as token, $X^j \in \mathbb{R}^d$. By concatenating all the d dimensional token vectors row-wise, the token matrix, $X_0 \in \mathbb{R}^{m\times d}$ is obtained. The token matrix is next passed to the *transformer encoder* (Dosovitskiy et al., 2020) which subsequently processes the tokens.

The two additional tokens are constructed using the prior knowledge about the state and the object present in the input image. Subsequently, these derived state and object tokens are concatenated along with existing tokens generated from the image. This prior knowledge in the form of two additional tokens helps the proposed KGT-Net to better understand the entanglement between visual features of state and object. Next we discuss the challenges involved in generating the state and the object tokens.

4.3.2 Challenges

In context of the current problem, each state-object compositional image contains at most one object and one state. Hence, integrating external knowledge about all possible states and objects present in the dataset will require generation of many state and object tokens.

It can be shown that the computational cost for *transformer encoder* (Dosovitskiy et al., 2020) to process the context dependency between all possible tokens is a quadratic function of the number of tokens (Mao et al., 2022). Naturally, adding knowledge about all possible states and objects to the *Transformers Encoder* is computationally expensive. However, to seek knowledge about the particular state or object which is present in an image, will require accessing the ground truth.

Hence, we propose a novel framework to integrate knowledge about state and object without using the ground truth. During training, the knowledge about the state and object are included by generating a pair of pseudo state and object tokens. Also, state and object ground truths are used to supervise the pseudo token generation process during training. During test since ground truths are not available, only the pseudo state and the pseudo object tokens are used.

4.3.3 Pseudo State and Pseudo Object Tokens

Let us assume that input image I has state and object ground truths GT_s and GT_o, respectively. Let $g_s \in \mathbb{R}^d$ and $g_o \in \mathbb{R}^d$ represent the corresponding state token and the object token generated from GT_s and GT_o, respectively. The state and object tokens, g_s and g_o are generated utilising the word embedding (Mikolov et al., 2013) of the state and object labels, respectively.

However, GT_s and GT_o are inaccessible during test time. Hence, we generate pseudo state token and pseudo object token, $\hat{g}_s \in \mathbb{R}^d$ and $\hat{g}_o \in \mathbb{R}^d$, respectively. The generated pseudo tokens are regularised to be similar to the tokens generated from the ground truth. In the process of generating the pseudo state and object tokens, the partial association between state and object features are also processed. Next we explain the generation process of pseudo tokens to incorporate partial association between state and object features.

4.3.4 Generation of Pseudo Object and Pseudo State Tokens

As in Section 2.3.1, assume there are N_s and N_o number of states and objects in the dataset, respectively. We create the State Knowledge Matrix by stacking the word embeddings (Mikolov et al., 2013) of all possible state labels, represented as $KM_s \in \mathbb{R}^{N_s \times d_{embed}}$. Here d_{embed} represents dimension of embedding vector of each state label as obtained from the Word2Vec (Mikolov et al., 2013). Using a similar approach we create the Object Knowledge Matrix, $KM_o \in \mathbb{R}^{N_o \times d_{embed}}$.

The word embeddings and the tokens from image patches are obtained from two distinct distributions. Hence, to make the word and the image features comparable, we use latent projection functions $\pi_s(\cdot)$ and $\pi_o(\cdot)$, respectively. We obtain the Transformed State and Object Knowledge Matrices as $E_o = \pi_o(KM_o)$ and $E_s = \pi_s(KM_s)$ where $E_o \in \mathbb{R}^{N_o \times d}$, $E_s \in \mathbb{R}^{N_s \times d}$. The functions $\pi_s(\cdot)$ and $\pi_o(\cdot)$ are approximated using an MLP based architecture as detailed in the subsection on implementation details under Section 4.4.

Once we obtain the Transformed State and Object Knowledge Matrices, proposed approach estimates a set of scalar weights, referred as gating weights. The gating weights represent partial association between state features and the object features in a composition. Next we explain the motivation to generate the gating weights and also the relevance of the strategy for modelling of partial association between state and object features.

Figure 4.3: Four images of a *peeled orange* are shown. Each *orange* is *peeled* to a different extent. Thus the visual features of the state-object composition *peeled orange* vary.

In CZSL problem there may be high amount of variability among the images from same state-object compositional class (see different extent of association between the features of state *peeled* and the corresponding features of object *orange* in Fig. 4.3). We argue that this association between state and object features is not modelled properly in the existing literature (Misra et al., 2017, Nagarajan and Grauman, 2018, Purushwalkam et al., 2019, Li et al., 2020, Nan et al., 2019, Xu et al., 2021b). More specifically, the quantitative measure of extent of presence of a state feature in an image composition cannot be represented through discrete binary values 0 or 1. In other words, mere existence or non-existence of a particular state or object features in a state-object composition is not sufficient for understanding the variability of state-object compositional image features. We propose that the quantification of extent of presence of state and object features can be any value within the range $[0, 1]$. Using the proposed gating strategy, we attempt to model the scope of partial association between state and object features in a state-object composition to better understand the intra-class variability.

We define two gating networks, the State Weight Proposal Gate (SWPG) and Object Weight Proposal Gate (OWPG) (see Fig. 4.2). The input to SWPG are E_s and the token matrix X_0. The input to OWPG are the E_o and X_0. Next, SWPG and OWPG generate a set of state and object gating scores, σ_s and σ_o, respectively with $0 \leq \sigma_s^i, \sigma_o^j \leq 1$ and $i \in \{1, 2, \ldots, N_s\}$, $j \in \{1, 2, \ldots, N_o\}$. Finally we obtain the pseudo state token $\hat{g}_s$ and object token $\hat{g}_o$ as follows,

$$\hat{g}_s = \sum_{i=1}^{N_s} \sigma_s^i \otimes E_s[i] \text{ and } \hat{g}_o = \sum_{j=1}^{N_o} \sigma_o^j \otimes E_o[j]. \tag{4.1}$$

Here $\otimes$ represents multiplication of a scalar with a vector where all the elements of the vector are multiplied by the scalar. Here $\hat{g}_s, \hat{g}_o$ are the pseudo state and pseudo object ground truth tokens, respectively. The contribution of features of each state in the pseudo state token are adaptively controlled through the gating weights, σ_s^i. Thus σ_s^i values quantify of the extent of presence of features of i^{th} state in the compositional image features. The learning process for state and the object pseudo tokens, $\hat{g}_s$ and $\hat{g}_o$ are discussed next.

4.3.5 Regularisation of Pseudo State and Object Tokens

We assume that g_s and $\hat{g}_s$ follow marginal probability distributions $p(g_s)$ and $p(\hat{g}_s)$, respectively. We express the mutual information between g_s and $\hat{g}_s$ as follows,

$$MI(\hat{g}_s; g_s) = KL\Big(p(\hat{g}_s, g_s)||p(\hat{g}_s)p(g_s)\Big) = \sum_{g_s,\hat{g}_s} p(\hat{g}_s, g_s) \log \frac{p(\hat{g}_s, g_s)}{p(\hat{g}_s)p(g_s)}. \tag{4.2}$$

Here $p(\hat{g}_s, g_s)$ represents the corresponding joint probability distribution. Our goal is to maximize the mutual information $MI(\hat{g}_s; g_s)$ such that the pseudo state tokens and the state tokens generated from the ground truth are as similar as possible. Evidently the functional representations of the marginal distributions in (4.2) is unknown. Hence, as in (Tian et al., 2019), we define a distribution $p_\tau(\cdot|\tau)$ conditioned on a latent scalar variable $\tau \in \{0, 1\}$. $\tau = 1$ indicates that we sample a tuple $(\hat{g}_s, g_s)$ from the joint distribution $p(\hat{g}_s, g_s)$. Similarly, $\tau = 0$ represents that we sample $\hat{g}_s$ and g_s from the respective marginal probability distributions $p(\hat{g}_s)$ and $p(g_s)$. Thus we have,

$$p_\tau(\hat{g}_s, g_s|\tau = 1) = p(\hat{g}_s, g_s), \tag{4.3}$$

$$p_\tau(\hat{g}_s, g_s|\tau = 0) = p(\hat{g}_s)p(g_s). \tag{4.4}$$

Without loss of generality, we assume the prior distribution $p_\tau(\cdot)$ to be equally likely for $\tau = 1$ and $\tau = 0$, i.e. $p(\tau = 0) = p(\tau = 1) = 0.5$. So, by Bayes' rule we can obtain the posterior for $\tau = 1$ as

$$\begin{aligned} p_\tau(\tau = 1|\hat{g}_s, g_s) &= \frac{p_\tau(\hat{g}_s, g_s|\tau = 1)p(\tau = 1)}{p_\tau(\hat{g}_s, g_s|\tau = 1)p(\tau = 1) + p_\tau(\hat{g}_s, g_s|\tau = 0)p(\tau = 0)} \\ &= \frac{p(\hat{g}_s, g_s)}{p(\hat{g}_s, g_s) + p(\hat{g}_s)p(g_s)} [\text{since } p(\tau = 0) = p(\tau = 1) = 0.5] \\ &\leq \frac{p(\hat{g}_s, g_s)}{p(\hat{g}_s)p(g_s)} [\text{since } p(\hat{g}_s, g_s) \geq 0]. \end{aligned} \tag{4.5}$$

Next, we get a lower bound of $MI(\hat{g}_s; g_s)$, as follows,

$$\begin{aligned} MI(\hat{g}_s; g_s) &= \sum_{g_s,\hat{g}_s} p(\hat{g}_s, g_s) \log \frac{p(\hat{g}_s, g_s)}{p(\hat{g}_s)p(g_s)}, \\ &\geq \sum_{g_s,\hat{g}_s} p(\hat{g}_s, g_s) \log p_\tau(\tau = 1|\hat{g}_s, g_s) [\text{using (4.5)}], \\ &\geq \sum_{g_s,\hat{g}_s} p_\tau(\hat{g}_s, g_s|\tau = 1) \log p_\tau(\tau = 1|\hat{g}_s, g_s) [\text{using (4.3)}], \\ &\geq \mathbb{E}_{p_\tau(\hat{g}_s,g_s|\tau=1)} \log p_\tau(\tau = 1|\hat{g}_s, g_s). \end{aligned} \tag{4.6}$$

To maximize the mutual information, $MI(\hat{g}_s; g_s)$, the lower bound of $MI(\hat{g}_s; g_s)$, as found out in (4.6), i.e. $\mathbb{E}_{p_\tau(\hat{g}_s,g_s|\tau=1)} \log p_\tau(\tau = 1|\hat{g}_s, g_s)$ is maximized. Evidently, the functional distribution of $p_\tau(\tau = 1|\hat{g}_s, g_s)$ is not known. Let $\psi_g : \hat{g}_s, g_s \rightarrow [0, 1]$ be a function which approximates $p_\tau(\tau = 1|\hat{g}_s, g_s)$. Since neural networks can be universal function approximator (Funahashi, 1989, Hornik et al., 1989), we implement $\psi_g(\cdot)$ using an MLP. The inputs

of the corresponding MLP are the ground truth and the pseudo ground truth, g_s and $\hat{g}_s$, respectively. The output from the MLP are the output probabilities, $p_\tau(\tau = 1|\hat{g}_s, g_s)$ and $p_\tau(\tau = 0|\hat{g}_s, g_s)$. So we re-write (4.6) as follows,

$$MI(\hat{g}_s; g_s) \geq \mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=1)} \log \psi_g(\hat{g}_s, g_s). \tag{4.7}$$

Next we represent the RHS of (4.7) as the state regularisation loss $\mathcal{L}_{reg\_s}$, i.e.,

$$\mathcal{L}_{reg\_s} = \mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=1)} \log \psi_g(\hat{g}_s, g_s). \tag{4.8}$$

However, $\mathcal{L}_{reg\_s}$ in the current form in (4.8) is not sufficient to train the above-discussed MLP, $\psi_g(\cdot)$. As already explained, the latent scalar vector τ can have values 0 and 1. However, the current $\mathcal{L}_{reg\_s}$ only regularises the positive value of τ, i.e. $\tau = 1$, without penalising the predictions for the cases where $\tau = 0$. Thus following a similar approach of using a cross-entropy loss for binary classification, we re-define $\mathcal{L}_{reg\_s}$ as follows,

$$\mathcal{L}_{reg\_s} = \mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=1)} \log \psi_g(\hat{g}_s, g_s) + \mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=0)} \log(1 - \psi_g(\hat{g}_s, g_s)). \tag{4.9}$$

The above explained process is intended for regularisation of the pseudo state tokens. Similar process is followed for generation of pseudo object token. Next the regularised pseudo state token and the pseudo object tokens, $\hat{g}_s$ and $\hat{g}_o$ are concatenated with the obtained image token matrix as follows,

$$X_0 \leftarrow Concat(X_0, \hat{g}_s, \hat{g}_o). \tag{4.10}$$

Next we explain the progressive tokenisation based projection module used in our approach.

4.3.6 Projection Module: Tokens-to-Token

The vanilla ViT (Dosovitskiy et al., 2020) uses an MLP based architecture in the *projection module* to obtain the patch embeddings or tokens from the image patches. However, this MLP based patch projection approach is shown to be inefficient to process the detailed visual features (Yuan et al., 2021a, Graham et al., 2021, Han et al., 2021, Yuan et al., 2021b). A number of new approaches have been proposed in recent years for patch projection (Yuan et al., 2021a, Graham et al., 2021, Han et al., 2021). We have followed the tokenisation based patch projection module as reported in (Yuan et al., 2021b) for our approach. Next, we briefly explain the steps involved in the above mentioned tokenisation approach as used in our context.

Initially patches are extracted from the input image $I \in \mathbb{R}^{h \times w \times c}$. While extracting patches from I, instead of conventional approach, overlap is allowed between patches. This overlapping strategy helps to make individual patches more aware about surrounding patches. In other words, information about neighbourhood structures is better incorporated. The target patch size is $k \times k$ with s pixel overlap allowed between neighbourhood patches. Also a padding of p pixels is allowed between neighbourhood patches. Here $(k - s)$ is similar to the stride in usual convolution operation. The number of generated patches from I is represented as n_p and is obtained as

$$n_p = \left\lfloor \frac{h + 2p - k}{k - s} + 1 \right\rfloor \times \left\lfloor \frac{w + 2p - k}{k - s} + 1 \right\rfloor. \tag{4.11}$$

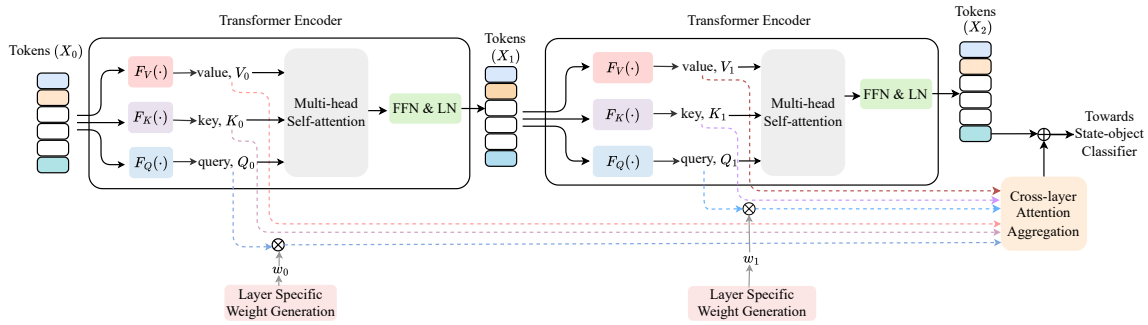


Figure 4.4: Block level representation of the proposed Layer-Wise Adaptive Attention Aggregation Module: Only two layers of *transformer encoder* (Vaswani et al., 2017) are shown for simplicity. FFN and LN modules in the diagram represent Feed-Forward Network and Layer Normalization layers, respectively (Vaswani et al., 2017).

Each of the patches have dimensions $k \times k \times c$ and n_p such patches are obtained. We flatten all patches along spatial dimensions to tokens $X_0 \in \mathbb{R}^{n_p \times (c \times k^2)}$. The obtained patches are again passed to *transformer encoder* (see Fig. 4.2). Next we explain the Layer Wise Attention Aggregation module of the proposed approach.

4.3.7 Layer-Wise Adaptive Attention Aggregation

In CZSL problem, the visual features of states in images are often observed to vary significantly, in comparison to features of objects. Due to this intra-class variation in visual features of the state, uniquely recognising the state features is challenging. We observe that in state-object composition, the state is a semantic description of the object. Also, the fine-grained visual features, like texture are useful for recognition of the state. Conventional *Transformer Encoders* are designed to process the image features sequentially through the layers of the *transformer encoder*. This sequential processing of visual features often fails to preserve the fine-grained and coarse-grained features and subsequently reducing the feature diversity (Yuan et al., 2021b). Hence, to preserve feature diversity through layers, we propose Layer-Wise Adaptive Attention Aggregation (LWAAg) module as described next. A block level representation of the LWAAg module is shown in Fig. 4.4. Our LWAAg module builds on top of the *transformer encoder* (Dosovitskiy et al., 2020). So, before reporting the proposed LWAAg module, we first briefly review the working principle of the *transformer encoder*.

4.3.7.1 Brief Review of Transformer Encoder

The *transformer encoder* consists of L_E number of layers. Each of the layer consists of a Multi-head Self Attention (MhSA) module, Layer Normalization (LN) module and Feed Forward Network (FFN) with skip connections (Dosovitskiy et al., 2020). First, we present the working principle of the MhSA module. For MhSA in the l^{th} layer of *transformer encoder*, we obtain three matrices viz. query Q_l, key K_l and value V_l, $l \in \{1, 2, \ldots, L_E\}$. These matrices are obtained by using projection functions, $F_Q(\cdot)$, $F_K(\cdot)$ and $F_V(\cdot)$, respectively (see Fig. 4.4).

If the token matrix at the input of the l^{th} layer is represented as $X_l \in \mathbb{R}^{m \times d}$, then we have,

$$Q_l = F_Q(X_l), K_l = F_K(X_l), V_l = F_V(X_l). \tag{4.12}$$

Next Q_l, K_l and V_l are utilised to obtain the final output feature matrix for MhSA in layer l as follows,

$$Attn_l(X_l) = \sigma\Big(\frac{Q_l(K_l)^T}{\sqrt{d}}\Big)V^l = \sigma\Big(\frac{F_Q(X_l)(F_K(X_l))^T}{\sqrt{d}}\Big)(F_V(X_l) + P). \tag{4.13}$$

Here $\sigma(\cdot)$ represents the usual sigmoid operation, $\sigma(x) = 1/(1 + exp(-x))$. Here $P \in \mathbb{R}^{m \times d}$ is the sinusoidal positional encoding (Dosovitskiy et al., 2020). Finally the response from MhSAs are passed through Layer Normalization, $LN(\cdot)$ and Feed Forward Network, $FFN(\cdot)$ as follows,

$$X_{l+1} = LN\big(FFN(Attn_l(X_l)) + Attn_l(X_l)\big). \tag{4.14}$$

X_{l+1} represents the output token matrix of the l^{th} layer of *transformer encoder* and input to the $(l+1)^{th}$ layer.

4.3.7.2 Proposed Layer-Wise Adaptive Attention Aggregation Module

In the proposed approach the intermediate layer query, Q, key, K and value V matrices (see (4.12)) from each of the L_E layers of the *transformer encoder* are collected. Besides a Layer Specific Weight Generation module is utilised to generate a set of scalar weights $\mathcal{W} = \{w_1, w_2, \ldots, w_{L_E}\}$. Next, the generated weight for the l^{th} layer, w_l is multiplied with the query matrix, Q_l in the l^{th} layer of the *transformer encoder*. The key, value and the modified query matrices from all layers of *transformer encoder* are aggregated and processed to obtain the final feature diversity preserving token, X_{fdp}. Next X_{fdp} is added with the output from the final layer of the *transformer encoder*, X_{L_E}. The complete steps are formally written in the following expressions.

$$Q_c = \psi_C(w_1 * q_1, \ldots, w_{L_E} * q_{L_E}), K_c = \psi_C(k_1, \ldots, k_{L_E}), V_c = \psi_C(v_1, \ldots, v_{L_E}). \tag{4.15}$$

Here Q_c, K_c and V_c represent the aggregated query, key and the value matrices, respectively. The weight for layer l of the *transformer encoder* as generated from the Layer-Specific Weight Generation Module is w_l (see Fig. 4.4). Also, $\psi_C(\cdot)$ represents the usual concatenation operation of tensors. The layer specific weight w_l is obtained as follows,

$$w_l = \begin{cases} e^{-(l/l_E)}, & \text{if } l < \lceil \frac{L_E}{2} \rceil, \\ 1, & \text{otherwise.} \end{cases} \tag{4.16}$$

We get the final feature diversity preserving token as follows,

$$X_{fdp} = softmax\left(\frac{Q_c K_c^T}{\sqrt{d}}\right)V_c. \tag{4.17}$$

Next we justify the working of the above mentioned equations. In (4.17) the query, Q_c is matched with the key K_C and the compatibility between Q_c and K_C is used to update V_c.

We observe that earlier layers of *transformer encoder* preserve less amount of discriminative fine-grained information in comparison to the final stages of *transformer encoder* (Raghu et al., 2021, Nguyen et al., 2020). Thus the weights w_l are useful to balance between fine-grained features and coarse-grained features of different layers of *transformer encoder*. Finally, we add the obtained feature diversity preserving token, X_{fdp} with the token from the output of the final layer of the *transformer encoder* as written below,

$$X \leftarrow X_{L_E} + \beta * X_{fdp}. \tag{4.18}$$

Here β is an user-defined parameter. Next we explain the loss functions of the proposed approach.

4.3.8 Loss Components

The proposed KGT-Net is trained by a combination of four loss components. The first two loss components are *object cross-entropy loss* and *state cross-entropy loss*. Other two loss components are *object and state regularization losses*. Each of these loss components is explained below.

4.3.8.1 State and Object Cross-entropy Loss

We obtain the final token matrix from the final layer of the *transformer encoder* in the KGT-Net (see Section 4.3.5). Next the final token matrix is passed to the *state classifier* and the *object classifier*. State and object classifiers are implemented using an MLP with N_s and N_o number of output nodes, respectively. The output of the state classifier is the predicted state probability vector, $p(s) \in [0,1]^{N_s}$. Similarly the predicted object probability vector is $p(o) \in [0,1]^{N_o}$. $p(s)$ is used to train the model through cross-entropy loss with respect to state ground truth, GT_s. The state cross-entropy loss, $\mathcal{L}_s$ is defined as,

$$\mathcal{L}_s = \mathcal{L}_{CE}(p(s), GT_s). \tag{4.19}$$

Here $\mathcal{L}_{CE}(\cdot)$ represents the usual cross-entropy loss. Similarly $p(o)$ is used to train the model through cross-entropy loss with respect to object ground truth, GT_o. The object cross-entropy loss, $\mathcal{L}_o$ is represented as,

$$\mathcal{L}_o = \mathcal{L}_{CE}(p(o), GT_o). \tag{4.20}$$

Next we discuss the regularisation loss.

4.3.8.2 Regularization Loss

In the state regularisation loss defined in (4.9) there exist expectation terms. To evaluate the expectation $\mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=1)}(\cdot)$, we need to estimate the complete distribution of $p_\tau(\hat{g}_s, g_s|\tau=1)$ as observed from the following expression,

$$\mathbb{E}_{p_\tau(\hat{g}_s, g_s|\tau=1)} \log p_\tau(\tau=1|\hat{g}_s, g_s) = \int p_\tau(\hat{g}_s, g_s|\tau=1) \log q_\tau(\tau=1|\hat{g}_s, g_s) dq. \tag{4.21}$$

As all possible values of $\hat{g}_s$ and g_s cannot be obtained, the $p_\tau(\hat{g}_s, g_s|\tau = 1)$ distribution and the corresponding expectation $\mathbb{E}_{p_\tau(\hat{g}_s,g_s|\tau=1)}(\cdot)$ are intractable in practice. So to evaluate (4.9), inspired by the idea of Gibbs Sampling (Sanborn and Griffiths, 2007), we use sampling strategy. Specifically we utilise the observed responses from the MLP, $\psi_g(\cdot)$, to be the unbiased estimator of the expectation terms in (4.9) as follows,

$$\mathbb{E}_{p_\tau(\hat{g}_s,g_s|\tau=1)} \log \psi_g(\hat{g}_s, g_s) \approx \frac{1}{N_B} \sum_{i=1}^{N_B} \log \psi_g(\hat{g}_s^i, g_s^i). \tag{4.22}$$

Here we represent the mini-batch of input images as I_i, $i = \{1, 2, \ldots, N_B\}$, N_B being the number of images in the mini-batch. For the situation $\tau = 1$, we access the true ground truth state annotation to create g_s^i for the image I_i. We also generate the $\hat{g}_s^i$ from that input image, I_i using (4.1). For the situation $\tau = 0$, we resort to negative sampling strategy.

For each image I_i in the input mini-batch, we sample another image, $I_j, j \neq i$. The sampling process is designed to ensure that the images I_i and I_j represent different states. Thus from the image I_j we generate $\hat{g}_s^j$, using (4.1). Next we simplify (4.9) and (4.22) to obtain the state regularisation loss as follows,

$$\mathcal{L}_{reg\_s} \approx \sum_{i=1}^{N_B} \log \psi_g(\hat{g}_s^i, g_s^i) + \sum_{i=1, j\neq i}^{N_B} \log\big(1 - \psi_g(\hat{g}_s^j, g_s^i)\big). \tag{4.23}$$

Following a similar approach for pseudo object token regularisation, we can have the object regularisation loss as follows,

$$\mathcal{L}_{reg\_o} \approx \sum_{i=1}^{N_B} \log \psi_g(\hat{g}_o^i, g_o^i) + \sum_{i=1, j\neq i}^{N_B} \log\big(1 - \psi_g(\hat{g}_o^j, g_o^i)\big). \tag{4.24}$$

The final loss is computed using (4.19), (4.20), (4.23) and (4.24) as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}_s \ + \lambda_2 \mathcal{L}_o + \lambda_3 \mathcal{L}_{reg\_s} + \lambda_4 \mathcal{L}_{reg\_o}, \tag{4.25}$$

where λ_1, λ_2, λ_3 and λ_4 are scalar parameters within [0,1] and $\sum_{i=1}^{4} \lambda_i = 1$. We omit the $\frac{1}{N_B}$ term in the final expression in (4.23) and (4.24) from (4.22) as the effect of $\frac{1}{N_B}$ term is incorporated while obtaining the optimum values for the loss coefficients λ_3 and λ_4 as described in the final loss equation. Next, we explain the inference strategy.

4.3.9 Inference Strategy

During inference, the state and object probabilities i.e. $p(s)$ and $p(o)$ are obtained from the state and object classifiers, as shown in Fig. 4.2. For input image from the test set, the final predicted state is the state having highest probability in $p(s)$ (similar step is followed for object prediction). Next we present the experiments section.

Dataset →	C-GQA							MIT-States							UT-Zappos50k						
Algorithm↓	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*state*	*obj*
LE	0.9	0.4	11.8	4.4	4.6	17.7	19.9	3.0	2.0	10.7	15.0	20.1	23.5	26.3	26.4	25.7	53.0	61.9	41.0	41.2	69.2
AAO	0.8	0.4	11.4	4.8	4.5	18.0	20.5	2.5	1.6	9.9	14.3	17.4	21.1	23.6	21.5	25.9	59.8	54.2	40.8	38.9	69.6
TMN	1.7	0.8	18.8	6.1	6.4	16.5	25.6	3.5	2.9	20.2	20.1	13.0	23.3	26.5	36.8	29.3	58.7	60.0	45.0	40.8	69.9
SymNet	2.9	1.0	20.6	7.0	7.3	21.4	25.1	4.3	3.0	24.4	25.2	16.1	26.3	28.3	25.9	23.9	53.3	57.9	39.2	40.5	71.2
CompCos	3.6	2.6	28.1	11.2	12.4	20.2	28.4	5.9	4.5	25.3	24.6	16.4	27.9	28.3	32.8	28.1	59.8	62.5	43.1	43.3	73.0
CGE	5.3	3.6	31.4	14.0	14.5	15.2	30.4	8.6	6.5	32.8	28.0	21.4	30.1	34.7	34.8	33.5	64.5	71.5	60.5	48.7	76.2
SCEN	6.4	5.2	28.9	15.4	17.5	28.1	32.8	7.2	5.3	29.9	25.2	18.4	28.2	32.2	40.2	32.0	63.5	63.1	47.8	47.3	75.6
CANet	5.0	3.3	30.0	13.2	14.5	17.5	22.3	7.4	5.4	29.0	26.2	17.9	30.2	32.6	38.2	33.1	61.0	66.3	47.3	48.4	72.6
KGT-Net	**7.4**	**5.8**	**32.5**	**17.1**	**18.8**	**30.7**	**34.8**	**9.4**	**7.4**	**34.1**	**30.2**	**23.0**	**32.0**	**36.2**	**43.0**	**34.6**	**65.5**	**73.8**	**61.8**	**51.2**	**77.4**

Table 4.1: Results on the CW-CZSL problem.

4.4 Experiments

4.4.1 Implementation Details

The state and object knowledge matrices, KM_s and KM_o are initialized with 300 dimensional Word2Vec (Mikolov et al., 2013) word embedding of the corresponding state and object labels, respectively. The state and object latent projection functions, $\pi_s(\cdot)$ and $\pi_o(\cdot)$ are implemented using a three-layer MLP with 300 input and 384 output nodes. The KGT-Net consists of 13 layer of *transformer encoders*. For progressive tokenisation based projection module explained in Section 4.3.6, we use open-source implementation provided by (Yuan et al., 2021b). We have used the pre-trained weights provided by (Yuan et al., 2021b) as the initialization for the KGT-Net in our model. The state classifier is implemented using an MLP with 384 dimensional input nodes and N_s number of output nodes. The Object Classifier is implemented using an MLP with 384 number of input nodes and N_o number of output nodes. We use Adam optimizer (Kingma and Ba, 2015) with learning rate 0.0005. The scalar weights for loss components in (4.25) have following values $\lambda_1 = 0.45, \lambda_2 = 0.45$, $\lambda_3 = 0.05$ and $\lambda_4 = 0.05$. For all the datasets, we train the model for 50 epochs. Early stopping is used based on the performance of the model on the validation set. The batch size is taken as 64 for MIT-States (Isola et al., 2015) and C-GQA (Naeem et al., 2021). For UT-Zappos50k (Yu and Grauman, 2014, 2017) dataset batch size of 32 has been used. Also, if for a particular input image (ex: *rotten apple*), there is no image in the mini-batch with different state or object annotation (ex: *rotten banana* or *ripe apple*), we skip evaluating the regularisation loss for that image. The model is implemented using PyTorch (version 1.8) on a NVIDIA RTX Titan GPU with 24 GB memory, CUDA 10.1 and cuDNN 7.6 using Python 3.6 code on a PC with Intel i9 3.3 GHz processor, 64 GB RAM and Linux operating system. Next, we report the details of the datasets used for evaluation of KGT-Net.

4.4.2 Compared Algorithms

In this section, we compare our algorithm against other relevant state-of-the-art CZSL algorithms: LabelEmbed (LE) (Misra et al., 2017), AttrAsOp (AAO) (Nagarajan and Grauman, 2018), TMN (Purushwalkam et al., 2019), SymNet (Li et al., 2020), CGE (Naeem et al., 2021), CompCos (Mancini et al., 2021), SCEN (Li et al., 2022), CAPENet (Wang et al., 2023b), KG-SP (Karthik et al., 2022), SAD-SP (Liu et al., 2023) and DRANet (Li et al.,

Dataset →	MIT-States					C-GQA					UT-Zappos50k				
Algorithm↓	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*	*val AUC*	*test AUC*	*seen*	*unseen*	*HM*
LE	0.9	0.3	14.2	2.5	2.7	0.7	0.08	19.2	0.7	1.0	18.0	16.3	60.4	36.5	30.5
TMN	0.8	0.1	12.6	0.9	1.2	0.2	0.1	12.6	0.9	1.2	9.8	8.4	55.9	18.1	21.7
AAO	1.3	0.7	16.6	5.7	4.7	0.3	0.7	16.6	5.7	4.7	15.1	13.7	50.9	34.2	29.4
SymNet	1.2	0.8	21.4	7.0	5.8	0.9	0.4	26.7	2.2	3.3	21.4	18.5	53.3	44.6	34.5
CompCos	1.9	1.6	25.4	10.0	8.9	1.0	0.4	28.4	1.8	2.8	22.9	21.3	59.3	46.8	36.9
CGE	1.4	1.0	32.4	5.1	6.0	2.2	0.5	32.7	1.8	2.9	23.8	23.1	61.7	47.7	39.0
KG-SP	1.5	1.3	28.4	7.5	7.4	1.2	0.8	31.5	2.9	4.7	27.8	26.5	61.8	52.1	42.3
SAD-SP	1.7	1.4	29.1	7.6	7.8	1.3	1.0	31.0	3.9	5.9	29.1	28.4	63.1	54.7	44.0
DRANet	2.0	1.5	29.8	7.8	7.9	1.4	1.05	31.3	3.9	6.0	28.0	28.8	65.1	54.3	44.0
KGT-Net	**3.0**	**2.5**	**35.1**	**11.2**	**11.0**	**2.8**	**2.1**	**33.1**	**6.3**	**6.8**	**30.0**	**29.9**	**66.6**	**55.9**	**45.2**

Table 4.2: Results on the OW-CZSL problem.

2023b). The brief literature review of these algorithms have been done in Section 4.2. The results for all these methods are reproduced from the papers (Naeem et al., 2021) and the corresponding open source implementations. Next we report the results of our approach.

4.4.3 Quantitative Results

The results of the proposed approach and other state-of-the-art approaches on the CW-CZSL and OW-CZSL evaluation protocols are shown in Tables 4.1 and 4.2, respectively. The last row on both the tables show result of the proposed KGT-Net. Our algorithm reports better result over state-of-the-art algorithms on all the metrics on each of the three datasets. More specifically, on C-GQA in CW-CZSL, we report improvements in top-1 *test* and *val AUC* by 0.6% and 1.0% respectively. On *seen* and *unseen* class recognition, proposed approach reports 1.1% and 1.7% improvement in CW-CZSL. Similar improvement is observed on MIT-States and UT-Zappos50k.

One of the major contributions of the proposed KGT-Net is its capability to adapt to partial state and object feature association through pseudo state and object token generation. As explained in Section 4.2, along with our approach, TMN and CGE have attempted to solve the partial feature association problem. As shown in Table 4.1, KGT-Net reports better results than both TMN and CGE.

Further, it can be observed that the problem of partial feature association is more challenging for state features than the object features. For example, the features of the state *peeled* may vary in a compositions *peeled apple* and *peeled orange* depending on how much peeled the fruit is. So we argue that the ability to adapt to partial association between state and object features should give rise to better state recognition accuracy. Evidently as shown in Table 4.1, KGT-Net performs at least 1.9% better state recognition accuracy over TMN and CGE respectively on C-GQA, MIT-States and UT-Zappos50k.

On unseen classes of C-GQA, KGT-Net achieves at least 10.1% better accuracy over the earlier algorithms (LabelEmbed, AttrAsOp, TMN, SymNet). This can be attributed to the following justification. C-GQA comprises of three times more objects and states than MIT-States dataset (see Table 2.1). Consequently, the number of possible state-object compositional classes of C-GQA is increased multiple times over MIT-States. Thus, earlier algorithms (SymNet, LabelEmbed, AttrAsOp and TMN) perform poorly on C-GQA dataset.

On UT-Zappos50k, KGT-Net (and also competing approaches SymNet, AAO and LE)

has reported lower *seen* accuracy in comparison to *unseen* accuracy. This can be justified as follows. The visual features for an image of state-object composition can be decomposed into three components, viz. (a) uniquely identifiable features of the state, (b) uniquely identifiable features of the object and (c) features that are formed due to the composition between the particular state and object. The first two components of features are discriminative in nature and specifically help the trained model to better generalise to unseen compositions. The third component of visual features, on the contrary are exclusive to the state-object composition. The third feature specifically helps to better recognise seen state-object compositions. In the context of UT-Zappos50k, states are the materials of the shoes (ex: *canvas*, *cotton*, *leather* etc.). The objects are annotated as the shoe types (ex: *boots-ankle*, *boots-knee-high*, *sandals*, *shoes-heels* etc.). As apparently visible from the human eye, there exists less amount of variability of state features in the state-object compositions for UT-Zappos50k. The less amount of variability of state features in images for UT-Zappos50k helps the KGT-Net and other competing approaches to perform better disentanglement of the state-object feature. Effective disentanglement of state and object features increases the discriminative capability of the model and helps the model to better generalise the unseen state-object compositions during test.

Usually the algorithms specifically designed for the CW-CZSL evaluation protocol fails to perform effectively in OW-CZSL due to the unrestricted large search space in OW-CZSL. The difference in performance of CW-CZSL and OW-CZSL, is more significant in large datasets (e.g. C-GQA), as due to the large size of the state-object compositional space. In the OW-CZSL problem, proposed approach either outperforms or reports competitive results with respect to the state-of-the-art-approaches. Specifically on the *test AUC* metric on OW-CZSL, proposed approach reports 1.0%, 1.05% and 1.1% improvement in *test AUC* for MIT-States, C-GQA and UT-Zappos50k, respectively. Similar improvement is reported over the *seen*, *unseen* and the *HM* metrics on all three datasets. Next we present the qualitative image classification results.

4.4.4 Qualitative Image Classification Results

We have shown some qualitative image classification results from MIT-States (Isola et al., 2015) and C-GQA (Naeem et al., 2021) datasets in Fig. 4.5. The first row represents the qualitative classification results from C-GQA and the second row represents results from MIT-States dataset. Evidently KGT-Net has achieved competitive results on both datasets. It has successfully recognised most of the images from MIT-States and C-GQA datasets.

However, KGT-Net has failed to recognise some of the images belonging to classes *broken road* and *wet cat*. We note the *top-3* σ_s and σ_o values for the incorrect predictions in Fig. 4.5. For the *broken road* image (the ground truth is *broken road*, whereas our prediction is *empty highway*), the *top-3* σ_s values and the corresponding states are $\{$*empty*$\rightarrow \sigma_s = 0.24$, *cracked* $\rightarrow \sigma_s = 0.18$, *broken* $\rightarrow \sigma_s = 0.13\}$. The correct state, i.e. *broken* has obtained third highest σ_s value.

Evidently all of the above mentioned states are relevant to the compositional image of *broken road*. Atzmon *et. al.* (Atzmon et al., 2020b) conducted a detailed analysis of the MIT-States dataset and concluded that there exists high amount of noise in MIT-States due to improper labelling. More specifically there are closely meaning labels in MIT-States dataset and often more than one labels are relevant for an image under consideration. All the *top-3*

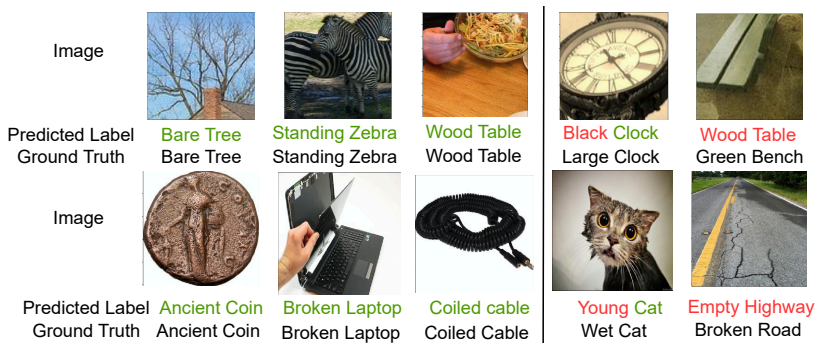


Figure 4.5: Qualitative image classification results on two real life datasets, C-GQA (first row) and MIT-States (second row): The green coloured texts represent the correct label prediction and red coloured texts represent incorrect classification. The first three columns represent images where the state and object are correctly predicted. The fourth column represents images for which object is correctly predicted but state is incorrectly predicted. Finally, the last column represents images where the state and object both are incorrectly predicted by our model.

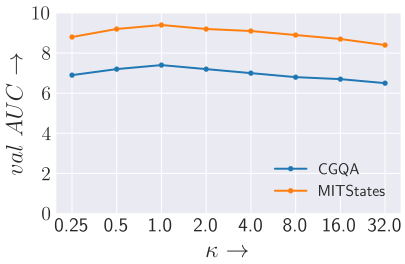


Figure 4.6: Experiment to evaluate the effectiveness of adapting to the partial association between features of states and objects. The κ parameters in (4.26) are varied and the corresponding variation in *val AUC* is reported.

state predictions by KGT-Net are relevant to the image under consideration. We obtain incorrect prediction due to very similar state labels being present in the MIT-States.

Similarly, we also note the *top-3* σ_o and the corresponding objects for the *broken road* image. The *top-3* σ_o and the corresponding objects are $\{$*road*$\rightarrow \sigma_s = 0.28$, *highway*$\rightarrow \sigma_s = 0.22$, *basement*$\rightarrow \sigma_s = 0.10\}$. In this case also the correct object prediction is present within the *top-3* σ_o retrievals. Thus the incorrect object prediction for the *broken road* image can be attributed to the ambiguous object label annotation in MIT-states dataset.

The *top-3* σ_s and the corresponding states for the *wet cat* image (ground truth is *wet cat*, prediction of KGT-Net is *young cat*) are $\{$*young*$\rightarrow \sigma_s = 0.22$, *wet*$\rightarrow \sigma_s = 0.18$, *furry*$\rightarrow \sigma_s = 0.14\}$. As before, the correct state prediction is present within the *top-3* σ_s retrievals. This can also be explained by the multiple relevant state labels in the MIT-States dataset in context of the input image. Next we present the ablation study for the proposed approach.

4.4.5 Ablation Study

In this section we discuss the utility of different components and parameters as used in the proposed KGT-Net.

Loss Components				Results		
$\mathcal{L}_s$	$\mathcal{L}_o$	$\mathcal{L}_{reg\_s}$	$\mathcal{L}_{reg\_o}$	*val AUC*	*state*	*object*
✓	✓	×	×	6.3	28.8	35.7
✓	✓	✓	×	6.6	31.7	35.9
✓	✓	×	✓	6.4	28.8	37.1
✓	✓	✓	✓	**7.4**	**32.8**	**37.8**

Table 4.3: The ablation study on the loss components on C-GQA.

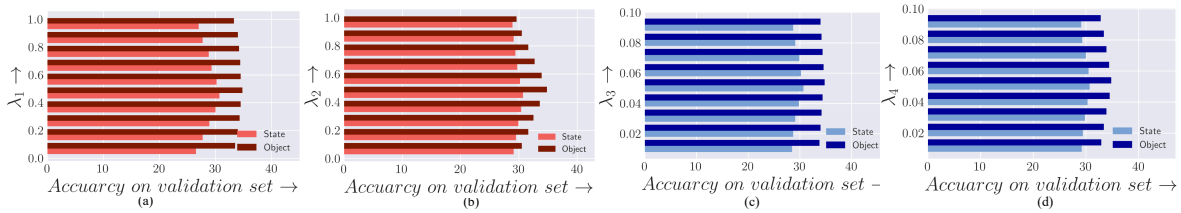


Figure 4.7: Visualization of the effect of parameters λ_1, λ_2, λ_3 and λ_4 in (4.25) on the state and object recognition accuracies in the validation set of C-GQA dataset.

4.4.5.1 Ablation Study on Loss Components

Here the effect of different loss components on the final recognition of our proposed KGT-Net is evaluated. The detailed results are shown in Table 4.3. We consider the $\mathcal{L}_s$ and $\mathcal{L}_o$ as part of our baseline model as they are essential to train the KGT-Net. More specifically, we have experimented with effects of the other two loss components, $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$. While excluding the loss components $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$, we have also excluded the pseudo state and object tokens from the model, respectively.

The last row of Table 4.3 shows that the inclusion of both $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$ (and correspondingly pseudo state and object tokens) improves the state and object recognition accuracies on the validation set by 4.0% and 1.9%, respectively. Similar improvement of 1.1% is observed on the validation set AUC (see the third and last rows of Table 4.3).

Among $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$, $\mathcal{L}_{reg\_s}$ reports a stronger effect on the model's performance. Comparing the last row and second last row of Table 4.3, $\mathcal{L}_{reg\_s}$ reports improvements of 1.0% and 4.0% in the *val* AUC and the state recognition accuracy, respectively. On the other hand, observing the last and the third last rows of Table 4.3, $\mathcal{L}_{reg\_o}$, provides improvements of 0.8% and 1.9% in the *val* AUC and the object recognition accuracy respectively.

The stronger effect of $\mathcal{L}_{reg\_s}$ (and correspondingly pseudo state token) over $\mathcal{L}_{reg\_o}$ (and correspondingly pseudo object token) can be justified from the fact that the partial association is more dominant for state features than the object features. Evidently, $\mathcal{L}_{reg\_s}$, intended for regularisation of the pseudo state tokens and correspondingly the partial membership values, provide higher improvement in final *val* AUC over $\mathcal{L}_{reg\_o}$. Next we discuss the effect of the loss coefficients in final loss of (4.25).

4.4.5.2 Ablation Study on Loss Coefficients

The variation in state and object recognition accuracies of the validation set due to change in λ_1, λ_2, λ_3 and λ_4 values of (4.25) are shown in Fig. 4.7. In (4.25), λ_1 controls the contributions of loss component $\mathcal{L}_s$, i.e. the state cross-entropy loss. In Fig. 4.7(a), we

Figure 4.8: Results from the test set of C-GQA dataset and the corresponding *top-3* σ_s^i values as predicted by SWPG module. For an input test image, if the highest value of σ_s^i, $(i \in \{1, 2, \ldots N_s\})$ is obtained for the ground truth state, then the image is shown in green border else in red border.

can see that due to variation in λ_1, we can see large variation in state recognition accuracy. However, the object recognition accuracy is relatively less dependent on λ_1. Similarly in Fig. 4.7(b), we can see large variation in object recognition accuracy, due to variation in λ_2. This can be justified as λ_2 controls the contributions of loss component $\mathcal{L}_o$ in (4.25). From Fig. 4.7(a) and 4.7(b), it can be observed that the best state and object recognition accuracies are obtained on the validation set for $\lambda_1 = 0.45$ and $\lambda_2 = 0.45$. Similarly, in (4.25), λ_3 and λ_4 control the contributions of loss components $\mathcal{L}_{reg\_s}$ and $\mathcal{L}_{reg\_o}$, respectively. Figs. 4.7(c) and 4.7(d) represent the variation in state and object accuracies due to variations in λ_3 and λ_4, respectively. Observing Figs. 4.7(c) and 4.7(d), we see that the best state and object recognition accuracies are obtained on the validation set for $\lambda_3 = 0.05$ and $\lambda_4 = 0.05$. Fig. 4.8 shows a set of test images from the C-GQA dataset and the corresponding *top-3* σ_s^i values $(i \in \{1, 2, \ldots N_s\})$ as obtained from the SWPG module (see Fig. 4.2). As in (4.1), σ_s^i value represents the extent of features of the i^{th} state in the visual features of the input image. Evidently results in Fig. 4.8 justifies that the σ_s^i values obtained from the SWPG module are effective in understanding the extent of association of state features in a compositional image. Next we evaluate the utility of the proposed partial association strategy.

4.4.5.3 Ablation Study on Effectiveness of Partial Association Between State and Object Features

As discussed in Section 4.3.4, in CZSL, there may be high amount of variability among the images from same state-object compositional class. The visual features of images of *peeled orange* is shown in Fig. 4.3. Thus the association between state and object features may be partial in nature. One of the major contributions of the proposed KGT-Net is its ability to adapt to partial association between state and object features.

Based on the input image features, the SWPG and OWPG generate a set of state and object gating weights, σ_s^i and σ_o^j, $i \in \{1, 2, \ldots, N_s\}$ and $j \in \{1, 2, \ldots, N_o\}$ (see Fig.4.2). The σ_s^i and σ_o^j values represent the extent of presence of i^{th} state and j^{th} object in the compositional image under consideration. In our approach, the σ_s^i and σ_o^j values can be any number in the closed interval $[0, 1]$, $\sum_{i=1}^{N_s} \sigma_s^i = 1$ and $\sum_{j=1}^{N_o} \sigma_o^j = 1$. Based on the σ_s^i and σ_o^j values, we incorporate external knowledge about a particular state and object in the model

for the input image (see (4.1))

In existing CZSL algorithms either a state can be present or be absent in a composition. In context of existing CZSL algorithms, the σ_s^i will be 1 for any one state and will be 0 for all other states. To investigate the effectiveness of modelling the partial association, we add an additional *temperature-based annealing*, $\mathcal{A}(\cdot)$ on the σ_s^i and σ_o^i values obtained from the SWPG and OWPG as follows,

$$\mathcal{A}(\sigma_s^i) = \frac{\frac{\sigma_s^i}{\kappa}}{\sum_{m=1}^{N_s} \frac{\sigma_s^m}{\kappa}}, i \in \{1, 2, \ldots, N_s\} \text{ and } \mathcal{A}(\sigma_o^j) = \frac{\frac{\sigma_o^j}{\kappa}}{\sum_{m=1}^{N_o} \frac{\sigma_o^m}{\kappa}}, j \in \{1, 2, \ldots, N_o\}. \quad (4.26)$$

Here κ is the temperature parameter and is set by the user. The variation in *val AUC* due to variation in κ is shown in Fig. 4.6. The *val AUC* drops as κ increases over 1.0. If κ decreases below 1.0, *val AUC* also decreases. The highest value of *val AUC* is achieved for $\kappa = 1$. The variation of *val AUC* with respect to κ is justified next.

Evidently as $\kappa \to \infty$, $\mathcal{A}(\sigma_s^i) \to \frac{1}{N_s}, \forall i$. Thus the σ_s^i values for all states are equal. Subsequently, following (4.1), external knowledge about all states are incorporated in creating the pseudo state token. This leads to incorporation of knowledge about unnecessarily many states while creating the pseudo state token. The inclusion of knowledge from too many states reduces the discriminative capability of the proposed KGT-Net. Thus the *val AUC* reduces with increasing κ beyond 1.0. However, it is practically infeasible to implement $\kappa \to \infty$ situation in code (which leads to $\mathcal{A}(\sigma_s^i) \to \frac{1}{N_s}$). So we have evaluated the boundary condition using an *average-pool* operation on σ_s^i. Thus after the *average-pool*, we obtain $\mathcal{A}(\sigma_s^i) = \frac{1}{N_s}$. On using the *average-pool*, *val AUC* achieved on MIT-States and C-GQA datasets are 8.5 and 7.0, respectively. Additionally, if $\kappa \to 0$, the distribution of $\mathcal{A}(\sigma_s^i)$ collapses to only one value, i.e.the highest $\mathcal{A}(\sigma_s^i)$ value is only preserved and all other $\mathcal{A}(\sigma_s^i)$ values are reduced to zero. Thus $\kappa \to 0$ represents the approaches of existing CZSL algorithms (Purushwalkam et al., 2019, Nagarajan and Grauman, 2018, Naeem et al., 2021, Mancini et al., 2021, Misra et al., 2017), where full association between state and object features is considered. Thus as κ value is reduced, the *val AUC* also decreases as the partial contribution from state features is diminished.

The proposed approach obtains highest *val AUC* corresponding to $\kappa = 1$ (see Fig. 4.6). This experimental observation justifies using un-altered σ_i^s values as obtained from the SWPG, in our approach. Similar argument also applies for variation of *val AUC* due to change in $\mathcal{A}(\sigma_o^j)$. Next we present the ablation study on using distinct sources of external knowledge in the proposed KGT-Net.

4.4.5.4 Ablation on the External Knowledge

KGT-Net incorporates external knowledge about the states and objects, while creating the state and object Knowledge Matrices, KM_s and KM_o, respectively (see Section 4.3.4). We have experimented with four external knowledge sources. These external knowledge sources have been broadly grouped into two categories, i.e. word embedding based sources and knowledge graph based sources. Word embedding based knowledge sources include Word2Vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2017) and GloVe (Pennington et al., 2014). Knowledge graph based knowledge source includes embeddings from the ConceptNet (Speer et al., 2017).

Next, the proposed approach is trained while KM_s and KM_o are created from external information Word2Vec and GloVe. When KM_s and KM_o are initialised from FastText based embedding, *val AUC* of 8.9 is achieved on MIT-States. Similarly, using GloVe and Word2Vec our algorithm reports *val AUC* of 8.4 and 9.4, respectively. ConceptNet reports significantly lower *val AUC* of 7.9. Next we briefly report the justification of variation in *val AUC* for respective external knowledge sources.

Evidently, Word2Vec and FastText are the two best performing knowledge sources in the current context. FastText utilises the character n-gram and is specially effective at out-of-vocabulary word. However, all three CZSL datasets (see Table 2.1) consist of commonly used words as state and object labels. Thus FastText provides no additional advantage in the context of CZSL. Also ConceptNet provides lowest *val AUC* among other knowledge sources. In this work, we have attempted to model partial association between state and object features. However, the knowledge graphs already incorporate the semantic relationship within states and within objects. Thus using the ConceptNet based external knowledge source causes the network to over-fit and hence inferior performance is observed.

4.4.5.5 Analysis on the Effectiveness of the Tokenisation based Projection Module

As explained in Section 4.3.6, the proposed approach utilises a tokenisation based projection module (Yuan et al., 2021b) in the *transformer encoder*. To evaluate the effectiveness of the tokenisation based projection module, we train the model with the tokenisation based projection as well as the projection module of the vanilla transformer (Dosovitskiy et al., 2020). On MIT-States dataset, the vanilla projection module achieves *val AUC* of 7.2. The *val AUC* achieved by the proposed approach using the tokenisation based projection module is 9.4. The 2.2% improvement in *val AUC* can be attributed to the overlapping patch generation approach in the tokenisation strategy. The overlapping patch generation approach helps to make the individual patches more aware of the local neighbourhood information and thus helps to extract effective image features.

Additionally, we also investigate the effectiveness of the proposed Layer-Wise Adaptive Attention Aggregation Module (LAAg). By training the model in absence of the LAAg, the proposed approach achieves *val AUC* of 7.0 and 9.1 on C-GQA and MIT-States datasets, respectively. Thus from Table 4.1 last row, we can see that due to inclusion of LAAg module we can achieve higher *val AUC* by at least 0.3% on both MIT-Sates and C-GQA datasets. This result can be justified to the following fact. The LAAg module helps to preserve diversity in the feature and helps to better recognition of state. Evidently the improvements in the results justify the use of LAAg.

4.5 Summary

In this work we have proposed a *Knowledge Guided Transformer* for the CZSL problem. The proposed model integrates knowledge about the possible states and objects with the proposed KGT-Net to better recognise the unseen state-object compositions. One of the major contributions of our work is that the proposed model incorporates possibility of partial association between state features and object features.

As discussed, CZSL approach requires annotated images for only a subset of state object compositions in the train set. Still CZSL is computationally expensive, if we consider the

fact that the images in the training set require annotations for both state and object. In the next chapter, we explore the partially supervised CZSL or the pCZSL problem, where for each image either the state or the object annotation is only available.

Chapter 5

Partially Supervised Compositional Zero-Shot Learning

5.1 Introduction

In this chapter, we attempt to address the problem of the partially supervised Compositional Zero-Shot Learning (pCZSL) (Karthik et al., 2021). In pCZSL, either the state annotation or the object annotation is available during training for each image. For example, corresponding to the image of *young elephant* as in Fig. 1.3, only *young* annotation is available in the training set of pCZSL. The pCZSL is a more realistic problem than the existing CZSL as pCZSL requires even less amount of labelled data than CZSL.[1]

A major challenge in pCZSL is that the visual features of the state depend largely on the context i.e. the object in the composition. For example, the state *ripe* produces distinctive visual features in the compositions *ripe orange* and *ripe banana*. This context dependency introduces intra-class variations in the features of the state *ripe*. Consequently, any algorithm for pCZSL struggles to learn unique and discriminative features of the state *ripe*. We refer this problem of pCZSL as *Contextual Dependency* (ContDp) challenge.

Vision Transformers (ViTs) (Dosovitskiy et al., 2020) due to patch level processing of the input image, are able to process the global context in an image. However existing ViTs (Carion et al., 2020, Dosovitskiy et al., 2020, Yuan et al., 2021b) suffer from the drawback that the computational costs of the ViTs are quadratic function of the number of patches. *Swin* transformer (Liu et al., 2021) utilises *Window based Multi-head Self-Attention (W-MhSA)* and *Shifted Window based Multi-head Self-Attention (SW-MhSA)* modules (Liu et al., 2021) to process the global context dependency in computational time linearly proportional to number of patches. We utilise a *swin* transformer (Liu et al., 2021) based *Hierarchical Feature Extractor (HFE)* in the proposed architecture (see Fig. 5.1).

It may be observed from Fig. 1.3 that in the images of the compositions *old elephant* and *young elephant*, in reference to the background, an *old elephant* will be of larger size (scale) than a *young elephant*. Thus understanding the difference between scale of the *elephant* and the corresponding background is a key challenge in this case. We refer this problem as *Cross-*

[1]Part of the work done in this chapter is under review as follows:
Aditya Panda and Dipti Prasad Mukherjee, "Partially Supervised Unseen State-object Image Classification by Discriminative Context Aggregation", under review in IEEE Transactions on Image Processing, 2025.

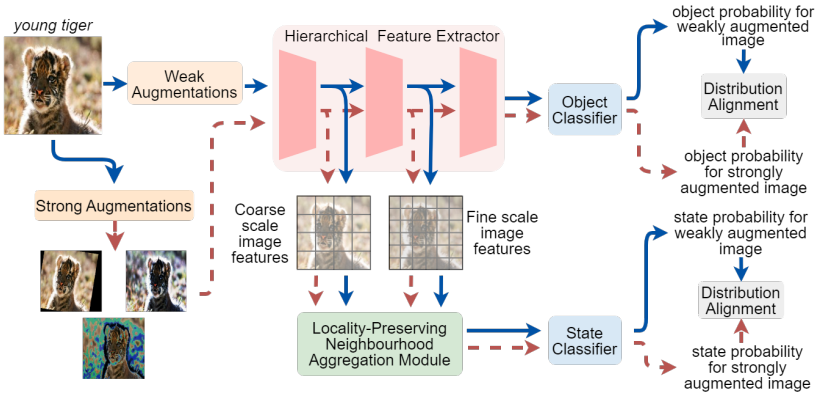


Figure 5.1: Block-level representation of our approach.

Scale Consistency (CrSC) challenge. Understanding the cross-scale consistency has not been attempted by the existing CZSL approaches (Misra et al., 2017, Naeem et al., 2021, Mancini et al., 2021, Xu et al., 2021b, Karthik et al., 2022). For effectively processing the cross-scale consistency we propose Locality-Preserving Neighbourhood Aggregation (LoPNA) module, as explained next.

Existing CZSL approaches (Saini et al., 2022, Li et al., 2022, Karthik et al., 2022) mainly use the features from the deepest layers of the image feature extractor. The deepest layer features from the image feature extractors although contain abstract semantics about the objects, still lack sufficient scale-related information (Wang et al., 2022a, Bose et al., 2024). Thus utilising features from final layer of image feature extractors serves as bottleneck for the existing CZSL approaches. In the proposed *swin* transformer based HFE, intermediate features of the different layers are of varying spatial resolution or scale and sensitive to objects of different scales in the image (Wang et al., 2022a).

However features from the shallower layers of HFE comprise of lower number of feature channels and have higher spatial dimensions in comparison to features from deeper layers of the HFE. So aggregating features from multiple layers of the HFE requires designing effective approach to align the dimensions of features from intermediate layers. Also, in CZSL the visual features of the state primitive are often significant over a small locality in the input image. For example the state *rotten* in *rotten apple* can be identified by specific *textures* and *dis-coloured patches*, which are locally identifiable. To alleviate the aforementioned challenges, the LoPNA module identifies locally discriminative feature and performs neighbourhood based self-attention to aggregate features from multiple intermediate layers of the HFE.

To leverage the unlabeled data in pCZSL, we propose a novel Class-balanced and Confidence-scaled Distribution Alignment (CCDA) loss based on α-divergence (Póczos and Schneider, 2011, Rényi, 1961). Next, the class probability distributions from a weakly augmented and a strongly augmented versions of the input image are aligned using the CCDA loss. *Strongly* augmented images incorporate larger amount of perturbation in comparison to *weakly* augmented images. Hence the network can recognise the state and object in a *weakly* augmented image, easier in comparison to the *strongly* augmented image. In pCZSL, when state or object annotation is not available, we use the prediction from the *weakly* augmented images

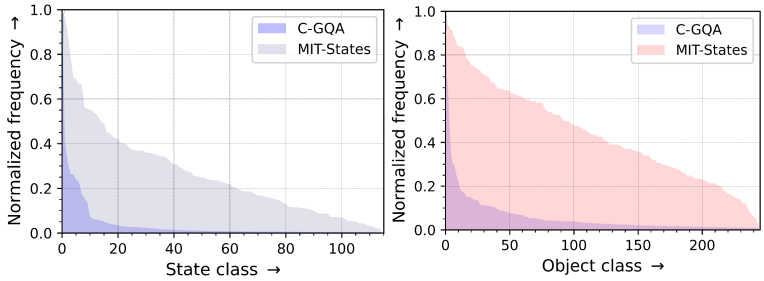


Figure 5.2: The figures represent the (normalized) number of images belonging to different state and object classes of MIT-States (Isola et al., 2015) and C-GQA (Naeem et al., 2021). The skewed nature of the graphs indicate the imbalance in the number of images per class in pCZSL datasets. Please note that the classes (state and object) are sorted based on their frequency. For C-GQA, the classes with very less number of images are ignored in the plots.

as already *explored knowledge*, to supervise the predictions corresponding to the *strongly* augmented images. Next we utilise the proposed CCDA loss to enforce similar prediction corresponding to *weakly* and *strongly* augmented images. Thus in an attempt to enforce similarity between *strongly* and *weakly* augmented images, we encourage the network to extract the meaningful and transformation invariant features from *strongly* augmented images as well. This approach helps to utilise the large amount of partially labelled images as available in pCZSL. Next we utilise the proposed CCDA loss to enforce similar prediction corresponding to *weakly* and *strongly* augmented images. Thus in an attempt to enforce similarity between *strongly* and *weakly* augmented images, we encourage the network to extract the meaningful and transformation invariant features from *strongly* augmented images as well.

The α-divergence proposed in (Rényi, 1961) uses same value of α across images of different classes. As shown in Fig. 5.2, CZSL datasets (Isola et al., 2015, Naeem et al., 2021, Yu and Grauman, 2014, 2017) have imbalance in number of images of different classes. Instead of using a fixed α value, we evaluate a *class-specific* α value for each state and object class while calculating the corresponding CCDA loss. The *class-specific* α values are inversely proportional to the number of images in the class under consideration. Thus for images corresponding to *minority classes*, corresponding *class-specific* α is high and subsequently the CCDA loss is also high.

In the proposed CCDA loss, we evaluate a *confidence* scaling factor, as the ratio of the maximum *confidence* achieved by the model to the sum of the *confidences* in all other non-maximal classes in the predicted class probabilities. The *confidence* scaling factor is next multiplied with the CCDA loss. Thus images which achieve high *confidence* on the *weakly* augmented image will have higher *confidence* scaling factor and higher contribution of the CCDA loss in the final loss. Existing augmentation approaches (Gong et al., 2021, Sohn et al., 2020, Berthelot et al., 2019b,a) mainly utilise parametric approach to filter out less confident predictions from the distribution alignment loss, whereas our approach is non-parametric.

Besides evaluating the proposed approach on pCZSL, we also evaluate the proposed approach on the Open World CZSL (OW-CZSL) (Mancini et al., 2021). In OW-CZSL, for each image in the training set both state and object annotations are provided. Proposed approach is evaluated on three benchmark datasets, MIT-States (Isola et al., 2015), C-GQA (Naeem et al., 2021) and UT-Zappos50k (Yu and Grauman, 2014, 2017). Proposed approach reports

competitive results in all the datasets across the evaluation protocols. The contributions of this chapter are summarised as follows.

- To better capture the cross-scale features, we propose a novel multi-scale feature aggregation approach using a Locality Preserving Neighbourhood Aggregation approach.
- An α-divergence based *distribution alignment* loss is proposed for leveraging the partially labelled data in pCZSL.
- We utilise a class-balanced scaling approach to address the data-imbalance issue in CZSL datasets. To the best of our knowledge, we are the first in attempting to address the data imbalance issue in CZSL. We also propose a non-parametric confidence based scaling approach in the *distribution alignment* loss.
- We achieve better results over existing approaches on both pCZSL and OW-CZSL problems on three widely used benchmark datasets.

The remaining part of the chapter is organised into four sections. The literature review is presented in Section 5.2. The methodology of the proposed approach is discussed in Section 5.3. Experimental protocols and the results are reported in Section 5.4. Finally the chapter is summarised in Section 5.5.

5.2 Related Works

pCZSL: Recently, KG-SP (Karthik et al., 2022) and Pro-CC (Huo et al., 2024) attempted to address the pCZSL problem. KG-SP utilised external knowledge (ConceptNet (Speer et al., 2017)) to estimate the feasibility of unseen state-object composition. Our approach on the contrary, does not require any external knowledge during training or inference stage. In Pro-CC (Huo et al., 2024), the intermediate response from the Object Classifier is added with the intermediate response in the State Classifier, in an attempt to process the context dependency between the state and object features. The classifiers in Pro-CC are implemented using the MLPs and are less capable to process the local structures in the image features. We argue that the context dependency between state and object visual features can be more effectively processed through sharing the feature response from intermediate layers of the image feature extractors. Besides, the existing CZSL algorithms (Karthik et al., 2022, Huo et al., 2024, Naeem et al., 2021, Purushwalkam et al., 2019, Nagarajan and Grauman, 2018) have not attempted to address the imbalance problem in the benchmark CZSL datasets.

Multi-scale Feature Aggregation: To extract multi-scale features from the image, earlier works (Vaillant et al., 1994, Rowley et al., 1995, Dollár et al., 2014) used multiple filters to sample a number of images, each at different scale from the input image. Subsequently the sampled images were passed through the feature extractor individually. However these approaches required one forward pass for each of the sampled images. On the contrary, Lin et al. (2017) leveraged the in-network feature pyramid of a CNN based network (He et al., 2016) and used intermediate features from different layers of the feature extractor as the multi-scale feature. The advantages of our approach over the existing feature pyramid based approaches (Lin et al., 2017, Guo et al., 2020, Girshick et al., 2014, Wang et al., 2020) are discussed next. To align the spatial dimensions of intermediate features, Feature Pyramid

Network (FPN) based approaches (Guo et al., 2020, Girshick et al., 2014, Wang et al., 2020, Lin et al., 2017) use interpolation techniques to upsample the lower spatial dimensional features from the deeper layers of the HFE. The aforementioned approach is ineffective to upsample the semantically rich features from deeper layers (Wang et al., 2022a). We use a projection function to project the features from the shallower layers to same spatial resolution of the deeper layer features. Next, existing approaches (Guo et al., 2020, Girshick et al., 2014, Wang et al., 2020, Lin et al., 2017) use 1×1 convolutional filter to align the channel dimension of the features. We observe the state features in state-object composition is locally discriminative and it is ineffective to capture the state-object context dependency using the 1×1 filter. We instead propose a Locality Preserving Neighbourhood Aggregation approach. Lastly after aligning the spatial and channel dimensions of features from the different layers, existing approaches (Guo et al., 2020, Girshick et al., 2014, Wang et al., 2020, Lin et al., 2017) element-wise add the individual layer features. The element-wise addition may dilute the fine-grained visual cues in the deeper layer features. We avoid the element wise addition and utilise a concatenation of the individual layer features.

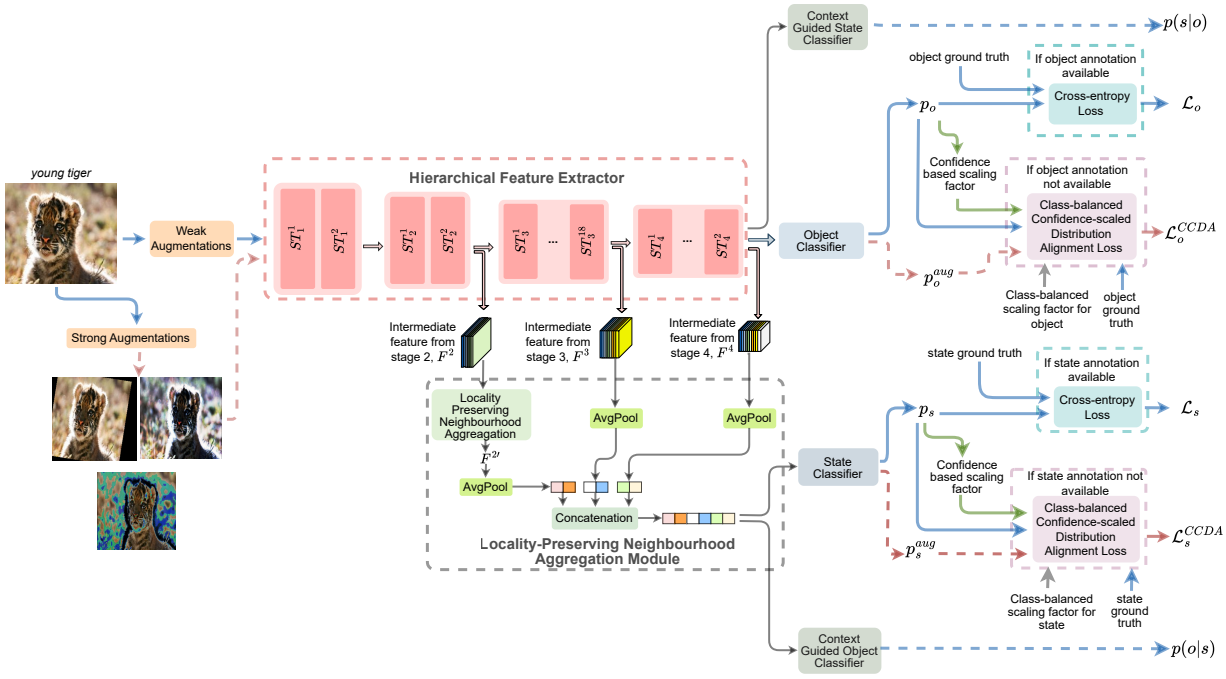


Figure 5.3: Proposed architecture consists of a *swin* transformer based Hierarchical Feature Extractor (HFE), Locality-Preserving Neighbourhood Aggregation (LoPNA) module and *distribution alignment* loss. The ST_i^j represents the j^{th} *swin* transformer block (ST) in layer i (Liu et al., 2021).

Partially-supervised Learning: This type of learning falls in the broad paradigm of semi-supervised learning (SSL) algorithms (Yarowsky, 1995, Lee et al., 2013, Riloff and Wiebe, 2003, Loog and Jensen, 2014). A group of approaches (Yarowsky, 1995, Lee et al., 2013, Riloff and Wiebe, 2003) considers a pseudo-label based approach for the SSL problem. In this approach a model, trained on the labelled data of the SSL, is referred to as a *teacher*. Next, the *teacher* model is used to predict labels of the unlabeled data and the predicted

labels are referred to as pseudo-label. The labeled images and pseudo-labels of the unlabeled images jointly supervises another model, referred to as *student*. Another group of approaches (Berthelot et al., 2019a, Gong et al., 2021, Sohn et al., 2020) considers augmentation based strategies to utilise the unlabeled data in a semi-supervised scenario. The *weak* and *strong* augmentations to an input image are utilized in (Sohn et al., 2020). The prediction corresponding to the *weakly* supervised image is referred to as pseudo-label. The pseudo-label guides the model's prediction predictions from the *strongly* augmented images. A convex combination of the *strongly* and *weakly* augmented images is used to regularise the model predictions (Berthelot et al., 2019b). Berthelot et al. (2019a), instead of considering a single *strongly* augmented image, consider multiple *strongly* augmented images. Gong et al. (2021) used an α-divergence (Póczos and Schneider, 2011, Rényi, 1961) based *distribution alignment* loss on the prediction by the model corresponding to *strongly* and *weakly* augmented images. Alike the work by Gong et al. (2021), we also used *distribution alignment* loss to encourage similarity between model predictions for *weakly* and *strongly* augmented images. Proposed work has two improvements over the existing work by Gong et al. (2021). First, the CZSL algorithms suffer heavily due to the class imbalance problem present in the benchmark datasets. We modified the vanilla α-divergence loss and proposed a class-balancing approach to better handle the data imbalance problem. Second, proposed approach does not require any threshold to selectively use the *distribution alignment* loss, as used in (Gong et al., 2021). Instead of using a threshold, our approach uses a *confidence* scaling approach. Next, we propose the methodology of the proposed approach.

5.3 Methodology

5.3.1 Partially Supervised CZSL (pCZSL) Problem Statement

In pCZSL (Karthik et al., 2021) either the state or the object annotation is available during training phase for each image. Thus in pCZSL, the annotation for the i^{th} image may be represented as (s_i, u) or (u, o_i). Here u represents unavailability of the corresponding annotation during training. Therefore in pCZSL, the model does not have any knowledge about the compositional labels during training. Hence all possible state-object compositions, $\mathcal{C}$, irrespective of their feasibility are considered as possible outcome of the model. For a dataset with N_s number of states and N_o number of objects, the number of possible predictions in pCZSL is $N_s \times N_o$.

5.3.2 Proposed Approach

Our approach consists of Hierarchical Feature Extractor, Locality-Preserving Neighbourhood Aggregation and augmentation based *distribution alignment* strategy (see Fig. 5.3). Next, we discuss the different components of the proposed approach.

5.3.2.1 Hierarchical Feature Extractor

The dependency of state features on the context i.e. the object present in the image, results in intra-class variations in the features of the state. Hence the deep-learning algorithms struggle to learn unique and discriminative features of the state. Understanding the contextual

information in the image is critical for improving compositional generalisation ability of a model.

Vision Transformers (ViTs) (Dosovitskiy et al., 2020), due to patch level processing, have a global understanding of the contextual information in the image (Fu et al., 2020). Still, the computational cost of existing ViTs (Carion et al., 2020, Li et al., 2023a, Dosovitskiy et al., 2020, Yuan et al., 2021b) is quadratically proportional to the number of patches as generated from the input image. *Swin* transformer (Liu et al., 2021) utilises a window based self-attention strategy using the *W-MhSA* and *SW-MhSA* modules to reduce the computational cost to a linear function of the number of patches. We utilise a *swin* transformer based backbone in HFE module of our approach to better process the contextual dependency with linear computational cost.

A key challenge in the CZSL is that the features of the state depend on difference between the sizes of the object and background in an image. As shown in Fig. 1.3, for compositions *young elephant* and *old elephant*, the features of the state *young* and *old* are inferred from the size (or scale) of the *elephant* present in the image *Cross-Scale Consistency* (CrSC) problem.

In HFEs (Liu et al., 2021), the feature responses from intermediate layers are often sensitive to objects of different scales within an image. To process objects at different scales for the CrSC problem, the image features at different scales need to be utilised. However, existing approaches for the CZSL (Saini et al., 2022, Li et al., 2022, Karthik et al., 2022) primarily rely on features extracted only from the deepest layer of these image feature extractors. Unfortunately, the deeper layer features lack in scale-related information (Wang et al., 2022a).

To better process the multi-scale features from the intermediate layers, we propose Locality-Preserving Neighbourhood Aggregation (LoPNA) to be explained in the next section. The features from intermediate layers of HFE are passed as input to the LoPNA module.

Our HFE has four layers, each consisting of $2, 2, 6$ and 8 numbers of *W-MhSA* and *SW-MhSA* modules (Liu et al., 2021). The early two feature extractor blocks have very few number of *W-MhSA* and *SW-MhSA* modules. So, our multi-scale feature extraction only utilises features from second, third and fourth layers, represented as F^2, F^3 and F^4, respectively (see Fig. 5.3). Also deepest layer features from HFE, F^4 is passed to the Object Classifier for object classification. The output from the Object Classifier represents the object probabilities, p_o. Next, we describe the strategy to process the multi-scale features using LoPNA module.

5.3.2.2 Locality-Preserving Neighbourhood Aggregation

Inspired by the success of self-attention across diverse vision tasks (Alfasly et al., 2022, Zhang et al., 2022b), we propose to aggregate features from intermediate layers of the HFE using self-attention. However the self-attention as proposed in (Vaswani et al., 2017), is unsuitable for multi-scale feature aggregation due to the following reasons. First we need to upsample the channel dimension of the features from the deeper layers of HFE, which is not straightforward in vanilla self-attention approach. Next, we observe that vanilla self-attention is not well equipped to process the locally sensitive discriminative feature. To overcome the aforementioned two drawbacks of self-attention, we propose Locality Preserving Neighbourhood Aggregation (LoPNA) module, as described next.

Alongside the variation in the channel dimension, the features extracted from the intermediate layers of HFE have varying spatial dimensions too. The features from the shallower

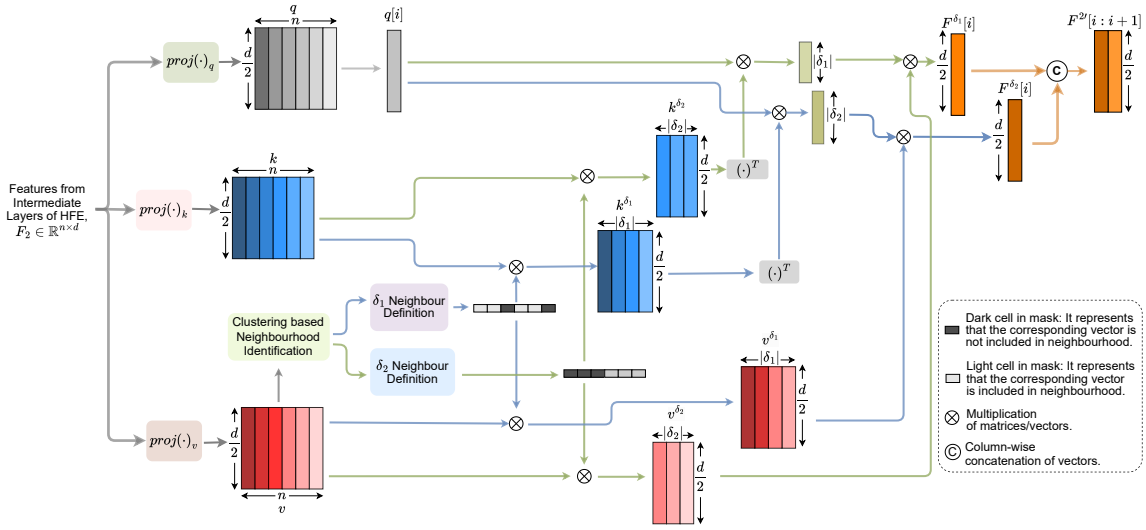


Figure 5.4: Proposed Locality Preserving Neighbourhood Aggregation approach.

layers (e.g. F^2) have larger spatial resolution in comparison to the features extracted from the deeper layers (e.g. F^3 and F^4). To aggregate features obtained from different layers, existing feature pyramid based approaches (Lin et al., 2017, Yu et al., 2020, Wang et al., 2019a, 2020) mainly resort to upsampling the features of lower resolution from the deeper layers to the same dimension as that of the features from shallower layers. Usually, interpolation based approaches are used for upsampling the features. However the interpolation based upsampling strategies are sub-optimal to upsample the semantically rich features from the deeper layers.

We use three projections functions $proj_k(\cdot)$, $proj_q(\cdot)$ and $proj_v(\cdot)$ with trainable parameters, to project the features from shallower layer, $F^2 \in \mathbb{R}^{n \times d}$ to a lower spatial dimensional space, same as the spatial dimension of the deeper layer features ($F^3, F^4 \in \mathbb{R}^{2n \times d/2}$). This approach is written as follows,

$$k = proj_k(F^2),\ q = proj_q(F^2) \text{ and } v = proj_v(F^2). \tag{5.1}$$

Here k, q and v represent the key, query and the value and k, q and $v \in \mathbb{R}^{n \times d/2}$. Also n and d represent the channel and spatial dimension of the intermediate layer feature, F^2, respectively. First we describe the self-attention strategy from (Vaswani et al., 2017) and then we discuss the motivation of our approach. The self-attention strategy in (Vaswani et al., 2017) next evaluates an output $z = \{z[1], z[2], \ldots, z[n]\}$, where $z[i] \in \mathbb{R}^{d/2}$. Each output $z[i]$ is computed as a weighted sum of the value vectors as follows,

$$z[i] = \sum_{j=1}^{n} \rho[i][j] v[j]. \tag{5.2}$$

The coefficient $\rho[i][j]$ in (5.2) is calculated using a soft-max function as follows,

$$\rho[i][j] = \frac{\exp(a[i][j])}{\sum_{l=1}^{n} \exp(a[i][l])}. \tag{5.3}$$

Here $a[i][j]$ is calculated using a dot-product attention between query and key as follows,

$$a[i][j] = \frac{q[i](k[j])^T}{\sqrt{d}}. \tag{5.4}$$

Following (Liutkus et al., 2021), we may interpret (5.2) as the classical weighted sum of the value vectors, where the weights are obtained from the attention matrix, a. Thus the final output of the self-attention is dependent on the effective addition of the value vectors. Inspired by this observation, we propose to perform a neighbourhood aggregation on the value vectors. We identify the clusters among the value vectors, to identify the neighbourhoods. To identify the clusters among the value vectors, we use the *fuzzy c-means* (FCM) algorithm (Dunn, 1973). The FCM algorithm (Dunn, 1973) is utilised for the following justifications. First, the discriminative local features in CZSL are often overlapping in nature. For example for the state *rotten* in *rotten apple*, the discriminative features like *textures* and *dis-coloured patches* often appear overlapped in an image.

In FCM, if there are f_c number of clusters and n value vectors, then the membership of value vector at index i to cluster r can be represented as μ_i^r, where $0 \leq \mu_i^r \leq 1$, $r \in \{1, 2, \ldots, f_c\}$ and $i \in \{1, 2, \ldots, n\}$. Thus in FCM a value vector may be a member of multiple clusters simultaneously depending on the μ_i^r. The flexibility to be present in multiple cluster in FCM allows for existence of overlapping clusters.

In our approach we use a scalar threshold, $c_\tau \in (0, 1)$. The i^{th} data point is assigned to a particular cluster if and only if $\mu_i^r \geq c_\tau$. Based on the threshold c_τ, any number of value vectors may be assigned to a cluster. After performing the clustering over the value vectors, we obtain the neighbourhood definitions, δ_1 and δ_2. Note that we utilise FCM to obtain the two clusters to process the global and local contexts as explained next. Here δ_1 and δ_2 represent the list of the indices of the value vectors assigned to clusters 1 and 2, respectively. Let the aggregated value be represented as v^{δ_1} and v^{δ_2} (see Fig. 5.4). Next, we gather the key vectors corresponding to the neighbourhood definitions, δ_1 and δ_2. Similarly the aggregated key vectors be represented as k^{δ_1} and k^{δ_2}. Following (5.4), we obtain the self-attention values as follows,

$$a^{\delta_1}[i][\delta_1[j]] = \frac{q[i](k[\delta_1[j]])^T}{\sqrt{d}}. \tag{5.5}$$

Next, following (5.3), we have,

$$\rho^{\delta_1}[i][\delta_1[j]] = \frac{\exp\big(a^{\delta_1}[i][\delta_1[j]]\big)}{\sum_{q \in |\delta_1|} \exp(a^{\delta_1}[i][\delta_1[q]])}. \tag{5.6}$$

Using the ρ^{δ_1} value from (5.6) we obtain the following,

$$z^{\delta_1}[i] = \sum_{j \in |\delta_1|} \rho^{\delta_1}[i][\delta_1[j]] v[\delta_1[j]]. \tag{5.7}$$

Next, the $z^{\delta_1}[i]$ values are concatenated to obtain the final cluster specific feature response, F^{δ_1} as follows,

$$F^{\delta_1} = concat(z^{\delta_1}[1], \ldots, z^{\delta_1}[n]). \tag{5.8}$$

Here $concat(\cdot)$ operation represents the usual concatenation operation. Using a similar ap-

proach as in (5.5), (5.6), (5.7) and (5.8) we obtain the cluster specific feature response, F^{δ_2}. Finally we concatenate the results of two neighbourhood aggregation as $F^{2'} = concat(F^{\delta_1}, F^{\delta_2})$. Next, we pass $F^{2'}$, F^3 and F^4 through average pooling operation, represented as $avgpool(\cdot)$ (see Fig. 5.3).

Besides, in existing FPN based approaches, after the features from the different layers are matched across the spatial and channel dimensions, obtained features are added element wise to get the aggregated multi-scale feature. We observe that the fine-grained visual features from deeper layers are often diluted in the element-wise addition of features (Wang et al., 2022a). Hence we propose to avoid element-wise addition. The final multi-scale feature in our approach is obtained by concatenation of the features from individual layers, i.e. the output from the $avgpool(\cdot)$ of the $F^{2'}$, F^3 and F^4 are concatenated and passed to the State Classifier for state feature recognition.

$$F_{\text{LoPNA}} = concat(avgpool(F^{2'}), avgpool(F^3), avgpool(F^4)). \tag{5.9}$$

Justification for using two clusters: Recent experimental observation and theoretical analysis reveal that the self-attention mechanism in Transformer Encoder acts as Low Pass Filter (LPF) on image features (Bai et al., 2022, Wang et al., 2022b). Thus cascading self-attention in a multi-layer Transformer Encoder setup results in loss of the high frequency features in an image. The high frequency components typically represent the local discriminative features. The loss of high frequency features i.e. the local details posses a major weakness for the self-attention mechanism in our LoPNA module.

So we propose to utilise the neighbourhood (e.g. δ_1) to compensate for the loss of the high frequency features. Inspired by the idea of adversarial training (Wang et al., 2019c, Zhang et al., 2019, Bai et al., 2022), we propose to incorporate high frequency noise in the final feature vector, F^{δ_1} from δ_1 neighbourhood. Features from the δ_2 neighbourhood, F^{δ_2} are not modified to preserve the usual low frequency component. Next we report the steps for the proposed approach.

First we generate a noisy feature vector, $\eta \in \mathbb{R}^{n \times d/2}$ with elements of η sampled from the interval $[-\epsilon, \epsilon]$. Here ϵ is a user-defined scalar. Next, the frequency domain representation of η, is obtained by using the Fast Fourier Transformation (Nussbaumer and Nussbaumer, 1982) and represented as $\text{FFT}(\eta)$. Next we pass $\text{FFT}(\eta)$ through a High Pass Filter (HPF) to remove the low frequency noise from η. The HPF, $\mathcal{F}_{\text{HPF}}(\cdot)$ is implemented utilisng the Gaussian Filter (Bu et al., 2023) as follows,

$$\mathcal{F}_{\text{HPF}}(u, v) = 1 - e^{-\frac{dist(u,v)}{2dist_0^2}}. \tag{5.10}$$

Here (u, v) represents the coordinates of points in the input feature map. The term $dist(u, v)$ represents the distance between the point (u, v) and the center point of the feature map. Here $dist_0$ represents the cut-off frequency of the HPF. The real part of the frequency domain representation of the noise is then passed through the aforementioned HPF to get the high frequency components in the noise, η. Next we perform the Inverse Fourier Transformation of the output from the HPF to get the high frequency component of the noise feature, η_{HF}. Next we add the high frequency noise F^{δ_1} as follows,

$$F^{\delta_1}_{\text{adv}} \leftarrow F^{\delta_1} + \eta_{\text{HF}}. \tag{5.11}$$

Next, $F^{2'}_{adv} = concat(F^{\delta_1}_{\text{adv}}, F^{\delta_2})$, and we also have the final adversarial features as follows,

$$F^{\text{adv}}_{\text{LoPNA}} = concat(avgpool(F^{2'}_{adv}), avgpool(F^3), avgpool(F^4)). \tag{5.12}$$

Since the LoPNA is specifically designed to process the multi-scale feature consistency, the output of the LoPNA, F_{LoPNA} (see (5.9)) is passed to the State Classifier. The output of the State Classifier is represented as state probabilities, p_s. Next, inspired by the idea of adversarial training (Wang et al., 2019c, Zhang et al., 2019, Bai et al., 2022) we also pass $F^{\delta_2}_{\text{adv}}$ (see (5.12)) to the State Classifier to obtain the state probabilities p^{adv}_s. p_s and p^{adv}_s are both used to supervise the network through cross entropy loss using state ground truth annotation (see Section 5.3.3 for description of the relevant adversarial loss). Next, we discuss the strategy for augmentation based *distribution alignment* loss evaluation.

5.3.2.3 Class-balanced and Confidence-scaled Distribution Alignment (CCDA) Loss

In pCZSL (Karthik et al., 2022), either the state or the object annotation is provided for each image in the training set. In absence of sufficient labels for training, deep learning models are prone to over-fitting to the small subset of labels that is provided during training (Huynh and Elhamifar, 2020). Here we propose an α-divergence based distribution alignment loss to effectively train proposed architecture for pCZSL.

Augmentation strategy and distribution alignment loss: To leverage the large amount of partially annotated data, we utilise two augmentation strategies, *strong* augmentation and *weak* augmentation. The *weak* augmentation is represented by $\mathcal{G}_{weak}(\cdot)$, $\mathcal{G}_{weak} : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^{3\times H\times W}$. The *weak* augmentation consists of *random flipping* and *center crop* perturbations. Similarly, we have used a *strong* augmentation $\mathcal{G}_{strong}(\cdot)$, $\mathcal{G}_{strong} : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^{3\times H\times W}$. $\mathcal{G}_{strong}(\cdot)$ consists of a set of augmentations with stronger perturbation to the image. Following RandAugment (Cubuk et al., 2020) *strong* augmentation includes *shearing transformation*, *alteration to brightness*, *contrast* and *sharpness* of the image.

Let us represent the two augmented images as I^{strong}_{aug} and I^{weak}_{aug}, where $I^{strong}_{aug} = \mathcal{G}_{strong}(I)$, $I^{weak}_{aug} = \mathcal{G}_{weak}(I)$. Along with I^{weak}_{aug}, we also simultaneously pass I^{strong}_{aug} to the proposed architecture and obtain the state and object probabilities corresponding to both the images. We represent the predictions from the *weakly* augmented image as p_s and p_o (for the sake of simplicity of symbols, we drop the superscript i from the p^i_s).

Similarly, for I^{strong}_{aug} corresponding state and object probabilities are represented as p^{aug}_s and p^{aug}_o. Next, we propose the following *distribution alignment* loss using Reny α-divergence (Póczos and Schneider, 2011, Rényi, 1961) as follows,

$$D_\alpha(p_s||p^{aug}_s) = \frac{1}{\alpha - 1} \log \sum_{m=1}^{N_s} \frac{p_s[m]^\alpha}{p^{aug}_s[m]^{\alpha-1}}. \tag{5.13}$$

Here α is a hyper-parameter, $\alpha \in (0, 1) \cup (1, \infty)$. The justification behind using the α-divergence based *distribution alignment* loss in (5.13) is as follows.

Evidently I^{strong}_{aug} incorporates significant perturbations. Hence the proposed network is less likely to learn effective features from I^{strong}_{aug}. Also the network is expected to be more confident in identifying the state from I^{weak}_{aug}. In (5.13) the state prediction corresponding to the *weakly* augmented image, p_s, works as supervision to the prediction corresponding to the *strongly* augmented image, p^{aug}_s. By minimizing the loss in (5.13) we enforce the network to

learn discriminative features even in the *strongly* perturbed images. Thus loss in (5.13) helps to train the model using the large amount of partially labelled data as available in pCZSL. Next, we discuss the class-balanced scaling approach.

Class-balanced scaling approach: We observe that some states and objects in the existing CZSL datasets have lower number of images in comparison to other states and objects as demonstrated in Fig. 5.2. Due to lower number of images, *minority classes* have lower contribution in the final loss during training. Thus the overall loss function is always dominated by the *majority classes*. The existing CZSL algorithms (Xu et al., 2021b, Panda et al., 2022, Nan et al., 2019) have not attempted to address this imbalance problem in the benchmark CZSL datasets.

A group of approaches are proposed to alleviate the problem of imbalanced datasets in deep learning. These algorithms can be classified into two groups. The first major line of approaches have considered re-sampling of additional data points from the *minority class* to compensate the effect of class imbalance (Ouyang et al., 2016, Wang et al., 2017, Huang et al., 2016). However re-sampling approaches have been observed to introduce large number of very similar samples, therefore slowing the training and makes the model prone to over-fitting. Thus we have worked using the other popular approach where the algorithm attempts to re-weight the loss function. The particular class re-weighting approach that we have followed is discussed next.

Limitation of the vanilla α-divergence: We analyse the effect of parameter α for the *distribution alignment* loss in (5.13). Let for a state m, $R = \frac{p_s[m]}{p_s^{aug}[m]}$. If $R > 1$, then with the increase in α value, the ratio $(R)^\alpha$ increases significantly. However, if $R < 1$ then with the increase in α ($\alpha > 1$), the ratio $(R)^\alpha$ decreases significantly. So as α changes, $D_\alpha(p_s||p_s^{aug})$ will be dominated by those $(R)^\alpha$ for which $R > 1$.

Instead of using a fixed α, we propose to use *class-specific* α for evaluation of the *distribution alignment* loss. The *class-specific* α values are inversely propositional to the number of images in the class under consideration. Thus for images corresponding to *minority classes*, corresponding *class-specific* α is high and subsequently the α-divergence based *distribution alignment* loss is also high. Thus the loss contribution of the *minority classes* are upsampled using the proposed approach. To obtain the class specific α, we use the *effective number*, discussed next.

Cui *et al.* (Cui et al., 2019) have introduced the concept of *effective number* of samples for each class in the dataset. They propose that all the samples belonging to a class are not equally useful for class-specific feature learning. They argue that as the number of samples in a class increases, the additional benefit of a newly added data point will be low. They propose to evaluate an *effective number* of samples for each state and object classes. The *effective number* (Cui et al., 2019) for each class is inversely propositional to the number of images belonging to that class, defined as follows.

Proposition 1: The *effective number* for class c, $c \in \mathcal{C}$ is $E_{n_c} = \frac{1-\beta^{n_c}}{1-\beta}$, where $\beta = \frac{N_t-1}{N_t}$ and n_c is the number of images in the class c.

From 2.3.1, N_t represent the number of image in the training set. Thus using the concept of *effective number* for each class, we evaluate the modified *class-specific* α for that class by dividing the existing α by the corresponding class's *effective number*. By modifying (5.13),

we obtain the *class-balanced distribution alignment* for state prediction loss as follows,

$$D_{\alpha_s}^{CDA}(p_s||p_s^{aug}) = \log \sum_{m=1}^{N_s} \frac{(p_s[m])^{\frac{\alpha_s}{E_{n_s}[m]}}}{(p_s^{aug}[m])^{\frac{\alpha_s}{E_{n_s}[m]}+1}}. \tag{5.14}$$

Here $E_{n_s}[m]$ represents the *effective number* for the state class m. We evaluate the relationship between the proposed Class Balanced Loss and the KL-divergence loss (Póczos and Schneider, 2011) in the following proposition.

Proposition 2: If KL-divergence between distributions p_s and p_s^{aug} is represented by $D_{KL}(p_s||p_s^{aug})$, then under the assumption of large dataset, it can be shown that,

$$D_{\alpha_s}^{CDA}(p_s||p_s^{aug}) \geq \frac{\alpha_s}{n_s N_s} D_{KL}(p_s||p_s^{aug}). \tag{5.15}$$

Here n_s represents the average number of images across the state classes in the dataset. The proof is provided in Appendix B, Section B.1.

As $\alpha_s \to (N_s n_s)$, the α_s value approximates the total number of images in the dataset, which is a large number for the CZSL datasets. In the α-divergence based approach, for $\alpha > 1$, the loss in (5.14) is dominated by the state classes which have $p_s[m] \gg p_s^{aug}[m]$, $m \in \{1, 2, \ldots, N_s\}$. A suitably trained model should have similar state and object predictions, irrespective of strong or weak augmentations. If the model predicts less values of the state probability for the strongly augmented image, due to the presence of the exponent α in (5.14), the corresponding image will have a higher contribution in the loss. For large value of α, if $p_s[m]$ is slightly greater than $p_s^{aug}[m]$ then also the factor R will be magnified to a large extent. Instead of a few classes, the loss contribution corresponding to most of the classes will be up-scaled due to presence of large α. The proposed CDA loss will be less selective and will tend towards the KL-divergence loss.

Confidence based scaling approach: As already explained, in the proposed approach we have utilised the prediction corresponding to the *weakly* augmented image to supervise the prediction of the *strongly* augmented image. However existing augmentation based approaches for semi-supervised learning (Gong et al., 2021, Sohn et al., 2020, Berthelot et al., 2019b,a) have observed that the model prediction corresponding to the *weak* augmentation of all the images are not suitable to supervise the prediction for the corresponding *strongly* augmented images. If the model is not sufficiently confident on the *weakly* augmented image then supervising the *strongly* augmented image may misguide the training of the model. Thus most of the semi-supervised learning approaches have used a threshold based selection strategy as explained next. In the threshold based strategy, only those images from the training set, for which the model's *confidence* (i.e. the highest class probability) on the *weakly* augmented image is above a predefined threshold, are used to supervise the prediction of the *strongly* augmented image using a *distribution alignment* loss. The highest class (state or object) probability among probabilities of all the classes (state or object) is reported as the *confidence* of a model.

The drawback of the existing threshold based approaches is that the performance of the approaches depend heavily on the values of the *confidence* threshold (Sohn et al., 2020). To circumvent this issue we propose a non-parametric approach as explained next. We evaluate the *confidence* scaling factor as the ratio of the maximum state *confidence* achieved by the model to the sum of the *confidences* in all other non-maximal state classes in the predicted

probability. The calculated *confidence* scaling factor is then multiplied with the *distribution alignment* loss component. Thus images which have achieved high *confidence* on the *weakly* augmented image will have high confidence scaling factor and correspondingly have higher contribution to the α-divergence based loss.

$$D_{\alpha_s}^{CCDA}(p_s||p_s^{aug}) = \log \sum_{m=1}^{N_s} \left(\frac{e^{\max(p_s)}}{e^{\max(p_s)} + e^{(\overline{p}_s)}} \right) \frac{(p_s[m])^{\frac{\alpha_s}{E_{n_s}[m]}}}{(p_s^{aug}[m])^{\frac{\alpha_s}{E_{n_s}[m]}+1}}. \tag{5.16}$$

Here $\max(p_s)$ is the *confidence* i.e. the highest state probability that the proposed model has achieved in deciding the state probability for the input image. Also $\overline{p}_s$ represents the sum of the probabilities of all the states excluding the state that has achieved the maximum probability. Similarly, for images without object annotation, we define the following loss,

$$D_{\alpha_o}^{CCDA}(p_o||p_o^{aug}) = \log \sum_{m=1}^{N_o} \left(\frac{e^{\max(p_o)}}{e^{\max(p_o)} + e^{(\overline{p}_o)}} \right) \frac{(p_o[m])^{\frac{\alpha_o}{E_{n_o}[m]}}}{(p_o^{aug}[m])^{\frac{\alpha_o}{E_{n_o}[m]}+1}}. \tag{5.17}$$

Here $E_{n_o}[m]$ represents the *effective number* for the object class m. Next, we describe the training and inference strategies for the proposed approach.

5.3.3 Training and Inference Strategy

Training Strategy: Recognising the state and object in CZSL is challenging mainly due to the entanglement between the visual features of the state and object. The features of a state in compositional image can be decomposed into two components, the *invariant state features* and the *context-dependent state features*. For the i^{th} image in the training set, $i \in \{1, 2, \ldots, N_t\}$, the *invariant state features*, obtained from LoPNA, are passed to the State Classifier to get the state probability, p_s^i. Next, the *context guided state features*, obtained from HFE are passed through Context Guided State Classifier (see Fig. 5.3) to obtain the context guided state probability $p_{s|o}^i$. The state probability, p_s^i is used to supervise the network through minimization of cross-entropy loss with respect to corresponding state ground truth, GT_s^i. The state cross-entropy loss, $\mathcal{L}_s$ is defined as,

$$\mathcal{L}_s = \sum_{i=1}^{N_t} \mathbb{1}_{GT_s^i \neq u} \mathcal{L}_{ce}(p_s^i, GT_s^i). \tag{5.18}$$

$\mathbb{1}_{GT_s \neq u}$ is the indicator random variable, which is only true when the state annotation is provided for the i^{th} image. $\mathcal{L}_{ce}(\cdot)$ represents the usual cross-entropy loss. We obtain the context guided state loss as follows,

$$\mathcal{L}_s^{cont} = \sum_{i=1}^{N_t} \mathbb{1}_{GT_s^i \neq u} \mathcal{L}_{ce}(p_{s|o}^i, GT_s^i). \tag{5.19}$$

Similarly, we can also split the object features into the *invariant object features* and the *context guided object features*. The *invariant object features* are obtained from the HFE. Corresponding invariant object probabilities p_o^i are obtained from the Object Classifier (see Fig. 5.3). The context dependent object probability $p_{o|s}^i$ are obtained from the Context

Guided Object Classifier. p_o^i is used to train the model through cross-entropy loss with respect to object ground truth, GT_o^i. The object cross-entropy loss, $\mathcal{L}_o$ is represented as,

$$\mathcal{L}_o = \sum_{i=1}^{N_t} \mathbb{1}_{GT_o^i \neq u} \mathcal{L}_{ce}(p_o^i, GT_o^i). \tag{5.20}$$

Next, we obtain the context guided object loss as follows,

$$\mathcal{L}_o^{cont} = \sum_{i=1}^{N_t} \mathbb{1}_{GT_o^i \neq u} \mathcal{L}_{ce}(p_{o|s}^i, GT_o^i). \tag{5.21}$$

Similarly, for images without state annotation, the state CCDA loss is evaluated using (5.16) as follows,

$$\mathcal{L}_s^{CCDA} = \sum_{i=1}^{N_t} \mathbb{1}_{GT_s^i == u} D_{\alpha_s}^{CCDA}(p_s^i || p_s^{aug\_i}). \tag{5.22}$$

Similarly, for images without object annotation, using (5.17) we define the object CCDA loss as follows,

$$\mathcal{L}_o^{CCDA} = \sum_{i=1}^{N_t} \mathbb{1}_{GT_o^i == u} D_{\alpha_o}^{CCDA}(p_o^i || p_o^{aug\_i}). \tag{5.23}$$

Next corresponding to the adversarial perturbation, we propose that the model should predict same state prediction in presence of high frequency adversarial perturbation. Thus the adversarial features from LoPNA $F_{\text{LoPNA}}^{\text{adv}}$ (see (5.12) are passed to the state classifier and we obtain the adversarial state predictions, p_s^{adv}. p_s^{adv} is used to train the model using the adversarial state loss as follows,

$$\mathcal{L}^{adv} = \sum_{i=1}^{N_t} \mathbb{1}_{GT_s^i \neq u} \mathcal{L}_{ce}(p_s^{adv\_i}, GT_s^i). \tag{5.24}$$

The final loss is computed using (5.18), (5.19), (5.20), (5.21), (5.22) (5.23) and (5.24) as follows,

$$\mathcal{L} = \lambda_1(\mathcal{L}_s + \mathcal{L}_o) + \lambda_2(\mathcal{L}_s^{cont} + \mathcal{L}_o^{cont}) + \lambda_3(\mathcal{L}_s^{CCDA} + \mathcal{L}_o^{CCDA}) + \lambda_4 \mathcal{L}^{adv}. \tag{5.25}$$

where λ_1, λ_2, λ_3 and λ_4 are scalar hyper-parameters. Next, we explain the inference strategy for the proposed algorithm.

Context Guided Inference Strategy: During inference, the state and object probabilities i.e. p_s and p_o are obtained from the state and object classifiers corresponding to *weakly* augmented images as shown in Fig. 5.3. Similarly we obtain the context guided state probability, $p_{s|o}$ from the Context Guided State Classifier. Next, we use a *filter* $f_k(\cdot)$ to selectively filter the top-k probabilities from $p_{s|o}$. $f_k(p_{s|o})$ is next integrated with the state probability, p_s to get the final state prediction as follows,

$$\hat{s} = \underset{s',o' \in \mathcal{C}}{\arg\max}\ \gamma_s p_{s'} + (1 - \gamma_s) f_k(p_{s'|o'}). \tag{5.26}$$

Dataset →	MIT-States				C-GQA				UT-Zappos50k			
Algorithm↓	*test AUC*	*seen*	*unseen*	*HM*	*test AUC*	*seen*	*unseen*	*HM*	*test AUC*	*seen*	*unseen*	*HM*
CGE	0.3	17.9	1.6	3.0	0.2	25.6	0.7	1.4	14.8	55.8	5.9	10.7
CompCos	0.4	10.8	2.0	3.6	0.4	24.3	0.4	0.7	16.6	52.4	4.1	7.6
Co-CGE	1.0	13.1	2.3	4.0	0.5	22.1	0.6	1.2	17.4	52.6	5.4	9.9
KG-SP	1.2	18.4	2.2	4.0	0.7	26.9	1.2	2.3	22.8	57.9	7.4	13.1
Pro-CC	-	14.1	2.9	4.8	-	24.1	1.1	2.0	-	55.1	8.1	14.1
Our Approach w ResNet-18	1.3	19.0	3.8	6.4	1.0	28.2	1.8	3.8	23.6	58.1	10.8	14.5
Our Approach	**1.4**	**23.0**	**6.5**	**7.9**	**1.3**	**36.1**	**4.2**	**6.5**	**24.9**	**63.5**	**14.3**	**16.1**

Table 5.1: Results on the pCZSL problem.

Similarly we obtain $p_{o|s}$ from Context Guided Object Classifier and the p_o from the HFE to obtain the final object prediction as follows,

$$\hat{o} = \arg\max_{s',o' \in \mathcal{C}} \gamma_o p_{o'} + (1-\gamma_o) f_k(p_{o'|s'}). \tag{5.27}$$

Here $\gamma_s, \gamma_o \in [0,1]$. Next, we present the experimental details including the results.

5.4 Experiments

5.4.1 Additional Dataset Details for pCZSL

We have already discussed about the datasets in Section 2.1. Here we only discuss the details regarding the pCZSL split. For OW-CZSL on we use the train-test split proposed by (Purushwalkam et al., 2019, Naeem et al., 2021). For pCZSL, we use the partial supervision spilt provided by (Karthik et al., 2022).

5.4.2 Compared Algorithms

On OW-CZSL, we compare our algorithm against nine other state-of-the-art algorithms, LE (Misra et al., 2017), AAO (Nagarajan and Grauman, 2018), TMN (Purushwalkam et al., 2019), SymNet (Li et al., 2020), CGE (Naeem et al., 2021), CompCos (Mancini et al., 2021), KG-SP (Karthik et al., 2022), DRA-Net (Li et al., 2023b), SAD-SP (Liu et al., 2023) and Pro-CC (Huo et al., 2024). On pCZSL we have compared our approach against CGE (Naeem et al., 2021), CompCos (Mancini et al., 2021), Co-CGE (Mancini et al., 2022), KG-SP (Karthik et al., 2022) and Pro-CC (Huo et al., 2024). The brief literature reviews of these algorithms have been done in Sections 2.2, 3.2 and 5.2. The results for all these methods are reproduced from the papers (Karthik et al., 2022, Mancini et al., 2021, Li et al., 2023b) and the corresponding open source implementations. Next, we report the implementation details of our work.

5.4.3 Implementation Details

We have used the pre-trained *swin-B* variant of *swin* transformer (Liu et al., 2021) as the proposed HFE. The backbone is fine-tuned during training stage of our approach. The State Classifier is implemented using an MLP with N_s number of output nodes. The Object Classifier is implemented using an MLP with N_o number of output nodes. We use Adam

Dataset →	MIT-States				C-GQA				UT-Zappos50k			
Algorithm↓	*test AUC*	*seen*	*unseen*	*HM*	*test AUC*	*seen*	*unseen*	*HM*	*test AUC*	*seen*	*unseen*	*HM*
TMN	0.1	12.6	0.9	1.2	NA	NA	NA	NA	8.4	55.9	18.1	21.7
LE	0.3	14.2	2.5	2.7	0.08	19.2	0.7	1.0	16.3	60.4	36.5	30.5
AAO	0.7	16.6	5.7	4.7	NA	NA	NA	NA	13.7	50.9	34.2	29.4
SymNet	0.8	21.4	7.0	5.8	0.4	26.7	2.2	3.3	18.5	53.3	44.6	34.5
CompCos	1.6	25.3	10.0	8.9	0.4	28.4	1.8	2.8	21.3	59.3	46.8	36.9
CGE	1.0	32.4	5.1	6.0	0.47	32.7	1.8	2.9	23.1	61.7	47.7	39.0
KG-SP	1.3	28.4	7.5	7.4	0.8	31.5	2.9	4.7	26.5	61.8	52.1	42.3
SAD-SP	1.4	29.1	7.6	7.8	1.0	31.0	3.9	5.9	28.4	63.1	54.7	44.0
DRA-Net	1.5	29.8	7.8	7.9	1.05	31.3	3.9	6.0	28.8	65.1	54.3	44.0
Pro-CC	1.6	27.6	10.6	7.8	0.54	29.0	2.6	3.8	23.6	62.2	48.0	39.9
Our Approach w ResNet-18	1.7	32.6	10.8	9.0	1.2	34.8	4.1	7.4	27.8	65.7	54.9	44.2
Our Approach	**1.9**	**33.8**	**11.1**	**9.4**	**1.5**	**37.8**	**4.8**	**8.2**	**29.9**	**66.8**	**56.0**	**45.1**

Table 5.2: Results on the OW-CZSL problem.

optimizer (Kingma and Ba, 2015) with learning rate 0.0005. Before using the *effective number* for classes, we normalise the *effective number* values using a softmax layer. The scalar weights for loss components in (5.25) have following values $\lambda_1 = 1.5$, $\lambda_2 = 1.75$, $\lambda_3 = 0.2$ and $\lambda_4 = 0.05$. For all the datasets, we train the model for 50 epochs. The batch size is taken as 32. The values of the parameter (α_s, α_o) for MIT-States, C-GQA and UT-Zappos50k are set at 8. The parameters c_τ is set to value 0.5. The parameter γ_s and γ_o both are set at value 0.95. The model is implemented using PyTorch (Paszke et al., 2019) on a NVIDIA RTX 4090 GPU with 24 GB memory, CUDA 11.7 using Python 3.8 on a PC with 256 GB RAM and Linux operating system. Next, we present the results of our approach.

5.4.4 Results

Results of the proposed approach as well as existing state-of-the-art algorithms on pCZSL are shown in Table 5.1. The results on the OW-CZSL are reported in Table 5.2. Amongst pCZSL and OW-CZSL, pCZSL is more challenging problem due to partial annotation in the training set. Evidently as shown in Table 5.1, proposed approach achieves the state-of-the-art results on MIT-States (Isola et al., 2015), C-GQA (Naeem et al., 2021) and UT-Zappos50k (Yu and Grauman, 2017, 2014) by sufficient margin. On *test AUC* metric, an improvement of minimum 0.2% is achieved across all three datasets. On MIT-States dataset, the proposed approach has reported 8.9% improvement in *seen* state recognition accuracy on pCZSL. Similarly, we have achieved 3.6% and 2.7% improvement in accuracy in the *unseen* and *HM* metrics, respectively.

All the algorithms, specifically on pCZSL problem, achieves high accuracy on *seen* metric and poor results on the *unseen* metric. Evidently, this observation implies the significant amount over-fitting of the models on the *seen* classes. Proposed approach has achieved the highest accuracy on *unseen* metric. It is evident that the proposed approach is less prone to over-fitting than the existing algorithms. This can be attributed to the fact that the CCDA loss has helped the network to prevent the over-fitting to *seen* classes in presence of partial annotation.

In general, algorithms designed for CZSL problem achieve inferior performance in OW-

CZSL, especially in large datasets (e.g. C-GQA), due to the large size of the state-object compositional space. In the OW-CZSL problem, proposed approach either outperforms or reports competitive results with respect to the state-of-the-art-approaches. Specifically on the *test AUC* metric on OW-CZSL, proposed approach reports 0.3%, 0.5% and 1.1% improvement in *test AUC* for MIT-States, C-GQA and UT-Zappos50k, respectively. Similar improvement is reported over the *seen*, *unseen* and the *HM* metrics on all three datasets.

In all the results in Table 5.1 the closest competitor to our approach are the KG-SP (Karthik et al., 2022) and Pro-CC (Huo et al., 2024). KG-SP utilised external knowledge (ConceptNet (Speer et al., 2017)) to evaluate the feasibility of unseen composition. Our approach, on the contrary does not require any external knowledge during training or inference stages. Pro-CC (Huo et al., 2024) proposed two classifiers, one each for the state and the object. Next, the intermediate response from the Object Classifier is added with the intermediate response in the State Classifier, in an attempt to process the context dependency between the state and object features. We argue that the context dependency between state and object visual features can be more effectively processed through sharing the feature response from intermediate layers of the image feature extractors. The classifiers in Pro-CC (Huo et al., 2024) are implemented using the MLPs and are less capable to process the local structures in the image features.

Effect of choice of the the backbone in our approach: Our approach uses *swin* transformer (Liu et al., 2021) based backbone in the HFE. The existing CZSL approaches (Li et al., 2020, Xu et al., 2021b, Li et al., 2023b) have used ResNet-18 (He et al., 2016) based backbone. For a fair comparison of existing approaches with our approach, we trained our model with ResNet-18 backbone and reported the results in Tables 5.1 and 5.2.

The results of our approach, when trained with ResNet-18 based HFE, report at least 0.1% improvement over the other approaches, in *test AUC* across all three datasets in pCZSL and OW-CZSL. Thus, with the ResNet-18 based HFE, our approach is able to achieve superior result across pCZSL and OW-CZSL evaluation settings. The aforementioned results justify the backbone agnostic effectiveness of our approach.

However as shown in Tables 5.1 and 5.2, with the *swin* transformer based HFE, better results over ResNet-18 based HFE are achieved. The better result of *swin* transformer over the ResNet-18 based HFE is due to the fact that *swin* transformer is able to better process context dependency between state and objects. Due to small field of view of operations like convolution and pooling, the ResNet-18 based HFE is unable to process long-range context dependency. Subsequently, the ResNet-18 based HFE is unable to process the variability in state features due to context dependency between visual features of the constituent primitives. Next, we report the ablation study.

5.4.5 Ablation Study

5.4.5.1 Ablation Study on Loss components

Here the effect of different loss components (see (5.25)) on the recognition results of the proposed approach is evaluated. The results in Table 5.3 represents the *val AUC* obtained after training the network using different loss combinations on the C-GQA and MIT-States. All possible combinations of loss components are not feasible to train the network. For example, it is not possible to train the model excluding state and object cross-entropy loss, $\mathcal{L}_s$ and $\mathcal{L}_o$. Hence this combination is excluded from the current discussion.

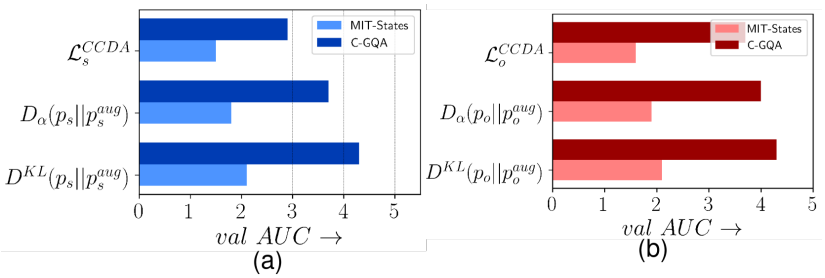


Figure 5.5: The result of the ablation study on the effectiveness of the distribution alignment loss, $\mathcal{L}_s^{CCDA}$.

Loss Component	Combination of loss components used during training							
$\mathcal{L}_s$	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{L}_o$	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{L}_s^{CCDA}$	×	×	✓	✓	✓	✓	✓	✓
$\mathcal{L}_o^{CCDA}$	×	✓	×	✓	✓	✓	✓	✓
$\mathcal{L}_s^{cont}$	×	×	×	×	×	✓	✓	✓
$\mathcal{L}_o^{cont}$	×	×	×	×	✓	×	✓	✓
$\mathcal{L}^{adv}$	✓	✓	✓	✓	✓	✓	×	✓
val AUC on MIT-States	1.3	1.4	1.6	1.7	1.8	1.9	2.0	**2.1**
val AUC on C-GQA	2.9	3.1	3.4	3.5	3.7	4.0	4.2	**4.3**

Table 5.3: The analysis of the loss components on pCZSL.

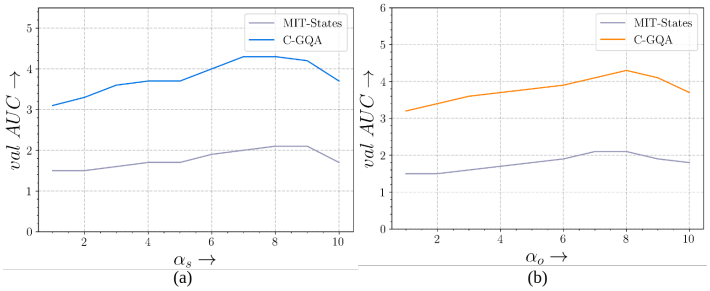


Figure 5.6: Ablation over the α_s and α_o parameters as used in (5.16) and (5.17).

The results in the cells at the last row-last column and last row-second column of Table 5.3 show that inclusion of both $\mathcal{L}_s^{CCDA}$ and $\mathcal{L}_o^{CCDA}$ improves the AUC on the validation set by 0.8% for the C-GQA dataset. Similar improvement of 0.4% is achieved on the *val* AUC of MIT-States.

Among $\mathcal{L}_s^{CCDA}$ and $\mathcal{L}_o^{CCDA}$, $\mathcal{L}_s^{CCDA}$ is observed to have greater contribution on the model's performance. Comparing the results in the last two rows of second and fourth columns of Table 5.3, $\mathcal{L}_s^{CCDA}$ improvements of 0.3% and 0.5% on *val* AUC can be observed on MIT-States and C-GQA respectively. On the other hand, observing the last two rows of second and third columns of Table 5.3, $\mathcal{L}_o^{CCDA}$ provides improvements of 0.1% and 0.2% on *val* AUC score for MIT-States and C-GQA, respectively.

The stronger effect of $\mathcal{L}_s^{CCDA}$ over $\mathcal{L}_o^{CCDA}$ can be justified from the fact that the CrSC and ContDp problems are more dominant for state features than the object features. Evidently,

$\mathcal{L}_s^{CCDA}$ provides higher improvement in final *val AUC* compared to $\mathcal{L}_o^{CCDA}$. Also $\mathcal{L}_{adv}$ reports similar improvement in *val AUC* on both MIT-States and C-GQA.

5.4.5.2 Analysis of CCDA Loss

Analysis of the class-balancing approach: To evaluate the effectiveness of the proposed class-balancing approach, we train the model with three different approaches, as described next. First approach utilises the KL-divergence (Póczos and Schneider, 2011) based distribution alignment (DA) loss as follows,

$$D^{KL}(p_s||p_s^{aug}) = \sum_{m=1}^{N_s} p_s[m] \log \frac{p_s[m]}{p_s^{aug}[m]}. \tag{5.28}$$

In the second approach, we train the proposed model with the α-divergence based DA loss as follows,

$$D_\alpha(p_s||p_s^{aug}) = \log \sum_{m=1}^{N_s} \frac{(p_s[m])^\alpha}{(p_s^{aug}[m])^{\alpha+1}}. \tag{5.29}$$

To solely evaluate the effectiveness of the α-divergence, in (5.29) the α values are kept constant irrespective of the classes.

The last approach in the experiment is the proposed class-balanced distribution alignment loss from (5.14). The results of the three aforementioned approaches are reported in Fig. 5.5. In 5.5, the KL-divergence based DA loss reports lower *val AUC* over both the fixed-α (5.29) and the class-balanced α based DA losses (5.14). The justifications for the results are discussed next.

In the α-divergence based approach, for $\alpha > 1$, the DA loss (see (5.29)) is dominated by the state classes which have $p_s[m] \gg p_s^{aug}[m]$, $m \in \{1, 2, \ldots, N_s\}$. A suitably trained model should have similar state and object predictions, irrespective of strong or weak augmentations. If the model predicts lower values of the state probability for the strongly augmented image, due to the presence of the exponent α in (5.29), the corresponding image will have a higher contribution in the loss. Subsequently the predictions with less values for probabilities for the strongly augmented images is penalized more. In (5.28), as no exponent like α is used, the KL-divergence based loss is less effective in extracting discriminating features across the strongly augmented images. The effectiveness of the class-balanced α-divergence approach over the fixed α based divergence approach is discussed next.

As shown in Fig. 5.2, CZSL datasets have imbalanced distribution of number of images in classes. So the contribution of the *minority classes* in the final loss term is low. Thus the model will struggle to effectively learn discriminative features for the *minority classes*. In the proposed class-balanced approach, we use an α-divergence based loss with the α values replaced by class specific exponents. The exponents used in the class-balanced approach are inversely proportional to the number of images in each class. Thus the resulting exponent in the class-balanced α-divergence is higher for *minority classes* and lower for *majority classes*. Subsequently, *minority classes* have more contribution in the final loss. So, the proposed $\mathcal{L}_{CCDA}$ is able to learn discriminative features over fixed α-divergence based distribution alignment loss.

Ablation Study of the Confidence-scaling Approach: To leverage the large amount of

unlabeled images in pCZSL, the prediction from weakly augmented input image, I_{aug}^{weak} are used to supervise the predictions from strongly augmented input image, I_{aug}^{strong}. However if the predictions from I_{aug}^{weak} are inaccurate, then the aforementioned approach will wrongly supervise the predictions from I_{aug}^{strong}.

In existing approaches (Sohn et al., 2020, Yang et al., 2023), if the model for I_{aug}^{weak} obtains *confidence* at least greater than the threshold τ then only predictions from I_{aug}^{weak} are used to supervise the predictions from the I_{aug}^{strong}. Note that for state prediction, the maximum state probability (or maximum object probability for object prediction) as predicted by the model is interpreted as the *confidence* of the model for input image. Other semi-supervised learning approaches like (Xu et al., 2021a, Wang et al., 2022c) have also utilised parametric approaches.

We propose a non-parametric approach to adaptively prevent the less confident predictions from guiding the predictions of the model from I_{aug}^{strong}. In our approach, if $max(p_s)$ increases, then the *confidence* scaling factor i.e. $\frac{e^{\max(p_s)}}{e^{\max(p_s)}+e^{(\overline{p}_s)}}$ also increases. Subsequently the images for which the model is more confident, will have higher contribution in the distribution alignment loss. By inclusion of the *confidence* scaling factor, we achieve an improvement of 0.2% and 0.3% on MIT-States and C-GQA datasets, respectively.

Ablation study of the multi-scale feature extraction: The LoPNA module in this chapter utilises the multi-scale features from the intermediate layers of the HFE. The output features from the LoPNA module, F_{LoPNA} is utilised to predict the state probability p_s and the context guided object probability $p_{o|s}$. The corresponding losses to supervise p_s and $p_{o|s}$ are $\mathcal{L}_s$ and $\mathcal{L}_o^{cont}$, respectively (see (5.18) and (5.21)). So to evaluate the effectiveness of the LoPNA module, we train the model by excluding $\mathcal{L}_s$ and $\mathcal{L}_o^{cont}$. In this experiment we obtain the *val AUC* on MIT-States and C-GQA datasets as 1.8 and 4.0, respectively. Thus the multi-scale feature extraction through the LoPNA module helps to achieve an improvement in *val AUC* by 0.3% on MIT-States and C-GQA datasets, respectively. Next, the improvement in *val AUC* due to the inclusion of the multi-scale features is discussed.

In pCZSL, estimating the scale of the objects in an image is useful (see images of *old elephant* and *young elephant* in Fig. 1.3). The features from different intermediate layers of the HFE are sensitive to objects of different scales as present in the input image (Wang et al., 2022a). Proposed LoPNA module attempts to address the aforementioned scale variation by processing the multi-scale features through neighbourhood aggregation.

5.4.5.3 Effect of Variation of the Parameters of CCDA Loss

The variations on the *val AUC* w.r.t. α_s and α_o parameters for MIT-States and C-GQA datasets are shown in Fig. 5.6. The *val AUC* w.r.t. α curves are steeper for the C-GQA dataset than the MIT-States dataset. This can be justified as follows. The value of α controls the effect of the CCDA loss. The CCDA loss is predominantly designed to alleviate the effect of data imbalance of the CZSL datasets. However as shown in Fig. 5.2, the imbalance issue is more dominant in the C-GQA dataset over the MIT-States dataset. Thus the sub-optimum value of the parameter α has higher effect over the *val AUC* of C-GQA dataset in comparison to MIT-States dataset. Next, we summarise the chapter.

5.5 Summary

In this chapter, a novel approach for the pCZSL and OW-CZSL problems is propsed. Proposed approach attempts to capture long-range context dependency in state-object compositional image using a Hierarchical Feature Extractor. Besides, our approach attempts to process cross-scale feature consistency using a Locality-Preserving Neighbourhood Aggregation module. Finally, specifically for the pCZSL problem, we have proposed a class-balanced confidence-scaled *distribution alignment* strategy.

In the next chapter, we shall explore the effectiveness of Vision Language Model based approach in context of CZSL. We shall attempt to design a VLM based approach to effectively address the context-dependency challenge in CZSL.

Chapter 6

Prompt-Driven Multi-Branch Disentanglement Network

6.1 Introduction

Prior CZSL approaches (Nan et al., 2019, Yang et al., 2020, Saini et al., 2022, Panda et al., 2023, 2022) project the image features of the input image and the text features (Mikolov et al., 2013, Bojanowski et al., 2017, Pennington et al., 2014) of the state and object labels to a shared embedding space. During inference, the compatibility between the image features and the text features in the shared embedding space, determines the state-object class label for the input image. However, the aforementioned approaches suffer from the drawback that the text feature extractors and the image feature extractors (He et al., 2016, Vaswani et al., 2017) are separately trained. The overall performance of the algorithm depends significantly on the effectiveness of the projection strategy. To circumvent this issue, in this chapter, we adapt a Vision-Language Model (VLM) (Radford et al., 2021) for the CZSL problem. VLMs (Radford et al., 2021, Jia et al., 2021) usually have a *text feature extractor* and an *image feature extractor*. The *text* and *image feature extractors* in VLMs are pre-trained on large-scale of paired text-image data.[1]

Inspired by the aforementioned advantage, a number of CZSL approaches were recently proposed (Nayak et al., 2022, Lu et al., 2023, Wang et al., 2023a, Huang et al., 2023) utilising the Contrastive Learning Image Pre-training (CLIP) (Radford et al., 2021) based VLM model. These approaches have not effectively addressed the four major requirements of the CZSL, viz the *disentanglement of constituent features* (as discussed in Chapter 2), *estimating the contextual dependency among the state and the object features* (as discussed in Chapter 3), *lack of feature diversity of text features from deterministic prompt template* and *scale-aware feature extraction*. Next, we briefly describe the challenges and our approaches in an attempt to solve them.

i) Challenge of disentanglement: As discussed in Chapter 2, the visual features of a compositional image mainly consist of three components, first the object agnostic state features, second the state agnostic object features and third, the new features which arise due to com-

[1]A part of the work in this chapter is under review as follows,
Aditya Panda and Dipti Prasad Mukherjee, "Prompt-Driven Multi-Branch Disentanglement Network for Compositional Zero-Shot Learning", communicated to IEEE Transactions on Pattern and Machine Intelligence, 2024.

position between state-object (Hao et al., 2023). Isolating the first and second components of features lead a CZSL algorithm to better generalise to unseen compositions. Lastly, isolating the third component of features is effective in recognising images from the seen compositions (the test set in CZSL comprises images from seen as well as unseen compositions).

The *text feature extractor* in CLIP (Radford et al., 2021) takes as input a prompt corresponding to the possible labels of the input image. Most of the existing approaches (Nayak et al., 2022, Xu et al., 2022, Lu et al., 2023) use a state-object joint prompt i.e. `a photo of <state><object>` (here `<state>` and `<object>` represent the state and object labels, respectively). The textual features (corresponding to a prompt), as extracted from the *text feature extractor* leverage the knowledge about the particular class from the pre-trained information in CLIP. We observe that only one single prompt is not effective to disentangle all three components of features in a compositional image (Hao et al., 2023). Without effective disentanglement of constituent features, CZSL methods are prone to over-fit to a small subset of seen compositions. Hence, we propose three distinct *text feature extractors* in our approach. These three branches take as input three different prompts, i.e. '`a photo of <state>`', '`a photo of <object>`' and '`a photo of <state><object>`'.

ii) Estimating contextual dependency between features of state and object: Due to the large-scale pre-training of CLIP (Radford et al., 2021) and scarcity of comparable training data for downstream tasks like CZSL, it is ineffective to fine-tune the CLIP (Radford et al., 2021) model for CZSL. Precisely, the compositional class recognition accuracy of CLIP decreases (Zhang et al., 2022c, Nayak et al., 2022) on fine-tuning pre-trained CLIP using CZSL datasets. Thus to adapt CLIP to downstream tasks without fine-tuning, different approaches have been proposed over the years (Zhou et al., 2022b,a, Khattak et al., 2023). Khattak et al. (2023) propose MaPLe, where the authors propose to insert learnable tokens in the intermediate layers of the *text feature extractor* and *image feature extractor* of CLIP. Inspired by MaPLe we also incorporate learnable tokens in the intermediate layers of *text* and *image feature extractors*.

A major challenge in CZSL is that within different images of same state-object composition, the visual features of state may vary. It can be observed that the text features can not accommodate the variation in the image features. Hence, we propose to share the intermediate layer features from the *image feature extractor* to the *text feature extractor*. We propose three Knowledge Coupling Modules (KCMs), namely the State Knowledge Coupling Module (SKCM), Object Knowledge Coupling Module (OKCM) and Joint Knowledge Coupling Module (JKCM) (see Fig. 6.1). In contrary, MaPLe (Khattak et al., 2023) only shares features from *text feature extractor* to the *image feature extractor* which is ineffective for processing the intra-class variation in image features.

iii) Lack of feature diversity of text features from deterministic prompt template: Extraction of deterministic text features from a fixed prompt template for each class (Khattak et al., 2023, Zhou et al., 2022a,b) inherently undermines the ability of the text features to adapt to variability of the image features. The lack of diversity in text features is more significant if the test set includes images of unseen classes. Although a number of VLM based approaches were proposed over the years (Khattak et al., 2023, Zhou et al., 2022b,a), the aforementioned issue is not still effectively addressed (Derakhshani et al., 2023).

The text features obtained from the state-object joint prompt include features about context in state-object composition (Huang et al., 2023). On the other hand, the text features extracted from the individual state prompt and the object prompt often lack in the context

related information. Thus, we intend to sample *state context token* and *object context token* from the state-object joint prompt space. Next, the sampled *state* and *object context tokens* are added with the tokens obtained from the existing deterministic state and object prompts and subsequently passed to the *text feature extractor*. The uncertainty involved in the random sampling process includes diversity in the *state* and *object context tokens*.

Here we propose an *encoder-decoder* based variational inference framework with mutual information based regularization. The variational inference framework helps to estimate the distribution of the token embeddings of the state labels and the object labels, from the joint state-object token embedding. Next, from the obtained distribution of the state and the object token embedding, we sample *state* and *object context tokens*. To ensure effective disentanglement in estimating the state and the object token embedding distributions, we use a mutual information based regularization approach. To the best of our knowledge, we are first to use a variational inference framework for prompt tuning in CZSL.

iv) Scale-aware feature extraction challenge: In CZSL, effective identification of the discriminative features of state and object often depends on the scale of the object in the image. For example, in the images of two compositions, *young bear* and *old bear* (see Fig. 1.3) the identification of the states *young* and *old* may depend on recognising the scale (or size) of the object *bear* in the image. However, existing CLIP based CZSL approaches (Nayak et al., 2022, Lu et al., 2023, Wang et al., 2023a, Huang et al., 2023) are designed to extract the single-scale global features from the input image. On the contrary, we propose to extract multi-scale features from the input image. Next, we summarise the contributions of our work in this chapter.

1. A CLIP (Radford et al., 2021) based approach with three distinct prompts is proposed for the CZSL problem to better disentangle the state and object features.
2. Three Knowledge Coupling Modules are proposed for better processing of the contextual dependency between state and object features.
3. To improve the generalization ability of the text features, a variational inference based framework with mutual information based regularization is proposed.
4. To extract scale-aware features, a multi-scale feature aggregation approach is proposed.
5. Results of the proposed approach outperform other conventional as well as VLM based CZSL approaches.

We summarise the recent works relevant to our approach in Section 6.2. Next, we discuss the proposed approach in Section 6.3. We present the results and experimental setup in Section 6.4 followed by summary in Section 6.5.

6.2 Related Works

Conventional CZSL approaches were already discussed in Sections 2.2, 3.2 and 4.2. Here we additionally discuss the VLM based CZSL approaches.

VLM based CZSL approaches: With the recent advent of VLMs, the state-of-the-art CZSL results are mostly achieved by approaches using VLMs. Nayak et al. (2022) first

attempt to adapt CLIP (Radford et al., 2021) for CZSL by replacing the class token (<cls>) in prompts with state-object label. Xu *et al.* (Xu et al., 2022) create a fully learnable *soft-prompt* including the prefix, state, and object text labels. DFSP (Lu et al., 2023) proposes a cross-modal fusion module to integrate image features with disentangled text features. The improvements of our approach over the aforementioned approaches (Nayak et al., 2022, Xu et al., 2022, Lu et al., 2023) are two-fold. First we use three independent prompts and corresponding *text feature extractors* for effective disentanglement of state and object features from the features of the compositional image. Besides we incorporate cross-modal knowledge sharing through the proposed Knowledge Coupling Modules (KCMs) to better process the contextual dependency between state-object features. Three recently proposed works (Huang et al., 2023, Wang et al., 2023a, Li et al., 2024), alike our approach, also use multiple prompts for CZSL. However aforementioned three CZSL approaches suffer from different weaknesses over our approach. The 'cross-modal traction' (CMT) module in Troika (Huang et al., 2023) uses a scaled dot-product attention (Vaswani et al., 2017) of text and image features. The results of the scaled dot-product are added with the text features (see (18) in Troika (Huang et al., 2023)). On the contrary, in our approach using KCMs, we share knowledge from the *image feature extractor* to guide the text features. Hence, the approach in Troika (Huang et al., 2023) is less effective for CZSL problem than our approach in processing the intra-class feature variation in CZSL. Besides proposed multi-scale feature extraction approach in this chapter helps to better solve scale-aware feature aggregation problem. The work in (Li et al., 2024) attempts to estimate text feature based feasibility of unseen state-object compositions to weed out infeasible predictions. However, the works in (Wang et al., 2023a, Li et al., 2024) are less effective to process intra-class variation in image features than our approach due to absence of knowledge sharing between the *text* and *image feature extractors*.

Deterministic prompt learning in VLMs: Large-scale pre-trained Vision-Language Models (VLMs) (Radford et al., 2021, Jia et al., 2021) have recently shown significant progress for the image classification problem. To adapt VLMs to downstream tasks, *prompt engineering* approach is proposed. Early *prompt engineering* techniques use pre-defined fixed text prompts (known as *hard-prompts*) (Petroni et al., 2019, Scao and Rush, 2021, Wei et al., 2022), as input to the *text feature extractor*. Although manually crafted *hard-prompts* may yield impressive results, the process of identifying the optimal *hard-prompts* is often laborious and computationally expensive (Zhou et al., 2022b,a). To overcome this drawback, Zhou *et al.* (Zhou et al., 2022b) propose *soft-prompt* approach. In *soft-prompt* approach (Zhou et al., 2022b), a part of the prompt is learned (by back-propagating the loss gradient) without fine-tuning the entire CLIP model. In *soft-prompt*, the input to the *text feature extractor* is usually of the form '[v1][v2][v3]<cls>', where [v1], [v2] and [v3] are learnable vectors and '<cls>' represents the possible class labels of the input image. Recently, Khattak *et al.* (Khattak et al., 2023) propose MaPLe, where the authors introduce the idea of introducing learnable tokens in the intermediate stages of the *image* and *text feature extractors*. Inserting learnable tokens in the intermediate stages of the *image* and *text feature extractors* help to adapt the frozen CLIP for the downstream tasks. Inspired by (Khattak et al., 2023), we also incorporate learnable tokens inside the *image* and *text feature extractors* in the proposed approach. In CZSL the visual features suffer from inter-class variation as well as intra-class variation due to contextual dependency between visual features of state and object. To better process the intra-class feature dependency, our approach shares the intermediate information not only from *text feature extractor* to *image feature extractor* but also from

image to *text feature extractor*. Besides to better process the scale-aware features, we also propose a multi-scale feature aggregation approach. A major improvement over the existing VLM based approaches (including MaPLe (Khattak et al., 2023)) is that our approach uses a novel variational-inference based prompt diversity increment approach.

Probabilistic prompt learning in VLMs: Most of the existing approaches (Zhou et al., 2022a,b, Khattak et al., 2023), use a specific prompt template for each class. Lu *et al.* (Lu et al., 2022) propose the first probabilistic prompt learning approach, where instead of a single deterministic prompt, a Gaussian ensemble of prompt templates are used. Next, the chapter in (Derakhshani et al., 2023) uses Bayesian inference to approximate the posterior prompt distribution using a Gaussian prior. However, our approach is different from the existing approaches due to the following reasons. First, to specifically suit the purpose of CZSL, we use dual encoder setup based Variational Auto-Encoder framework. Besides we have also used a novel information theoretic regularization approach for effective disentanglement of state and object token embedding. The methodology of the proposed approach is reported next.

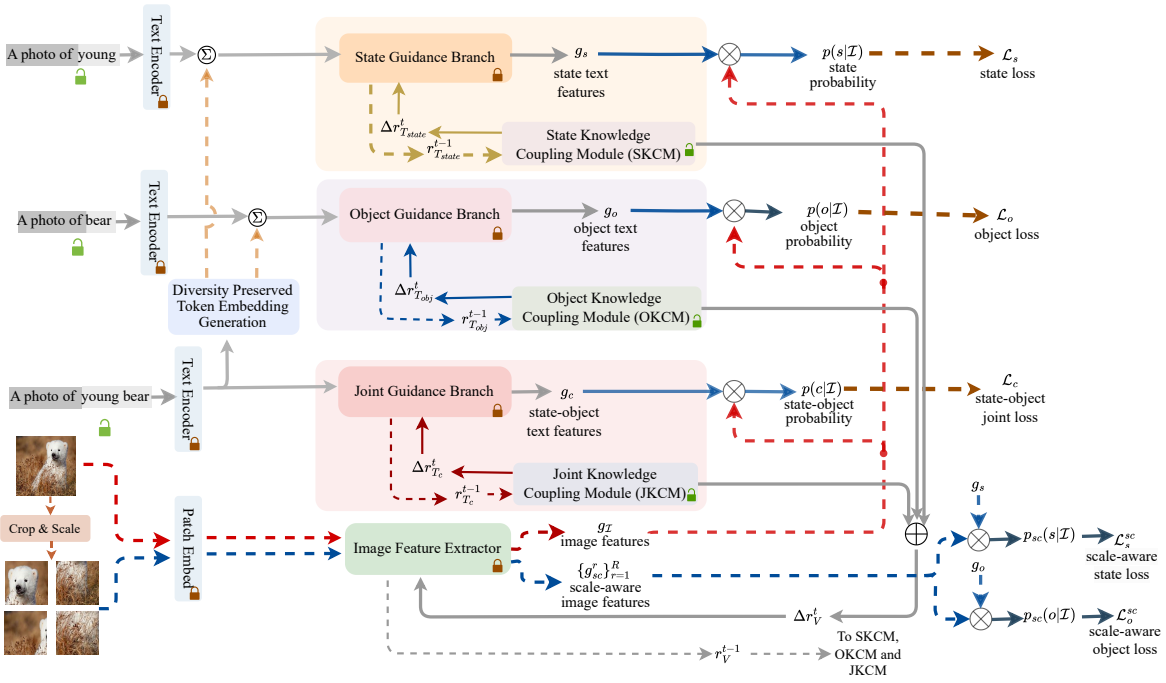


Figure 6.1: Proposed Prompt Guided Multi-Branch Disentanglement Network. The *closed-lock* sign beside a module represents that the parameters of the corresponding module are fixed during training. The *open-lock* sign represents that the parameters of corresponding module are updated during training.

6.3 Methodology

Before reporting the proposed approach, brief review the CLIP architecture.

Revisiting CLIP Basics: CLIP consists of two disjoint backbones, the *image feature ex-*

tractor and the *text feature extractor*. Following prior CLIP based CZSL literature (Nayak et al., 2022, Lu et al., 2023, Huang et al., 2023), Vision Transformer (ViT) backbone (Vaswani et al., 2017) is used in our approach.

Let us represent the *text feature extractor* as $\mathcal{F}_T(\cdot)$ and the *image feature extractor* as $\mathcal{F}_I(\cdot)$. The *text feature extractor* consists of L_T number of layers. Each layer of the *text feature extractor* consists of a *transformer encoder* (Vaswani et al., 2017). The *transformer encoder* in layer t, $t \in \{1, 2, \ldots, L_T\}$ is represented as $\mathcal{F}_T^t(\cdot)$. The *transformer encoder* generates feature representation for input prompt by first tokenising the words in the prompt and then obtaining the word embedding of the tokens. Let the number of tokens corresponding to the input prompt be k and each token be converted into an embedding of d dimensions. So we represent the resultant word embedding (which is also the input to the first layer of the *text feature extractor*) as follows, $f^0 = \{f_1^0, f_2^0, \ldots, f_k^0\}$, where $f_i^0 \in \mathbb{R}^d$. The *end-of-sentence* token is represented by f_{eos}^0 and appended at the end of the text tokens. Here f^t is the input to layer t of the *text feature extractor* as shown below,

$$[f^t, f_{eos}^t] = \mathcal{F}_T^t(f^{t-1}, f_{eos}^{t-1}), t \in \{1, 2, \ldots, L_T\}. \tag{6.1}$$

The $f_{eos}^{L_T}$ token, as obtained from the *transformer encoder* in the layer L_T, is considered as the final output of the *text feature extractor*. Next, using a projection function, $\phi_T(\cdot)$, the obtained text features are projected to a shared embedding space as follows,

$$g_T = \phi_T(f_{eos}^{L_T}). \tag{6.2}$$

The *image feature extractor* is also built using the *transformer encoders* (Vaswani et al., 2017). *Transformer encoder* in layer t of the *image feature extractor*, $t \in \{1, 2, \ldots, L_I\}$ is represented as $\mathcal{F}_I^t(\cdot)$. Here L_I represents the number of layers of *transformer encoder* in the *image feature extractor*. First, an input image, I, is split into m number of patches, $\{I_j^p | j = 1, \ldots, m\}$. Next, each patch, I_j^p is transformed into a d dimensional patch embedding vector, $e_j^0 \in \mathbb{R}^d$. All the patch embeddings are represented as $e^0 = \{e_1^0, e_2^0, \ldots, e_m^0\}$. Along with the embedding of the patches, a learnable classifier token, e_{cls}^0 is also added (Vaswani et al., 2017). The patch embeddings are next passed to *transformer encoder* of the first layer. The output patch embedding from layer t of the *transformer encoder*, $\mathcal{F}_I^t(\cdot)$ is represented as follows,

$$[e^t, e_{cls}^t] = \mathcal{F}_I^t(e^{t-1}, e_{cls}^{t-1}), t \in \{1, 2, \ldots, L_I\}. \tag{6.3}$$

The $e_{cls}^{L_I}$ token obtained from the *transformer encoder* of the final layer, is considered as the output of the *image feature extractor*. Alike the text features, the output from *image feature extractor* is projected to the same shared embedding space, using a projection function, $\phi_I(\cdot)$ as follows,

$$g_I = \phi_I(e_{cls}^{L_I}). \tag{6.4}$$

For image classification problem, prompts are created for all possible class labels in the dataset. The text features corresponding to the prompts for all the classes are then projected to the shared embedding space. Let the set of text features for all possible classes be denoted as $g_T = \{g_T^1, \ldots, g_T^{|\mathcal{C}|}\}$ ($|\mathcal{C}|$ is the number of classes). The image features are matched for similarity with the text features of each of the classes. The class for which the similarity between the image features and the corresponding text features is highest, is predicted as the

final class, $\hat{c}$ as follows,

$$p(\hat{c}|I) = \underset{\forall c}{argmax} \frac{\exp(g_I, g_T^c/\tau)}{\sum_{c'=1}^{|\mathcal{C}|} \exp\left(g_I, g_T^{c'}/\tau\right)}. \tag{6.5}$$

Here τ is a user-defined parameter (Radford et al., 2021).

6.3.1 Prompt Guided Disentanglement Network

The schematic diagram of our approach is shown in Fig. 6.1. Our approach comprises of three Guidance branches, viz. the State Guidance Branch, the Object Guidance Branch, the Joint Guidance Branch. Besides, there are three Knowledge Coupling Modules. Next, we describe the different components of our architecture.

6.3.1.1 Feature Disentanglement using Multiple Prompts

The visual features of a state-object compositional image may be decomposed into three components (Hao et al., 2023). The first and the second components are the object agnostic state features and state agnostic object features. The third component represents the features that are formed due to the composition between the state and the object. By identifying the unique features of state and object, a CZSL algorithm becomes more discriminative in nature and is able to better recognise unseen compositions during inference. On the other hand, the state-object joint features are useful to effectively recognise the seen compositions.

To utilise the pre-trained information in CLIP, we propose three prompts, one each for state, object and state-object. The prompt initialisation is of the form `'a photo of <state>'`, `'a photo of <obj>'` and `'a photo of <state><obj>'`. Prompts corresponding to state, object and state-object are passed to the respective *text feature extractor*, known as State Guidance Branch (SGB), Object Guidance Branch (OGB) and Joint Guidance Branch (JGB). The prompts for state, object and joint branches are represented as, $f_{s_i} = [f_0^{s_i}, \ldots, f_k^{s_i}, v_{s_i}], i \in \{1, 2, \ldots, N_s\}$, $f_{o_j} = [f_0^{o_j}, \ldots, f_k^{o_j}, v_{o_j}]$, $j \in \{1, 2, \ldots, N_o\}$ and $f_{c_{ij}} = [f_0^{c_{ij}}, \ldots, f_k^{c_{ij}}, v_{s_i}, v_{o_j}]$, respectively. The learnable part of the prompt are v_{s_i} and v_{o_j} initialised with the state and object labels, respectively. Next, the state prompt is passed to SGB to obtain the state text features, g_{s_i} as $g_{s_i} = \phi_{SGB}(\mathcal{F}_{SGB}(f_{s_i}))$ (similar to (6.1) and (6.2)). Similarly, we obtain the object text features, g_{o_j} and state-object joint text features, $g_{c_{ij}}$ as follows, $g_{o_j} = \phi_{OGB}(\mathcal{F}_{OGB}(f_{o_j}))$ and $g_{c_{ij}} = \phi_{JGB}(\mathcal{F}_{JGB}(f_{c_{ij}}))$. The input image I is passed to the *image feature extractor*, $\mathcal{F}_I(\cdot)$ and the projection function $\phi_I(\cdot)$, as follows, $g_I = \phi_I(\mathcal{F}_I(I))$.

Next, the visual features of the input image and the corresponding textual features of state, object and state-object are utilised to get the corresponding marginal state and marginal object probabilities alongside the joint state-object probability as follows,

$$p(\zeta|I) = \frac{exp(g_I \cdot g_\zeta/\tau)}{\sum_{k=1}^{N_s} exp(g_I \cdot g_\zeta/\tau)}. \tag{6.6}$$

Here $\zeta \in \{s_i, o_j, c_{ij}\}$, represent the state, object and state-object joint classification. Next, the cross-modal knowledge transfer strategy is discussed.

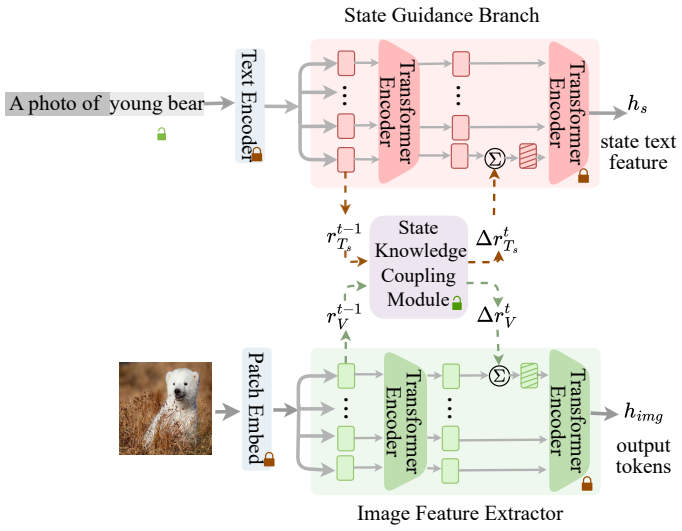


Figure 6.2: Working of the proposed State Knowledge Coupling Module (SKCM).

6.3.1.2 Cross-Modal Knowledge Transfer

Due to challenges in fine-tuning the CLIP (Radford et al., 2021) model, different approaches have been proposed over the years to adapt CLIP for downstream tasks (like image classification and CZSL) (Zhou et al., 2022b,a, Khattak et al., 2023). In this chapter, inspired by Khattak et al. (2023) we incorporate learnable tokens in the intermediate layers of *text* and *image feature extractors* to adapt CLIP for CZSL. However, we observe that the approach in MaPLe (Khattak et al., 2023) is not effective to process the contextual dependency in CZSL problem, as reported next.

Intra-class variation in visual features: In an attempt to solve CZSL, a major challenge lies due to intra-class variation in visual features. For example, as mentioned earlier, in different images of the class *peeled apple*, the apples may be *peeled* to different extent in different images. Subsequently the visual features of *peeled apple* vary significantly, giving rise to intra-class variation in the image features. Also, the text features are incapable to adapt to variation in the image features.

In our approach we utilise three *text feature extractor* branches, one each for the state, object and state-object joint labels. The corresponding *text feature extractors* are termed as SGB, OGB and JGB, respectively. As the tokens from intermediate layers of *image feature extractor* are aware of the variability in image features, the incorporation of the features from the *image feature extractor* helps the SGB, OGB and JGB to better process the context dependency between state and object features. We propose three Knowledge Coupling Modules (KCMs), namely the State Knowledge Coupling Module (SKCM), Object Knowledge Coupling Module (OKCM) and Joint Knowledge Coupling Module (JKCM) (see Fig. 6.1). The intermediate features from the SGB and *image feature extractor* are passed to the SKCM and the obtained output is concatenated with the existing tokens in the *image feature extractor*. Similar approach is followed for OKCM and JKCM. Let us represent the learnable tokens to be inserted in the SGB as $r_{T_s}^t \in \mathbb{R}^d$. The complete process is formally

described as follows.

$$[r_{T_s}^t, f^t, f_{eos}^t] = \mathcal{F}_{SGB}^t(r_{T_s}^{t-1}, f^{t-1}, f_{eos}^{t-1}), t \in \{1, \ldots, L_T\}. \tag{6.7}$$

Here $\mathcal{F}_{SGB}^t(\cdot)$ represents the *transformer encoder* at layer t of the *text feature extractor* in SGB. For OGB and JGB, similar equations like (6.7) may be written. The learnable tokens to be inserted in the *image feature extractor* be represented as $r_I^t \in \mathbb{R}^d$. With the inclusion of r_I^t in the *image feature extractor*, (6.3) may be updated as follows,

$$[r_I^t, e^t, e_{cls}^t] = \mathcal{F}_I^t(r_I^{t-1}, e^{t-1}, e_{cls}^{t-1}), t \in \{1, \ldots, L_I\}. \tag{6.8}$$

Next, the intermediate tokens from *transformer encoders* in layer t of SGB and *image feature extractor* are passed to the State Knowledge Coupling Module (SKCM). The output of the SKCM is added with tokens to be inserted before the *transformer encoders* in layer $(t+1)$ in SGB and *image feature extractor* (see Fig. 6.2). The above-mentioned steps are formally written in the following expressions.

$$[\Delta r_{T_s}^{t-1}, \Delta r_I^{t-1}] = \psi_{SKCM}(r_{T_s}^{t-1}, r_I^{t-1}). \tag{6.9}$$

Here $\psi_{SKCM}(\cdot)$ represents the State Knowledge Coupling Module. Next, to use knowledge from previous stage tokens and the updated tokens, we use residual connection as follows,

$$r_{T_s}^t = r_{T_s}^t + \Delta r_{T_s}^{t-1}, \quad r_I^t = r_I^t + \Delta r_I^{t-1}. \tag{6.10}$$

The updated text and image tokens, $r_{T_s}^t$ and r_I^t are used in (6.7) and (6.8), respectively. We propose to insert the learnable tokens in the early layers of the *transformer encoders* in *text* and *image feature extractors*. Through the learnable tokens, the knowledge is incorporated in the earlier layers of the *transformer encoder* and helps to better process the state and the object features through subsequent layers of *transformer encoder*. Thus we allow addition of learnable tokens in *text* and *image feature extractors* only upto a κ number of layers, where $0 < \kappa < \min(L_I, L_T)$. The parameter κ is referred to as *coupling depth*. Thus we have,

$$\begin{aligned} [f^t] =& \mathcal{F}_{SGB}^t(r_T^{t-1}, f^{t-1}), t \in \{1, 2, \ldots, \kappa\}, \\ [r_{T_s}^t, f^t] =& \mathcal{F}_{SGB}^t(r_{T_s}^{t-1}, f^{t-1}), t \in \{\kappa+1, \ldots, L_T\}. \end{aligned} \tag{6.11}$$

Similarly, for *image feature extractor* we have,

$$\begin{aligned} [e^t, e_{cls}^t] =& \mathcal{F}_I^t(r_I^t, e^{t-1}, e_{cls}^{t-1}), t \in \{1, 2, \ldots, \kappa\}, \\ [r_I^t, e^t, e_{cls}^t] =& \mathcal{F}_I^t(r_I^{t-1}, e^{t-1}, e_{cls}^{t-1}), t \in \{\kappa+1, \ldots, L_I\}. \end{aligned} \tag{6.12}$$

Next, we present the diversity preserving prompt learning approach.

6.3.1.3 Diversity Preserving Prompt Learning

In CZSL problem, the text features are usually extracted from a fixed prompt based template like '`a photo of <state>`' or '`a photo of <state><obj>`'. The template based text features are often invariant to inter-class and intra-class variations in the image. Extraction

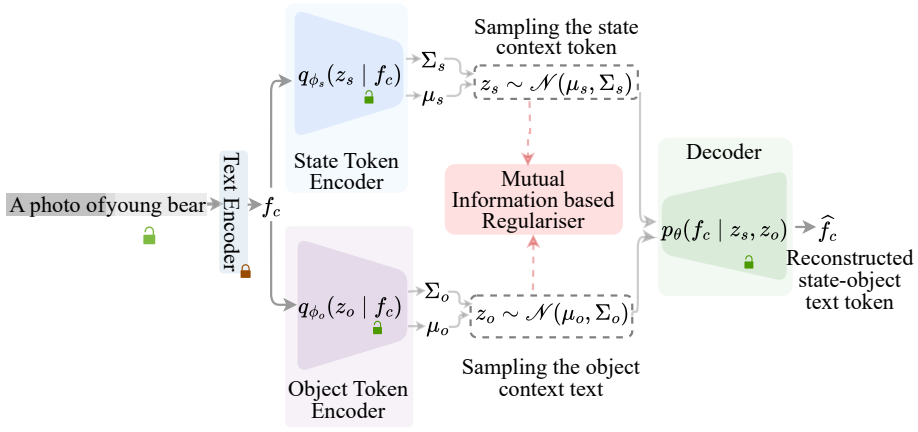


Figure 6.3: Proposed encoder-decoder based diversity preserving prompt tuning approach. We use the usual re-parameterization trick (Kingma and Welling, 2013).

of deterministic text features from a fixed prompt for a class (Khattak et al., 2023, Zhou et al., 2022a,b), reduces the ability of the text features to adapt to variability of the features of the input image. In CZSL, the aforementioned problem is typically more significant as the test set comprises of images from the unseen classes.

The text features obtained from the state-object joint prompts includes of rich contextual information (Huang et al., 2023). On the other hand, the text features extracted from the individual state prompt and the object prompt often lack in the contextual information in the compositional space. Thus, we intend to sample *state context token* and *object context token* from the state-object joint compositional prompt space. Next, the sampled *state* and *object context tokens* are added with the tokens obtained from the existing deterministic state and object prompts and subsequently passed to the SGB and OGB, respectively. The uncertainty in the random sampling process includes diversity in the *state* and *object context tokens*.

Here we propose an *encoder-decoder* based variational inference framework with mutual information based regularization. The variational inference framework helps to estimate the distribution of the token embeddings corresponding to the state prompt and the object prompt, from the joint state-object token embedding. As shown in Fig. 6.3, our approach has two encoders, to estimate the state token embedding distribution representation and the other for the object token embedding distribution representation. Next, from the obtained distribution of the state and the object token embedding, we sample *state* and *object context tokens*. We sample the *state context token* embedding, z_s from the corresponding distribution $q_{\phi_s}(z_s|f_c)$. Similarly we sample the *object context token* embedding, z_o from the distribution $q_{\phi_o}(z_o|f_c)$. Here f_c represents the state-object joint token embedding. Also ϕ_s, ϕ_o and θ are the parameters for the two encoders and the decoder. Next, sampled residual state token embedding and the object token embedding are added to the deterministic state token embedding, f_s^0 and deterministic object token embedding, f_o^0 as follows,

$$f_s^0 = f_s^0 + z_s \text{ and } f_o^0 = f_o^0 + z_o \tag{6.13}$$

Next, we discuss the strategy to train the encoders and the decoder using the varia-

tional inference approach. The marginal likelihood of the state-object joint token embedding $\log p(f_c)$, may be represented as follows,

$$\begin{aligned}\log p(f_c) &= \log \int p_\theta(f_c|z_s, z_o)p(z_s)p(z_o)dz_s dz_o, \\ &= \log \int \frac{p_\theta(f_c|z_s, z_o)q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)p(z_s)p(z_o)}{q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)} dz_s dz_o, \\ &= \log \Bigg(\mathbb{E}_{q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)}[p_\theta(f_c|z_s, z_o)\big(\frac{p(z_s)}{q_{\phi_s}(z_s|f_c)}\big)\big(\frac{p(z_o)}{q_{\phi_o}(z_o|f_c)}\big)]\Bigg),\end{aligned} \tag{6.14}$$

We maximise the marginal likelihood of the state-object joint token embedding, $\log p(f_c)$, by maximising the lower bound, $\mathcal{L}_{LB}$ obtained from (6.14). However for ease of implementation, instead of maximising $\mathcal{L}_{LB}$, we minimise $-\mathcal{L}_{LB}$.

It may be observed that $\mathcal{L}_{LB}$ from (6.14) has no specific expressions to ensure disentanglement between z_s and z_o. Thus to ensure the disentanglement between the sampled z_s and z_o, we propose a mutual information regulariser, as reported next. First, to ensure disentanglement between z_s and z_o, we minimise the mutual information between the token embeddings z_s and z_o, $MI(z_s; z_o)$.

However, only reducing the overlap between z_s and z_o is insufficient for disentanglement of z_s and z_o, if z_s and z_o are not individually informative with respect to f_c. So, we also encourage z_s and z_o to be as much informative as possible with respect to f_c. Thus we maximise the mutual information between f_c and z_s, given z_o, $MI(f_c; z_s|z_o)$. Using a similar argument, we also maximise the mutual information between f_c and z_o, given z_s $MI(f_c; z_o|z_s)$. So, the overall disentanglement specific mutual information based regularizer can be represented as follows,

$$\max_{\phi_s,\phi_o,\theta} \big(MI(f_c; z_s|z_o) + MI(f_c; z_o|z_s)\big) + \min_{\phi_s,\phi_o,\theta} MI(z_s; z_o). \tag{6.15}$$

Next (6.15) can be equivalently represented as $\mathcal{L}_{MI}$,

$$\mathcal{L}_{MI} = MI(z_s; z_o) - MI(f_c; z_s|z_o) - MI(f_c; z_o|z_s). \tag{6.16}$$

Following (Hwang et al., 2020), we can express $MI(z_s; z_o)$ as follows,

$$MI(z_s; z_o) = MI(z_s; f_c) + MI(z_o; f_c) - MI(f_c; z_s, z_o). \tag{6.17}$$

Here $MI(f_c; z_s, z_o)$ represents the mutual information in f_c with respect to z_s and z_o. In our dual-encoder setup consisting of state and object encoders (see Fig. 6.3), minimizing the expression $-MI(f_c; z_s, z_o)$ is equivalent to maximizing $MI(f_c; z_s, z_o)$. However maximizing $MI(f_c; z_s, z_o)$ effectively implies z_s and z_o to be more informative with respect to f_c. We observe that maximizing, $MI(f_c; z_s|z_o)$ and $MI(f_c; z_o|z_s)$ is also effective to make the representations z_s and z_o informative, respectively. Thus we can re-write $\mathcal{L}_{MI}$ as follows,

$$\mathcal{L}_{MI} \approx MI(z_s; f_c) + MI(z_o; f_c) - MI(f_c; z_s|z_o) - MI(f_c; z_o|z_s). \tag{6.18}$$

We expand the expression $MI(z_s; f_c)$ as follows,

$$MI(z_s; f_c) \leq KL(q_{\phi_s}(z_s|f_c)||p(z_s)). \tag{6.19}$$

The detailed derivation of the result in (6.19) is provided in Appendix C, Section C.1. Since (6.18) minimizes the expression $MI(z_s; f_c)$, we minimise the upper bound of $MI(z_s; f_c)$, as obtained from (6.19). Using a similar approach we also get $MI(z_o; f_c)$ as follows,

$$MI(z_o; f_c) \leq KL(q_{\phi_o}(z_o|f_c)||p(z_o)). \tag{6.20}$$

Next, we consider the expansion of the terms $MI(f_c; z_s|z_o)$ and $MI(f_c; z_o|z_s)$ from (6.18), as follows,

$$\begin{aligned} MI(f_c; z_s|z_o) &= \int p(f_c, z_s, z_o) \log \frac{p(f_c, z_s|z_o)}{p(f_c|z_o)p(z_s|z_o)} dz_o dz_s, \\ &= \int p(f_c, z_s, z_o) \log \frac{p(f_c, z_s, z_o)}{p(f_c|z_o)p(z_s, z_o)} dz_o dz_s, \\ &= H(f_c) + \mathbb{E}_{q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)}\, p_\theta(f_c|z_s, z_o) - KL(q_\phi(z_o|f_c)||p(z_o)). \end{aligned} \tag{6.21}$$

The derivation of the result in (6.21) is reported in Appendix C, Section C.1. Evidently $\mathcal{L}_{MI}$ is minimized over the learnable parameters ϕ_s, ϕ_o and θ_s. Hence, the term $H(f_c)$ is ignored from the evaluation of $\mathcal{L}_{MI}$. The final loss to be optimised is as follows,

$$\begin{aligned} &\mathcal{L}_{enc\_dec} = -\mathcal{L}_{LB} + \eta\mathcal{L}_{MI} \\ &= (1 + 2\eta)\Big(KL\big[q_{\phi_s}(z_s|f_c)||p(z_s)\big] + KL\big[q_{\phi_o}(z_o|f_c)||p(z_o)\big]) - \mathbb{E}_{q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)} \log\big(p_\theta(f_c|z_s, z_o)\big)\Big). \end{aligned} \tag{6.22}$$

Here η is an user defined hyper parameter. In (Hwang et al., 2020), authors utilise a triple-encoder dual-decoder setup for extracting domain invariant and domain specific features for the problem of sketch-based image retrieval. In this chapter, we have used a dual encoder-single decoder framework for effective disentanglement of the state and object token embedding distribution in context of CZSL problem.

6.3.1.4 Scale-aware Feature Extraction

The VLMs like CLIP are specifically trained for object recognition and the state recognition remains challenging using CLIP based feature extractors (Nayak et al., 2022, Lu et al., 2023). We observe that the state is dependent on the scales of objects in the input image. The features of the state *young* and *old* for a *bear* can be inferred by the size/scale of a *bear* in the image. So, identifying scale-aware features is an important requirement for CZSL. Next, we describe our approach to process mutli-scale features.

We randomly crop R number of patches at different scales from image, I. Next, obtained cropped patches are re-scaled to the resolution of I. Let the up-sampled images be represented as $I_{sc} = \{I^1, I^2, \ldots, I^R\}$. Next, we pass each of those patches to the *image feature extractor*. The extracted image features corresponding to all the crops are represented as follows, $g_{sc} = \{g^1_{sc}, g^2_{sc}, \ldots, g^R_{sc}\}$. Using the text features of the state and the object labels at the output

of the SGB and OGB, we obtain the state probability for i^{th} state class from the cropped image patch r as follows. Next the probability of observing the state s_i from the R number of cropped and re-scaled image patches is represented as,

$$p_{sc}(s_i|I) = \frac{1}{R}\sum_{r=1}^{R}\frac{exp(g_{sc}^{r}\cdot g_{s_i}/\tau)}{\sum_{k=1}^{N_s} exp(g_{sc}^{r}\cdot g_{s_k}/\tau)}. \tag{6.23}$$

Here $i \in \{1, 2, \ldots, N_s\}$. Next, we aggregate the state probability obtained from (6.6) and the scale aware state probability from (6.23) to obtain the final state probability as written next,

$$\overline{p}(s_i|I) = \nu \cdot p_{sc}(s_i|I) + (1-\nu)\cdot p(s_i|I). \tag{6.24}$$

Here ν is a user defined scalar, $\nu = 0.5$. Similarly, we also obtain the aggregated object probability for object class j as $p(o_j|I)$. Next, we analyze the risk in predicting state-object compositions compared to individual state or object predictions.

6.3.1.5 Analysis of Risk in the State-object Joint Prediction

We have used a multi-branch architecture, where the state and the object predictions are obtained from disentangled state and object *text feature extractors*. In this section, we theoretically analyse the effectiveness of this multi-branch approach in error free prediction. We attempt to find the relationship between probability of error of the individual state and object predictions and the corresponding state-object joint prediction. Let h_s represent a hypothesis that maps the input image, I to a state label y_s, $h_s : I \rightarrow y_s$. Also the collection of all such possible hypotheses are represented by the set $\mathcal{H}_s$, where $h_s \in \mathcal{H}_s$. Next, we report the definition of the *risk* of the hypothesis h_s as follows,

$$\mathcal{R}(h_s) = P_{I\sim\mathcal{D}}[h_s(I) \neq y] = \mathbb{E}_{I\sim\mathcal{D}}\left[\mathbb{1}_{h_s(I)\neq y_s}\right]. \tag{6.25}$$

Here $\mathbb{1}_{h_s(I)\neq y_s}$ is an indicator random variable which is only true if the event $h_s(I) \neq y_s$ is true. However, evaluating the *risk*, $\mathcal{R}(h_s)$ requires knowing the data distribution, which is intractable. Hence, the idea of *empirical risk* was proposed. For a training dataset $\mathrm{T} = \{I_1, I_2, \ldots, I_{N_t}\}$, the corresponding *empirical risk* is evaluated as follows,

$$\hat{\mathcal{R}}_{\mathrm{T}}(h_s) = \frac{1}{N_t}\sum_{k=1}^{N_t}\left[\mathbb{1}_{h_s(I_k)\neq y_s}\right]. \tag{6.26}$$

For the sake of simplicity, we assume that the state prediction is a two class problem, i.e. $h_s : I \rightarrow \{0, 1\}$. Similarly, let $\mathcal{H}_o$ represent the set of hypotheses that map the input image to the object class probabilities. We assume that the object prediction is a two class prediction problem, i.e. $h_o : I \rightarrow \{0, 1\}$.

Next, the state-object joint prediction is obtained by performing the element-wise product of the corresponding state and object predictions. The hypothesis that maps the input image to the state-object joint prediction is represented as $h_c = h_s \cdot h_o$, with $h_c \in \mathcal{H}_c$. Next, we propose the following proposition,

Proposition 1: With probability $(1-\delta)$, $\delta > 0$, the empirical risk of the state-object joint

hypotheses may be bounded as follows,

$$\hat{\mathcal{R}}(h_c) \leq \mathcal{R}(h_c) + \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_s) + \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_o) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2N_t}}. \tag{6.27}$$

Here $\hat{\mathfrak{R}}_{\mathrm{T}}(\cdot)$ represents the Rademacher complexity (Bartlett and Mendelson, 2002), defined on the training set T as follows (Bartlett and Mendelson, 2002),

$$\hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_s) = \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{n} \sigma_i h_s(x_i) \right]. \tag{6.28}$$

Here $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_n]$ represents uniform random variable in the interval $\{-1, +1\}$, also referred as the Rademacher variable (Bartlett and Mendelson, 2002). The result in *proposition* 1 implies that the risk in state-object joint prediction is upper-bounded by the risk in the state prediction and risk in the object prediction. Next, we proceed with the proof of *proposition* 1. To proof *proposition* 1 it is sufficient to prove the following,

$$\hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_c) \leq \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_s) + \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_o). \tag{6.29}$$

We utilise the following expression in the subsequent steps, $h_s h_o = (h_s + h_o - 1)\mathbb{1}_{h_s+h_o>1}$.

$$\begin{aligned}
\hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_c) &= \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i h_c(x_i) \right] \\
&= \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i h_s(x_i).h_o(x_i) \right] \\
&= \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i (h_s + h_o - 1)\mathbb{1}_{h_s+h_o>1}(x_i) \right] \\
&= \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i \mathbb{1}_{h_s+h_o>1} h_s(x_i) + \sigma_i \mathbb{1}_{h_s+h_o>1} h_o(x_i) - \mathbb{1}_{h_s+h_o>1}(x_i) \right]
\end{aligned} \tag{6.30}$$

Since the function $v : x \to (x-1)$ is 1-Lipsciltz continuous in the interval $[0, 2]$, we can apply Talagrands Contraction lemma (Talagrand, 1995) to obtain the following results,

$$\begin{aligned}
\hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_c) &\leq \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i h_s(x_i) \right] + \mathbb{E}_\sigma \left[\sup \frac{1}{N_t} \sum_{i=1}^{N_t} \sigma_i h_o(x_i) \right], \\
&\leq \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_s) + \hat{\mathfrak{R}}_{\mathrm{T}}(\mathcal{H}_o). \qquad \square
\end{aligned} \tag{6.31}$$

Next, using the result from (6.31) in the basic expression of Rademacher complexity (Bartlett and Mendelson, 2002), we get the result of *Proposition* 1. Next, we describe the training and the inference strategies for our approach.

6.3.1.6 Training Strategy

Proposed approach is trained using a combination of five loss components as discussed next. The state loss, $\mathcal{L}_s(\cdot)$ is evaluated using the marginal probability of state classes (see (6.6)) and the state ground truth, GT_s as follows,

$$\mathcal{L}_s = \mathcal{L}_{CE}(p(s|I), GT_s). \tag{6.32}$$

Here $\mathcal{L}_{CE}(\cdot)$ represent the usual cross-entropy loss. Using a similar approach we obtain the object marginal probability from (6.6). Next, the object ground truth, GT_o is used to evaluate the object cross-entropy loss, as follows,

$$\mathcal{L}_o = \mathcal{L}_{CE}(p(o|I), GT_o). \tag{6.33}$$

Similarly, the loss corresponding to the state-object joint probability obtained from (6.6) is utilised as follows,

$$\mathcal{L}_c = \mathcal{L}_{CE}(p(c|I), GT_c). \tag{6.34}$$

Similarly, the state and the object probabilities corresponding to the scale-aware cropped images from (6.23) is represented as follows,

$$\mathcal{L}_s^{sc} = \mathcal{L}_{CE}(\overline{p}(s|I), GT_s) \text{ and } \mathcal{L}_o^{sc} = \mathcal{L}_{CE}(\overline{p}(o|I), GT_o). \tag{6.35}$$

In addition to the aforementioned loss components, we use (6.22) to obtain the final loss for the our approach as follows,

$$\mathcal{L} = (\mathcal{L}_s + \mathcal{L}_o) + \mathcal{L}_c + \lambda_1(\mathcal{L}_s^{sc} + \mathcal{L}_o^{sc}) + \lambda_2 \mathcal{L}_{enc\_dec}. \tag{6.36}$$

Here λ_1 and λ_2 are scalar values.

6.3.1.7 Inference Strategy

During inference, the state-object textual features are obtained from the JGB. The image features are obtained from the *image feature extractor*. Next, we multiply the obtained image features with the text features from the JGB to obtain p_c (see (6.6)). The particular state-object compositional class that achieves the highest probability is predicted as the final compositional label for the input image.

Following prior approaches (Lu et al., 2023, Mancini et al., 2022), we utilise a feasibility-based filtering strategy to eliminate infeasible compositions in open-world CZSL. For each unseen state-object composition, $c_u = (s', o')$, we evaluate the feasibility from the perspective of the object as follows. The maximum *cosine similarity* between the word embedding (Pennington et al., 2014) of the object label, o' and word embedding (Pennington et al., 2014) of all other objects that form feasible compositions with the state s', is evaluated and represented as $z_{c_u}^{o'}$. Similarly, the feasibility of c_u from state perspective, $z_{c_u}^{s'}$ is evaluated. The final feasibility z_{c_u} is evaluated as, $z_{c_u} = (z_{c_u}^{o'} + z_{c_u}^{s'})/2$. Next, composition, c_u is considered as a possible outcome of the model in open-world CZSL if $z_{c_u} \geq \gamma$. Here γ is a user-defined threshold. Additional details regarding the inference strategy is provided in Appendix C, Section C.2.1. Next we present the experiment section.

Dataset →	MIT-States				C-GQA				UT-Zappos50k			
Algorithm↓	*AUC*	*seen*	*unseen*	*HM*	*AUC*	*seen*	*unseen*	*HM*	*AUC*	*seen*	*unseen*	*HM*
LE	2.0	15.0	20.1	10.7	0.8	18.1	5.6	6.1	25.7	53.0	61.9	41.0
AAO	1.6	14.3	17.4	9.9	0.7	17.0	5.6	5.9	25.9	59.8	54.2	40.8
TMN	2.9	20.2	20.1	13.0	1.1	23.1	6.5	7.5	29.3	58.7	60.0	45.0
SymNet	3.0	24.2	25.2	16.1	2.1	26.8	10.3	11.0	23.4	49.8	57.4	40.4
CompCos	4.5	25.3	24.6	16.4	2.6	28.1	11.2	12.4	28.1	59.8	62.5	43.1
CGE	6.5	32.8	28.0	21.4	4.2	33.5	15.5	16.0	33.5	64.5	71.5	60.5
Co-CGE	6.6	32.1	28.3	20.0	4.1	33.3	14.9	15.5	33.9	62.3	66.3	48.1
SCEN	5.3	29.9	25.2	18.4	5.5	28.9	25.4	17.5	32.0	63.5	63.1	47.8
CSP	19.4	46.6	49.9	36.3	6.2	28.8	26.8	20.5	33.0	64.2	66.2	46.6
DFSP	20.6	46.9	52.0	37.3	10.5	38.2	32.0	27.1	36.0	66.7	71.7	47.2
HPL	20.2	47.5	50.6	37.3	7.2	30.8	28.4	22.4	35.0	63.0	68.8	48.2
Troika	22.1	49.0	53.0	39.3	12.4	41.0	35.7	29.4	41.7	66.8	73.8	54.6
Our Approach	**22.6**	**51.0**	**54.1**	**41.0**	**13.0**	**43.1**	**37.4**	**30.9**	**42.4**	**69.9**	**75.2**	**56.0**

Table 6.1: Results on the CW-CZSL problem.

6.4 Experiments

6.4.1 Implementation Details.

Following prior approaches (Nayak et al., 2022, Lu et al., 2023), *image feature extractor* and *text feature extractor* are implemented using a pre-trained CLIP ViT-L/14 model (Radford et al., 2021). All three KCMs are implemented using a Multi-Layer Perceptron (MLP), a Gaussian Error Linear Units (GELU) (Hendrycks and Gimpel, 2016), followed by another MLP. We have used the Adam optimiser (Kingma and Ba, 2015) with learning rate $5e-5$ (for C-GQA and MIT-States) and $1e-5$ (for UT-Zappos50k). A weight decay parameter of $5e-5$ has been used. The models across all three datasets are trained for 50 epochs. In (6.36), $\lambda_1 = 0.3$ and $\lambda_2 = 0.1$ values are used. We set the value of $\kappa = 10$ for MIT-States and UT-Zappos50k. $\kappa = 9$ is used for C-GQA. The batch size of 32, 16 and 32 is used for MIT-States, C-GQA and UT-Zappos50k datasets, respectively. For open-world CZSL, the prediction threshold is taken as $\gamma = 0.4$. The model is implemented using PyTorch 2.0 (Paszke et al., 2019) on a dual GPU set-up consisting of two NVIDIA RTX Titan GPUs with total of 48 GB memory, CUDA 12.1 on a PC with 128 GB RAM.

6.4.2 Compared Algorithms.

We compare the proposed approach against other state-of-the-art CZSL algorithms: LE (Misra et al., 2017), AAO (Nagarajan and Grauman, 2018), TMN (Purushwalkam et al., 2019), SymNet (Li et al., 2020), CGE (Naeem et al., 2021), Co-CGE (Mancini et al., 2022), CompCos (Mancini et al., 2021), KG-SP (Karthik et al., 2022), SCEN (Li et al., 2022), CSP (Nayak et al., 2022), DFSP *(t2i)* (Lu et al., 2023), HPL (Wang et al., 2023a) DRA-Net (Li et al., 2023b) and Troika (Huang et al., 2023). Literature review of these algorithms is done in Section 6.2. The results for the compared algorithms are reproduced from (Lu et al., 2023, Mancini et al., 2022, Li et al., 2023b) and the respective open-source implementations.

Dataset →	MIT-States				C-GQA				UT-Zappos50k			
Algorithm↓	*AUC*	*seen*	*unseen*	*HM*	*AUC*	*seen*	*unseen*	*HM*	*AUC*	*seen*	*unseen*	*HM*
LE	0.3	14.2	2.5	2.7	0.08	19.2	0.7	1.0	16.3	60.4	36.5	30.5
AAO	0.7	16.6	5.7	4.7	-	-	-	-	13.7	50.9	34.2	29.4
TMN	0.1	12.6	0.9	1.2	-	-	-	-	8.4	55.9	18.1	21.7
SymNet	0.8	21.4	7.0	5.8	0.43	26.7	2.2	3.3	18.5	53.3	44.6	34.5
CGE	1.0	32.4	5.1	6.0	0.47	32.7	1.8	2.9	23.1	61.7	47.7	39.0
CompCos	1.6	25.4	10.0	8.9	0.39	28.4	1.8	2.8	21.3	59.3	46.8	36.9
KG-SP	1.3	28.4	7.5	7.4	0.78	31.5	2.9	4.7	26.5	61.8	52.1	42.3
DRA-Net	1.5	29.8	7.8	7.9	1.05	31.3	3.9	6.0	28.8	65.1	54.3	44.0
CSP	5.7	46.3	15.7	17.4	1.20	28.7	5.2	6.9	22.7	64.1	44.1	38.9
DFSP	6.8	47.5	18.5	19.5	2.4	38.3	7.2	10.4	30.3	66.8	60.0	44.0
Troika	7.2	48.8	18.7	20.1	2.7	40.8	7.9	10.9	33.0	66.4	61.2	47.8
Our Approach	**8.7**	**50.2**	**20.8**	**21.8**	**3.4**	**41.8**	**8.8**	**11.9**	**33.8**	**67.8**	**62.5**	**49.1**

Table 6.2: Results on the OW-CZSL problem.

6.4.3 Results

Quantitative results. We compare the proposed approach on the CW-CZSL and OW-CZSL CZSL in Tables 6.1 and 6.2, respectively. Compared CZSL approaches are divided into two groups, one group with conventional feature extractors (He et al., 2016, Vaswani et al., 2017) and the other group being VLM based approaches, separated by a horizontal line in Tables 6.1 and 6.2. On the *AUC* metric, our approach outperforms the existing CZSL approaches on the closed-world setting by at least 0.5% across all three datasets.

One of the contributions of our approach is the use of three distinct prompts for effective isolation of state, object and state-object joint features. We observe that the effective isolation of unique features of state and object helps the model to better generalise on unseen compositions during test. Our approach achieves at least 1.1% improvement on *unseen* metric across all three datasets on closed-world CZSL. Similar improvement is also observed on *seen* metric. The improvements in *unseen* and *seen* metrics justify the use of three distinct prompts.

Another contribution of our approach is to process the contextual dependency between visual features of state and object. For processing the contextual dependency, our approach utilises the KCMs. We report the state and object recognition accuarcies of our approach and other compared approaches in Table 6.3. Due to the contextual dependency, higher feature variations are observed in the features of state in comparison to object. From Table 6.3, our approach achieves improvements of 4.8% and 3.2% in state recognition accuracies on MIT-States and C-GQA, respectively. The improvement in state recognition accuracy justifies the effectiveness of KCMs in our approach.

In all three datasets, in both open-world and closed-world CZSL, the closest results to our approach are reported by Troika (Huang et al., 2023). From the implementation angle, the 'cross-modal traction' (CMT) module in Troika uses a scaled dot-product attention (Vaswani et al., 2017) of text and image features. The results of the scaled dot-product are added with the text features (see (18) in Troika). On the contrary, in our approach using KCMs, we have shared knowledge from the *image feature extractor* to guide the text features. Hence, the approach in Troika is less effective for CZSL problem than our approach in processing

the intra-class feature variation in CZSL. Besides proposed multi-scale feature extraction in our approach helps to better solve scale-aware feature aggregation problem. HPL (Wang et al., 2023a) is even less effective to process intra-class variation in image features than our approach and Troika, due to absence of knowledge sharing between the *text* and *image feature extractors*. Qualitative classification results of our approach is reported next.

Figure 6.4: Qualitative image classification results: The green coloured texts represent the correctly predicted labels and the red coloured texts represent incorrect predictions.

Qualitative results. The qualitative image classification results of our approach is reported in Fig 6.4. The states like *ripe*, *peeled* and *sliced* are difficult to identify, due to variation in visual features among different compositions of these states. Evidently the proposed approach successfully recognises the compositions *ripe banana*, *peeled orange* and *sliced bread*. However, proposed approach incorrectly recognises the composition *dark sky* as *cloudy sky*. It can be observed that the predicted state (*cloudy*) is also relevant to the input image of *dark sky*. It has been reported in (Atzmon et al., 2020b) that in CZSL datasets (Isola et al., 2015) often more than one labels from the dataset are relevant for an input image. Proposed approach incorrectly predicts some compositions due to multiple relevant state labels being present in the dataset.

6.4.4 Ablation Study

On utility of the three prompts. The model is trained excluding the state prompt, f_s, object prompt, f_o and state-object joint prompt, f_c, one by one and report the respective *val AUC*

Dataset →	MIT-States		C-GQA	
Algorithm↓	*state*	*object*	*state*	*object*
TMN	23.3	26.5	16.5	25.6
SymNet	26.3	28.3	21.4	25.1
CompCos	27.9	28.3	20.2	28.4
CGE	35.7	44.4	17.5	34.6
SCEN	28.2	32.2	13.0	27.9
CSP	40.6	49.7	36.1	40.8
DFSP	47.8	57.8	40.0	43.6
Our Approach	**52.6**	**60.8**	**43.2**	**47.8**

Table 6.3: State and object recognition results in CW-CZSL.

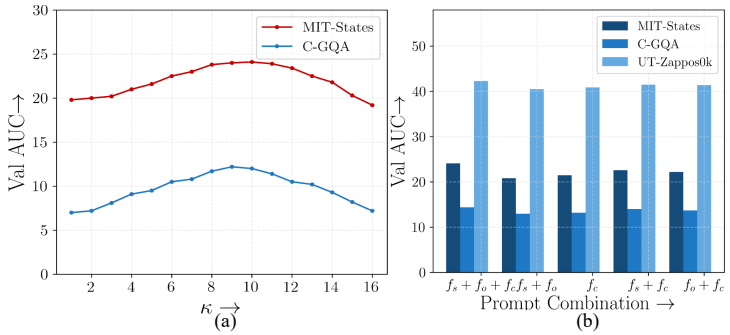


Figure 6.5: (a) Analysis to evaluate the utility of three different prompts. (b) Visualization of the variation of the *val AUC* of the model with variation of coupling depth, κ.

in Fig. 6.5(b). While excluding a particular prompt we also exclude the corresponding KCM and GB from training. Between f_s and f_o, we observe that inclusion of f_s causes higher improvement on *val AUC*. This is due to the fact that state features poses more variability due to contextual dependency. Thus state recognition is more difficult than object recognition and f_s has a stronger effect on the *val AUC* than f_o. Across MIT-States and C-GQA, the inclusion of f_c contributes in highest improvement in *val AUC*. Here f_c is useful in extracting the state-object joint features from the pre-trained information already present in CLIP (Radford et al., 2021). Thus f_c helps in recognising the seen compositional images.

On the effectiveness of the coupling depth. The variation of *val AUC* with *coupling depth*, κ, is shown in Fig. 6.5(a). Highest *val AUC* is obtained for the optimum value of $\kappa = 10$ for MIT-States and $\kappa = 9$ for C-GQA. As discussed in Section 6.3.1.1, the inclusion of learnable tokens is intended to help the model better understand the contextual dependency between state and object features. When κ is low, the learnable tokens are inserted only into a few early layers of *image feature extractor* and *text feature extractor*. Thus a low value of κ is not sufficient to process the contextual dependency that is prevalent between state and object features. Higher value of κ implies that the learnable tokens are integrated in earlier as well as later layers of *image feature extractor* and *text feature extractor*. Inserting a learnable token at the later layers, perturbs the fine-grained features extracted in the *transformer encoder*.

On the loss components. The *val AUC*, after training the network using different loss combinations (see (6.36)), is reported in Table 6.4. The results in the cells at the third row

Loss Components					MIT-States	C-GQA
$\mathcal{L}_s$	$\mathcal{L}_o$	$\mathcal{L}_c$	$\mathcal{L}_s^{sc}$	$\mathcal{L}_o^{sc}$	*val AUC*	*val AUC*
✓	✓	×	✓	✓	19.0	12.5
×	×	✓	✓	✓	19.4	12.9
✓	×	✓	✓	✓	22.3	13.9
×	✓	✓	✓	✓	21.7	13.7
✓	✓	✓	×	×	23.1	14.0
✓	✓	✓	✓	×	23.5	14.2
✓	✓	✓	×	✓	23.3	14.1
✓	✓	✓	✓	✓	**24.1**	**14.4**

Table 6.4: Ablation study on effectiveness of the loss components in the closed-World CZSL.

and last row of Table 6.4 show that inclusion of $\mathcal{L}_c$ improves *val AUC* by 5.1% and 1.9% for the MIT-States and C-GQA datasets, respectively. Among $\mathcal{L}_s$ and $\mathcal{L}_o$, $\mathcal{L}_s$ is observed to have dominant effect on the model's performance. Comparing the results reported in the Table 6.4 (sixth and last rows), $\mathcal{L}_s$ improvements of 2.4% and 0.7% on *val AUC* can be observed on MIT-States and C-GQA respectively. Due to inclusion of $\mathcal{L}_o$, improvements of 1.8% and 0.5% on *val AUC* are observed for MIT-States and C-GQA respectively. $\mathcal{L}_s$ has a stronger effect over $\mathcal{L}_o$ as contextual dependency is dominant for state features than the object features. An important aspect of our approach is the inclusion of the multi-scale images to extract the scale-aware visual features. To evaluate the effectiveness of scale-aware feature extraction, we experiment by training the model by including and excluding the scale-aware feature aggregation module. The relevant results are reported in last 3 rows of the Table 6.4. Evidently by inclusion of $\mathcal{L}_s^{sc}$ and $\mathcal{L}_o^{sc}$, we achieve 0.8% and 0.6% improvement in *val AUC* for MIT-States dataset. Thus the proposed approach reports higher improvement in *val AUC* due to inclusion of multi-scale image features during training.

On effectiveness of the diversity preserving prompt tuning. Next, we analyse the effectiveness of the variational inference approach and the mutual information based regularization to incorporate diversity in the text feature from deterministic state and object prompts (see Section 6.3.1.3). To see the efficacy of the aforementioned approach, the model is trained by excluding the loss $\mathcal{L}_{enc\_dec}$ from (6.22). We also exclude the *state context token* and the *object context token* as added in (6.13). On the MIT-States dataset, the *val AUC*, *seen* and the *unseen* metrics report 0.2%, 0.7% and 1.2% improvements, respectively, when trained including the *state* and the *object context tokens* and $\mathcal{L}_{enc\_dec}$. The improvements observed in results may be attributed to the fact that the tokens obtained from the state-object joint prompt have richer contextual information. Also, our approach of randomly sampling the *state* and the *object context tokens* from the corresponding distributions helps to preserve the diversity. Further, formulating prompt learning as a variational inference (VI) problem is shown to improve generalization ability of the model on unseen prompts compared to a deterministic prompt based approaches (Huang et al., 2023, Nayak et al., 2022, Lu et al., 2023, Wang et al., 2023a, Huang et al., 2023). As already reported, highest improvement is achieved across the unseen class recognition accuracy which justifies the better generalizability of our model to prompts for unseen compositional classes.

Comparison with MaPLe (Khattak et al., 2023). To evaluate the effectiveness of the proposed approach over MaPLe, we use the implementation of MaPLe to train and evalu-

ate MaPLe on CZSL. MaPLe reports 1.6% and 0.9% lower *val AUC* over our approach on MIT-States and C-GQA, respectively. The better result as reported by our approach can be justified as follows. In our approach, using KCMs, the intermediate layer knowledge from *image feature extractors* is shared to *text features extractors* and vice-versa. However in MaPLe only text features are shared to *image feature extractor*. As discussed in Section 6.3.1.2, the visual features in CZSL suffer from intra-class variations. The text features extracted from the state and object labels are fixed for each state-object composition. Thus text features are unable to adapt to the inter-class variations in different images. Our approach shares intermediate layer knowledge from *image feature extractor* to the intermediate layers of *text feature extractors* to better process the intra-class variation. Besides the proposed multi-scale feature aggregation approach is also useful to better extract the scale-aware features in CZSL. We incorporate additional ablation study experiments like the analysis of the loss coefficient, analysis of the number of crops R as used in (6.23), in the Appendix C, Section C.2, under heading, "Ablation study on the number of cropped images".

6.5 Summary

In this chapter, a novel CLIP based approach for CZSL is proposed. To effectively disentangle the state and object features, we propose three independent prompts in our approach. To process the contextual dependency between features of state and object in CSL, we adapt the CLIP model by integrating set of learnable tokens in the intermediate layers of *image feature extractor* and *text feature extractor*. We also explored a multi-scale feature extraction approach to better extract the multi-scale features in CZSL. Besides we explore the risk in predictions of the current approach. Proposed approach outperforms conventional vision based approaches as well as the CLIP based approaches for CZSL. In the next chapter we conclude the thesis by summarising the work done in the thesis and reporting the future direction of research.

Chapter 7

Conclusion

7.1 Contributions

CZSL is a relatively new problem, first explored by Mishra *et. al*, in 2017 (Misra et al., 2017). Next, Purushwalkam et al. (2019) rationalized the CZSL problem statement by proposing a set of new train-test splits and evaluation metrics. Still, there are a lot of challenges in the CZSL problem. In this this thesis, we have addressed some of the challenges in CZSL problem.

The isolation of visual features of state and object in CZSL is inherently challenging, due to entanglement between visual features of constituent primitives. The focus of the second chapter of our work is the disentanglement of the visual features of state and object. A sequential learning approach with a gradient penalization loss component for effective disentanglement of constituent primitives is also proposed.

In the chapter 3, we have worked on the OW-CZSL problem. Here, we have proposed a new frequency-based feasibility prediction approach for estimating the feasibility of the unseen state-object compositions. We have also proposed a graph-based architecture with cross-layer knowledge sharing, for modelling the contextual dependency between state and object features.

Next, in chapter 4, we have attempted to model the existence of partial association among the features of state and object in a compositional image. We have proposed a novel Knowledge Guided Transformer Network for modelling the partial association. To the best of our knowledge, no existing CZSL algorithm has made attempt to model partial association between the visual features of state and object. Besides, we propose a Layer-wise Adaptive Attention Aggregation approach to better process the feature propagation through the multiple layers of transformer encoder in the proposed approach.

In chapter 5, we have explored the pCZSL problem. The pCZSL requires less amount of training data as for each compositional image either the state or the object annotation is accessible for training purpose. Thus to utilise the large amount of partially labelled data in pCZSL, we propose an α divergence based distribution alignment loss. We have also reported a class-balancing strategy in attempt to take care of the data imbalance issue in the CZSL datasets. Also, a novel Locality Preserving Neighbourhood Aggregation approach is proposed for the pCZSL problem.

Lastly in Chapter 6, we have worked on a Vision Language Model (VLM) based CZSL

approach. To better adapt to the contextual dependency in image features, we proposed a variational-inference based framework to incorporate diversity in the textual features, extracted from a VLM. Besides, we also theoretically analyse the risk in state-object joint prediction in case of CZSL.

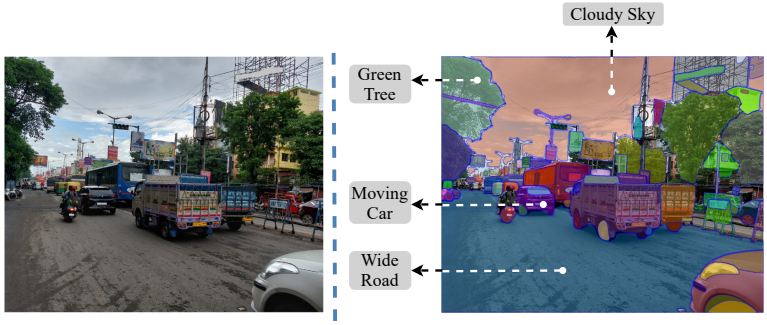


Figure 7.1: A special use case of the CZSL problem in context of autonomous driving: A scene (sub-figure (a)) decomposed into multiple segments (sub-figure (b)). We propose to recognise the *state-object* compositions in context of autonomous driving. The states may be local in nature, like *moving*, *green* and *wide* for the corresponding compositions like *moving car*, *wide road* and *green tree*.

7.2 Future Directions

CZSL is an interesting data efficient learning approach that has only been in focus of the research community in the recent years. There are many potential applications of CZSL which remains to be explored. In the following sections, we note two such potential applications of CZSL.

7.2.1 Recognising Unseen State-object Compositions in the Wild

Object recognition in the wild is a long standing challenge in the relevant scientific community. Existing researches in the field of CZSL have not been explored this direction. For example, it will be interesting to work on a potential application of the CZSL problem, to recognise the unseen state-object compositions in context of autonomous driving systems.

We propose that the view obtained from a dashboard-mounted camera (in an autonomous vehicle) should be segmented into state and object. For example, a typical scene as in Fig. 7.1(a), could be described as a collection of segments like *wide road*, *cloudy sky*, *tall building* and so on as shown in Fig. 7.1(b). A scene from the video of a front-view camera on board an autonomous vehicle, is often very complicated and difficult to interpret. A typical such scene is composed of a number of segments which are changing fast. Definitions and compositions of such scenes change depending on a number of factors. A common scenario would be the variation in visual appearances due to videos taken in day time or in the evening. Similarly, weather conditions, for example *sunny* and *foggy* changes in definitions of scene compositions. Due to fast changing scene contents, the scene descriptions get complicated often due to merging segments or when disconnected segments are generated.

7.2.2 Recognising Unseen Compositions in the Multi-attribute Environment

Current state-of-the-art CZSL datasets evaluate effectiveness of an algorithm by recognising a single state and a single object. Real-world situations usually involve compositions having multiple objects. Besides, each object generally occurs with multiple states. In an image where multiple states are present, annotating such an image as only a single state composition usually introduces annotation bias, as explained next.

For example, if an image of *peeled ripe banana* is annotated as *ripe banana*, the model will incorrectly reduce the likelihood that the image belongs to the two states *peeled* and *ripe*. It may be noted that the states *peeled* and *ripe*, both are valid states in the CZSL dataset (Isola et al., 2015). Insufficient state annotations (in the aforementioned compositional images) will mislead the model into overlooking the correct states that actually exist.

Besides, the existing single-state-single-object datasets (Isola et al., 2015, Naeem et al., 2021, Yu and Grauman, 2014, 2017) only consider the contextual dependency between states and the objects, without accounting for the co-occurrence relationships between different states. For instance, the states *brown* and *burnt* frequently co-occur when depicting the images of *burnt* food, and understanding these associations can enhance the capabilities of the model to estimate the discriminative features of the constituent primitives.

7.2.3 Open Vocabulary State-object Compositions Recognition

In CZSL datasets, the test set consists of images of unseen state-object compositions. However, the set of states and the set of objects are same in both the training set and the test set. The recently proposed VLMs (Radford et al., 2021, Jia et al., 2021) are pre-trained on massive amount of image-text paired data. By leveraging the pre-trained information in VLMs, it has been shown that the zero-shot recognition of unseen object classes is fesible (Radford et al., 2021). Inspired by the aforementioned observations, in future a more generalised CZSL problem, where the test set may consist of new states and new objects (not present in the training set), may be explored.

Appendix A

Supplementary for Chapter 2

A.1 Derivation the Result Used in (2.10)

The detailed steps to simplify (2.10) are given as follows,

$$\begin{aligned} MI(f_{img}, f_{state}|f_{obj}) &= MI(f_{state}, f_{img}) - MI(f_{state}, f_{obj}) \\ &= H(f_{state}) + H(f_{img}) - H(f_{state}, f_{img}) - H(f_{state}) - H(f_{obj}) + H(f_{state}, f_{obj}) \\ &= H(f_{img}) - H(f_{state}, f_{img}) - H(f_{obj}) + H(f_{state}, f_{obj}), \end{aligned} \tag{A.1}$$

where $H(\cdot)$ represents the entropy (Cover, Thomas M, 1999). For any two random variables X and Y, we can write $MI(X, Y) = H(X) + H(Y) - H(X, Y)$ (Cover, Thomas M, 1999).

Next we derive (2.17). As already explained, k represents the minimum number of images in which a features of a state can be present. Thus we can have $\sum_s p(f^r_{state}|f^s_{img}) \geq k * p(f^r_{state}|f^r_{img})$. So we have

$$H(f_{state}) \geq -\sum_r p(f^r_{state}|f^r_{img}) \frac{1}{N} \log\left(\frac{k}{N} p(f^r_{state}|f^r_{img})\right) \tag{A.2}$$

The RHS of (A.2) can be simplified as

$$\begin{aligned} &= \log\left(\frac{N}{k}\right) \sum_r p(f^r_{state}|f^r_{img}) - \sum_r p(f^r_{state}|f^r_{img}) \log[p(f^r_{state}|f^r_{img})] \\ &= \log\left(\frac{N}{k}\right) - \sum_r p(f^r_{state}|f^r_{img}) \log\left(p(f^r_{state}|f^r_{img})\right). \end{aligned} \tag{A.3}$$

Next we derive the equation (2.18).

$$
\begin{aligned}
H(f_{state}, f_{img}) &= -\sum_{\forall f_{img}} \sum_{\forall f_{state}} p(f_{state}, f_{img}) \log \big(p(f_{state}, f_{img})\big) \\
&= -\sum_{\forall f_{img}} \sum_{\forall f_{state}} p(f_{state}|f_{img}) p(f_{img}) \log \big(p(f_{state}|f_{img}) p(f_{img})\big) \\
&= -\sum_{\forall f_{img}} \sum_{\forall f_{state}} p(f_{state}|f_{img}) p(f_{img}) \{\log \big(p(f_{state}|f_{img})\big) + \log \big(p(f_{img})\big)\} \\
&= -\sum_{\forall f_{img}} [\sum_{\forall f_{state}} p(f_{state}|f_{img}) p(f_{img}) \log \big(p(f_{img})] - \sum_{\forall f_{img}} \sum_{\forall f_{state}} p(f_{state}|f_{img}) p(f_{img}) \{\log \big(p(f_{state}|f_{img})\big)\} \\
&= -\sum_{\forall f_{state}} p(f_{state}|f_{img}) \sum_{\forall f_{img}} p(f_{img}) \log \big(p(f_{img})] - \sum_{\forall f_{img}} p(f_{img}) \sum_{\forall f_{state}} p(f_{state}|f_{img}) \{\log \big(p(f_{state}|f_{img})\big)\} \\
&= \sum_{\forall f_{state}} p(f_{state}|f_{img}) H(f_{img}) + \sum_{\forall f_{img}} p(f_{img}) H(f_{state}|f_{img}) \\
&= H(f_{img}) + H(f_{state}|f_{img}). \qquad \text{(A.4)}
\end{aligned}
$$

Appendix B

Supplementary for Chapter 5

B.1 Proof of Proposition 2

We stated the relationship between the proposed Class Balanced loss (5.14) and the KL-divergence loss (Póczos and Schneider, 2011) in *proposition 2* in the main manuscript. Here we proof the proposition.

Proof: To proceed with the proof of the above mentioned proposition, we first simplify the expression of $D_{\alpha_s}^{CDA}(p_s||p_s^{aug})$ using Jensen's inequality (Menéndez et al., 1997), as follows,

$$\begin{aligned}
&\log \sum_{m=1}^{N_s} \frac{1}{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right) \geq \frac{1}{N_s} \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right), \\
&\text{or}, \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right) - \log(N_s) \geq \frac{1}{N_s} \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right), \\
&\text{or}, \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right) \geq \frac{1}{N_s} \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right) + \log(N_s), \\
&\text{or}, \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right) \geq \frac{1}{N_s} \log \sum_{m=1}^{N_s} \left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}} \right).
\end{aligned} \tag{B.1}$$

Next, we assume that the dataset under consideration is sufficiently large. In other words we assume $N_t \to \infty$. Under assumption, we have the following results,

$$\lim_{N_t \to \infty} \beta = \lim_{N_t \to \infty} \frac{N_t - 1}{N_t} = 1. \tag{B.2}$$

Using (B.2), we simplify the effective number for state class m, i.e. $E_{n_c}[m]$ as follows (note that we have dropped the index m as the results apply irrespective of any class under large

dataset assumption),

$$\begin{aligned}\lim_{N_t\to\infty} E_{n_s} &= \lim_{N_t\to\infty} \frac{1-\beta^{n_s}}{1-\beta},\\ &= \lim_{\beta\to 1} \frac{1-\beta^{n_s}}{1-\beta},\\ &= \lim_{\beta\to 1} n_s\beta^{n_s-1} = n_s.\end{aligned}$$

Next, using (B.1), we further simplify the expression of $D_{\alpha_s}^{CDA}(p_s||p_s^{aug})$ from (5.14) as follows,

$$\begin{aligned}D_{\alpha_s}^{CDA}(p_s||p_s^{aug}) &= \log \sum_{m=1}^{N_s} \frac{1}{N_s}\left(\frac{p_s[m]^{\frac{\alpha_s}{E_{n_s}[m]}}}{p_s^{aug}[m]^{\frac{\alpha_s}{E_{n_s}[m]}+1}}\right)\\ &\geq \frac{1}{N_s}\sum_{m=1}^{N_s} \log\left(\frac{p_s[m]^{\frac{\alpha_s}{n_s}}}{p_s^{aug}[m]^{\frac{\alpha_s}{n_s}+1}}\right),\\ &= \frac{\alpha_s}{N_s * n_s}\sum_{m=1}^{N_s} \log\left(\frac{p_s[m]}{p_s^{aug}[m]^{1+\frac{n_s}{\alpha_s}}}\right),\\ &= \frac{\alpha_s}{N_s * n_s}\left(\sum_{m=1}^{N_s} \log\left(\frac{p_s[m]}{p_s^{aug}[m]}\right) - \frac{n_s}{\alpha_s}\sum_{m=1}^{N_s}\log\left(p_s^{aug}[m]\right)\right),\\ &\geq \frac{\alpha_s}{N_s * n_s}\left(\sum_{m=1}^{N_s} \log\left(\frac{p_s[m]}{p_s^{aug}[m]}\right)\right),\\ &\geq \frac{\alpha_s}{N_s * n_s}\left(\sum_{m=1}^{N_s} p_s[m]\log\left(\frac{p_s[m]}{p_s^{aug}[m]}\right)\right),\\ &= \frac{\alpha_s}{N_s * n_s} KL\left(p_s||p_s^{aug}\right). \qquad \blacksquare\end{aligned} \tag{B.3}$$

B.2 Qualitative Image Classification Results

In addition to the qauntitative results discussed in Section 5.4.4, a few qualitative image classification results are shown in Fig. B.1. The first two rows represent the successful and unsuccessful image classification results from MIT-States. The third and fourth rows represent image classification results from C-GQA. Last two rows represent image classification results from UT-Zappos50k.

Proposed approach has been able to successfully classify some challenging compositions like *peeled orange* and *sliced bread*. *Peeled orange* is a difficult to classify composition as different images of *orange* may be *peeled* to different extent. This gives rise to intra-class variation in the visual features of the *peeled*.

UT-Zappos50k represents images of shoes where state represents the material of the shoe and the object represents the shoe type. Thus UT-Zappos50k poses a distinct set of challenges over MIT-States and C-GQA as it requires fine-grained recognition of state i.e. the material

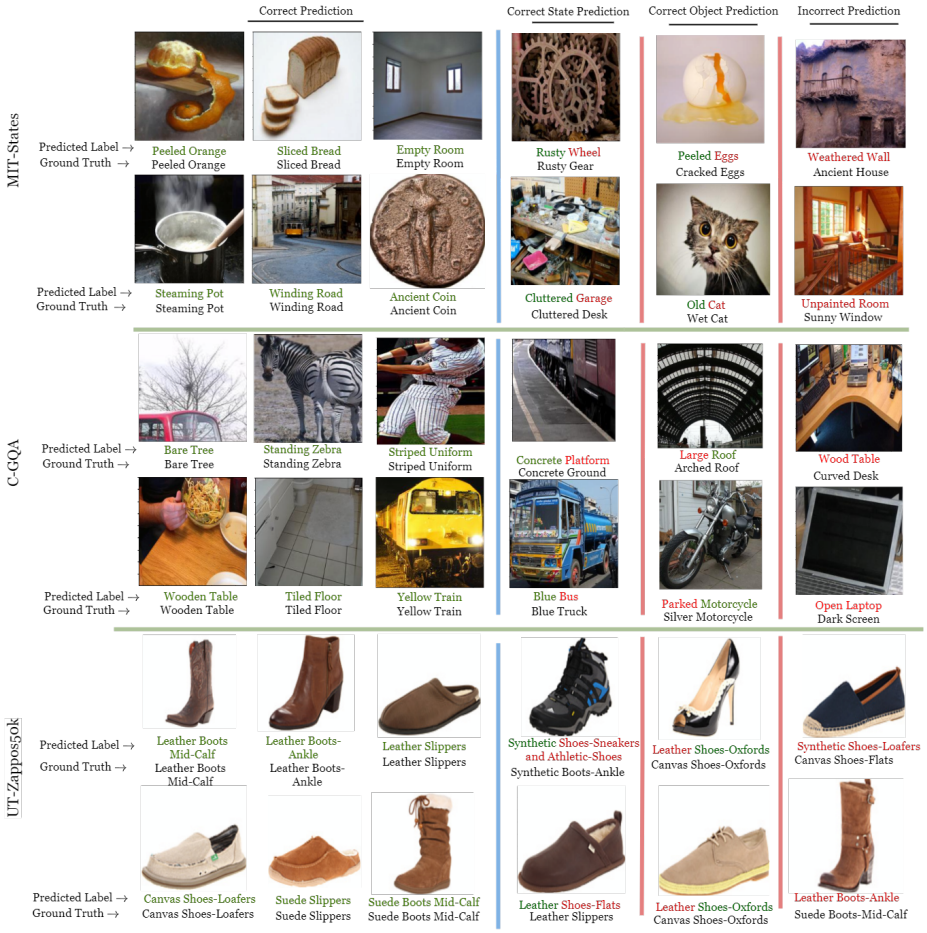


Figure B.1: Qualitative image classification results: The first three columns represent the successful image classification results. The fourth column represents the cases where the state has been successfully predicted but not the object. Fifth column represents the images with the incorrect state and correct object prediction. Final column represents images for which both state and object are incorrectly predicted. The text in green represents correctly predicted label. Similarly, text in red represents incorrectly predicted label.

of the shoe. Evidently proposed approach has successfully able to classify the images from UT-Zappos50k also.

We have also represented some failure cases of proposed approach in Fig. B.1. In image of *rusty gear*, although proposed approach successfully recognised the state *rusty*, it mistakenly classified the object *gear* as the object *wheel*. This is due to the similar visual appearance (like circular nature) of the objects *wheel* and *gear*. Similarly, for the image of the composition *curved table*, the proposed approach has predicted *wood table* which is also partially correct by observing the image. Hence we conclude that some of the labels of images of the CZSL datasets are ambiguous in nature and is often very challenging to uniquely identify the state-object composition.

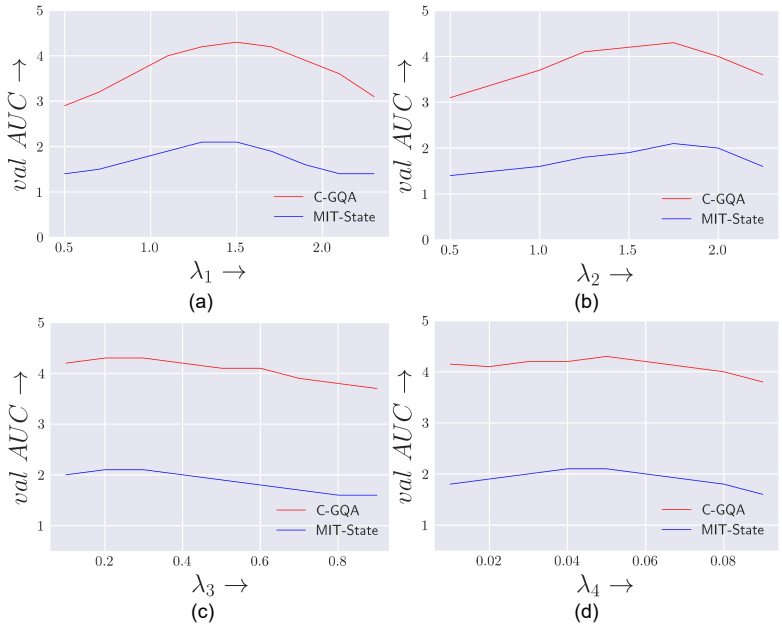


Figure B.2: Ablation over the variations in *val AUC* due to variation in λ_1, λ_2, λ_3 and λ_4 parameters as used in (5.25) on C-GQA and MIT-States datasets.

B.3 Additional Details Regarding Ablation of Loss Coefficients

The variations in *val AUC* due to change in λ_1, λ_2, λ_3 and λ_4 values of (5.25) are shown in Fig. B.2. In Fig. B.2(a) and (b), variation in λ_1 results in relatively higher variation in *val AUC* over λ_2. The parameter λ_1 controls the amount of the state and object cross entropy losses ($\mathcal{L}_s$, $\mathcal{L}_o$). $\mathcal{L}_s$ trains the network through invariant state probability p_s. p_s is obtained from the *invariant state features*. Similarly $\mathcal{L}_o$ trains the network through invariant object probability p_o. p_o is obtained from the *invariant object features*.

The loss coefficient λ_2 controls the amount of the context-guided state and object cross entropy losses ($\mathcal{L}_s^{cont}$ and $\mathcal{L}_o^{cont}$). The *invariant state features* and *invariant object features* are unique to that particular state and object. Thus the *invariant state and object features* are more useful to state and object recognition than the *context-guided state* and *object features*. Hence the network is more sensitive if the λ_1 value is changed from its optimum value in comparison to the λ_2 value.

B.4 Ablation Analysis on the Effect of the Parameters γ_s and γ_o

In (5.26), γ_s decides the association between the invariant state probability, p_s and the context-guided state probability $p_{s|o}$, in the final state-object probability. The visual features of state in the state-object composition have two constituent components, the *invariant state features* and the *context-guided state features*. The *context-guided state features* help to better process the variability of the state features due to context dependency in state features. Thus the *invariant state features* are more discriminative in nature. As shown in Fig. B.3, a higher value of γ_s, $\gamma_s = 0.95$ leads to better *val AUC*. This is to due to the fact that the higher

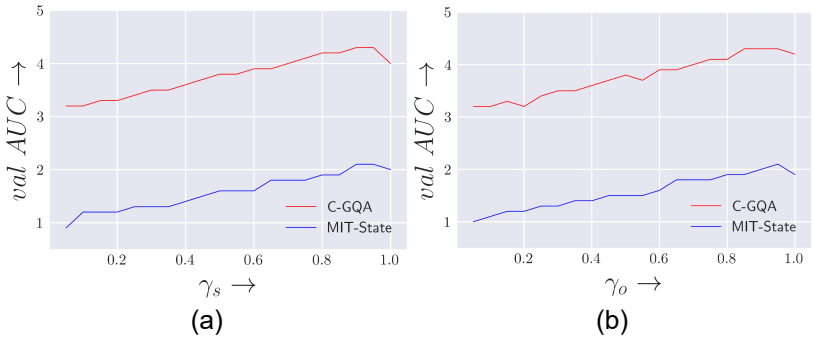


Figure B.3: Ablation over the γ_s and γ_o parameters as used in (5.26) and (5.27).

extent of presence of invariant state probability helps to better state recognition.

B.5 Additional Details Regarding Adversarial Training

In the LoPNA module of our proposed approach (see discussions in Section 5.3.2.2), we have incorporated adversarial perturbation in the features aggregated from the different neighbourhoods. Specifically, we add perturbation in the form of high frequency adversarial noise. The adversarial perturbation is added by sampling a noise $\eta \in [-\epsilon, \epsilon]$. Following existing adversarial training approaches (Bu et al., 2023, Bai et al., 2022), we train the model while maximizing the adversarial noise η within $[-\epsilon, \epsilon]$. We utilise Projected Gradient Descent (PGD) (Madry et al., 2017) to find the optimum perturbation by maximising η within $[-\epsilon, \epsilon]$.

Appendix C

Supplementary for Chapter 6

C.1 Diversity Preserving Prompt Learning

In this section we provide the derivation of the result used in Section 6.3.1.3 of Chapter 6. First, we start with derivation of result in (6.20) of Chapter 6.

$$\begin{aligned} MI(z_s; f_c) &= \int p(z_s, f_c) \log \frac{p(z_s, f_c)}{p(f_c)p(z_s)} dz_s df_c, \\ &= \int p(f_c) q_{\phi_s}(z_s|f_c) \log \frac{q_{\phi_s}(z_s|f_c)}{p(z_s)} dz_s df_c, \leq KL(q_{\phi_s}(z_s|f_c)||p(z_s)), [\text{Since } 0 \leq p(f_c) \leq 1]. \end{aligned} \tag{C.1}$$

Next, we discuss the steps involved to obtain the result in (6.21) of Chapter 6, Section 6.3.1.3.

$$\begin{aligned} MI(f_c; z_s|z_o) &= \int p(f_c, z_s, z_o) \log \frac{p(f_c, z_s|z_o)}{p(f_c|z_o)p(z_s|z_o)} dz_o dz_s df_c, \\ &= \int p(f_c, z_s, z_o) \log \frac{p(f_c, z_s, z_o)}{p(f_c|z_o)p(z_s, z_o)} dz_o dz_s df_c, \\ &= \int p(f_c, z_s, z_o) \log \frac{p_\theta(f_c|z_s, z_o)}{p(f_c|z_o)} dz_o dz_s df_c, \\ &= \int p(f_c, z_s, z_o) \log \frac{p_\theta(f_c|z_s, z_o)p(z_o)}{p(f_c, z_o)} dz_o dz_s df_c, \\ &= \int p(f_c, z_s, z_o) \log \frac{p_\theta(f_c|z_s, z_o)p(z_o)}{q_\phi(z_o|f_c)p(f_c)} dz_o dz_s df_c, \\ &= \int p(z_s, z_o|f_c)p(f_c) \log \frac{p_\theta(f_c|z_s, z_o)p(z_o)}{q_\phi(z_o|f_c)p(f_c)} dz_o dz_s df_c, \\ &= \int q_{\phi_s}(z_s|f_c) q_{\phi_o}(z_o|f_c) p(f_c) \log \frac{p_\theta(f_c|z_s, z_o)p(z_o)}{q_\phi(z_o|f_c)p(f_c)} dz_o dz_s df_c, \\ &= \int [-p(f_c) \log p(f_c)] df_c - \int q_{\phi_o}(z_o|f_c) \log \frac{q_{\phi_o}(z_o|f_c)}{p(z_o)} dz_o df_c \\ &+ \int q_{\phi_s}(z_s|f_c) q_{\phi_o}(z_o|f_c) \log p_\theta(f_c|z_s, z_o) dz_o dz_s, \end{aligned} \tag{C.2}$$

$$= H(f_c) + \mathbb{E}_{q_{\phi_s}(z_s|f_c)q_{\phi_o}(z_o|f_c)}\, p_\theta(f_c|z_s, z_o) - KL(q_\phi(z_o|f_c)||p(z_o)). \tag{C.3}$$

C.2 Additional Ablation Study

In addition to the Ablation Study reported in 6.4.4, we report experimental analysis regarding different components of our architecture and we also justify the choice of values for different parameters in our approach.

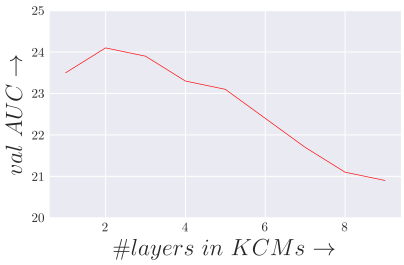


Figure C.1(a): Variation of *val AUC* with respect to variations in the number of MLPs in the KCM modules.

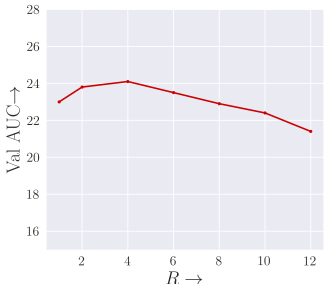


Figure C.1(b): Variation of *val AUC* with respect to variation in the number of patches cropped from the input image, R on MIT-States dataset.

Ablation study of the architecture of Knowledge Coupling Modules (KCMs): In our approach, all three KCMs (i.e. SKCM, OKCM and JKCM) are implemented using a Multi-Layer Perceptron (MLP), a Gaussian Error Linear Units (GELU), followed by another MLP. To evaluate the efficacy of the design choices in KCMs, we perform the following experiment. We vary the number of MLPs in the KCMs. The resulting *val AUC* achieved while training the model by different number of MLPs are noted and shown in Fig. C.1a.

From Fig. C.1a, the optimal *val AUC* is achieved for two layers of MLPs in KCMs (i.e. SKCM, OKCM and JKCM in our approach). By using more MLPs in the KCMs, number of trainable parameters in KCMs increases, leading to over-fitting of the model. Also, on decreasing the number of MLP layers from 2 to 1, the *val AUC* reduces as single MLP is ineffective in processing the cross-modal interactions. Also it may be noted that the results achieved by MLP-based current implementation and MhSA-based implementation (Vaswani et al., 2017) of KCMs are comparable. However, we avoid MhSA due to quadratic computational complexity of MhSAs (Yuan et al., 2021a).

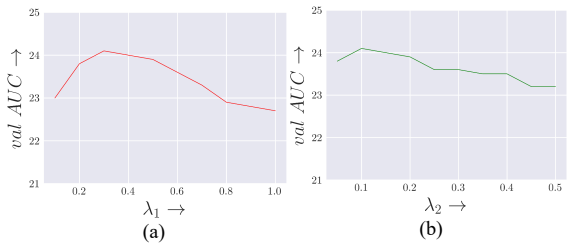


Figure C.2: Variation of *val AUC* with respect to variations of λ_1 and λ_2 in (6.36).

Ablation study on the number of cropped images: As described in the sub-section 'Scale-

aware Feature Extraction' (under Section 6.3.1.4 in Chapter 6), to extract the scale-aware image features, we first randomly crop multiple patches at different resolutions from the input image. Let the number of patches as cropped from the input image be R. Next, the cropped patches are up-scaled to the same resolution as the input image and passed to the *image feature extractor*. The cropped patches are utilised to obtain multi-scale features from the input image.

The results in Fig. C.1b show that the results of our approach is optimum for value of $R = 4$ and decreases further if R is decreased or increased. As the R is increased, too many cropped patches are obtained from the input image. Thus there is higher possibility of obtaining overlapping patches from the input image. Due to the redundant information, the model tends to over-fit to the training data. If a smaller number of cropped patches are obtained, then insufficient multi-scale information is extracted.

Ablation of the loss coefficients: The variations of *val AUC* with respect to variations in λ_1 and λ_2 in (6.36) of the main manuscript are shown in Fig. C.2. The optimum value of the *val AUC* is obtained around value of 0.3 for λ_1 and around 0.1 for λ_2. The losses $\mathcal{L}_s^{sc}$ and $\mathcal{L}_o^{sc}$ attempt to process the scale-dependent features. Thus $\mathcal{L}_s^{sc}$ and $\mathcal{L}_o^{sc}$ help in understanding the context dependency between visual features of state and object. The stronger sensitivity of *val AUC* to variation in λ_1 is due to the aforementioned justification. Similar results were also observed for C-GQA and UT-Zappos50k.

Dataset →	MIT-States	C-GQA	UT-Zappos50k
Embedding strategy ↓	*val AUC*	*val AUC*	*val AUC*
FastText	6.6	2.5	30.4
Word2Vec	6.8	2.6	30.7
GloVe	**7.6**	**3.4**	**33.8**

Table C.1: Experiment to study the effectiveness of embedding strategies in our approach, in OW-CZSL.

C.2.1 Ablation Study of Embedding Strategies on the Open-world CZSL

Following prior approaches (Lu et al., 2023, Mancini et al., 2022), we include a feasibility-based filtering approach in OW-CZSL evaluation protocol. The feasibility-based filtering approach is utilised to eliminate infeasible compositions from the prediction of the model. In a CZSL dataset, with N_s number of states and N_o number of objects, the number of possible predictions in OW-CZSL is $N_s \times N_o$ (Mancini et al., 2021). For each unseen state-object compositions, $c_u = (s', o')$ in the OW-CZSL, we evaluate the feasibility of the composition due to the presence of object, as well as due to the presence of state. The detailed process is described in the main paper in Section 6.3.1.7, under 'Inference Strategy' title. Next, we briefly describe our approach to evaluate the feasibility of the object to be present in the unseen state-object compositions, in the OW-CZSL.

The maximum *cosine similarity* between the word embedding (Pennington et al., 2014) of the object label, o' and word embedding (Pennington et al., 2014) of all other objects that form feasible compositions with the state s', is evaluated and represented as $f_{c_u}^{o'}$. Similarly, the feasibility of c_u from state perspective, $f_{c_u}^{s'}$ is evaluated as follows. The maximum *cosine similarity* between the word embedding of the state s' in the composition and word embedding of all other the states that form feasible compositions with the object in the current

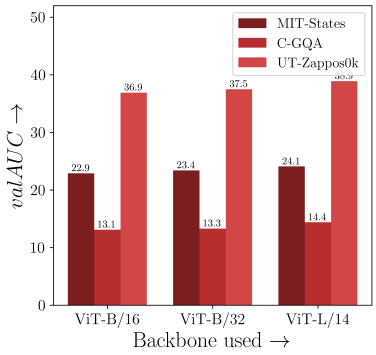


Figure C.3: Ablation study on effect of different backbones as used in proposed approach.

composition, o' is evaluated. The final feasibility of composition c_u, i.e. f_{c_u} is evaluated as, $f_{c_u} = (f^{o'}_{c_u} + f^{s'}_{c_u})/2$. Next, the composition, c_u is considered as a possible outcome of the model in OW-CZSL if $f_{c_u} \geq \theta$. Here θ is a user-defined threshold.

Next, we evaluate the effectiveness of the embedding strategy used in our approach. Specifically we evaluate the *val AUC* while using the GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) word embeddings. The results are reported in Table. C.1. Evidently, GloVe reports the best results in the current context. FastText utilises *n-gram* approach and is specially suitable at out-of-vocabulary word. The CZSL datasets consist of commonly used words as state and object text labels. So, FastText is not able to provide any additional advantage in the context of CZSL. So we propose that the GloVe embedding is better suited for our current approach.

C.2.2 Analysis of the Backbone Network

The *feature extractors* in our approach are based on the ViT (Vaswani et al., 2017). In our approach the 'ViT-L/14' backbone is used. The 'L' in 'ViT-L/14' represents the 'large' variant of ViT with more number layers of *transformer encoders* (Vaswani et al., 2017) and the number 14 represents that the *transformer encoder* splits the input image into 14×14 dimensional patches. In this section we also evaluate the performance of two other backbones for CLIP i.e. 'ViT-B/16' and 'ViT-B/32'. The *val AUC* of our approach on the closed-world CZSL evaluation protocol are shown in Fig. C.3. Evidently the 'ViT-L/14' backbone achieves highest *val AUC* across all three datasets. The better performance of 'ViT-L/14' backbone can be attributed to its higher number of *transformer encoders*, in comparison to 'ViT-B/16' and 'ViT-B/32' backbones. Higher layers (i.e. higher number of *transformer encoders*) in 'ViT-L/14' backbone help the proposed approach to better adapt the contextual dependency in CZSL through the GBs and KCMs.

Analysis of using CLIP-Adaptor in our approach: We incorporate a set of trainable tokens in the intermediate stages of the *feature extractors* (*text* and *image*) of the proposed approach. In our approach, the learnable tokens to be inserted into the intermediate stages of *image feature extractor* are functionally dependent on the outputs from the intermediate stages of *text feature extractor* and vice versa (for detailed discussion, see Section 6.3.1.2).

Recently, in an attempt to adapt the CLIP model to downstream tasks, Gao *et al.*proposed an approach termed CLIP-Adapter (Gao et al., 2024). In the CLIP-Adapter based approach,

Datasets →	MIT-States	C-GQA
Approach ↓	*val AUC*	*val AUC*
Our Approach with Adapter	21.9	13.0
Our Approach	**24.1**	**14.4**

Table C.2: Experiment to study the effectiveness of CLIP-Adapter (Gao et al., 2024) in our approach in closed-World CZSL.

a neural network module termed *adapter* is appended at the end of the *feature extractors* (*text* and *image*). The parameters of the *feature extractors* are kept frozen during training, whereas the parameters of the *adapter* are updated in the training stage.

Regarding implementation of the *adapter*, we follow the original implementation by (Gao et al., 2024) as described next. The *adapter* is implemented using a MLP, a ReLU layer followed by another MLP and finally another ReLU layer. The results of the corresponding experiments are shown in Table C.2.

The results in Table C.2 report that our approach outperforms the *adapter* based approach.The improvement of our approach over the *adapter* can be justified as follows. The *adapter* fine-tunes a small number of additional neural network layers only at the end of *text* and *image feature extractors*. Our approach incorporates the learnable tokens in the intermediate stages of *text* and *image feature extractors*. Using the KCMs and GBs, our approach is specifically designed for the context dependency challenge and the disentanglement challenge in CZSL (see Section 6.3.1.2 for detailed discussion on these challenges of CZSL). Hence, our approach is more effective for the CZSL problem than the *adapter* based approach.

List of Publications

List of Publications:

Journal papers:

1. Aditya Panda and Dipti Prasad Mukherjee, "Knowledge guided transformer network for compositional zero-shot learning". ACM Transactions on Multimedia Computing Communications and Applications, 2024, doi: 10.1145/3687129.

2. Aditya Panda and Dipti Prasad Mukherjee, "Compositional zero-shot learning using multi-branch graph convolution and cross-layer knowledge sharing". Pattern Recognition, 145 (2024): 109916, doi: 10.1016/j.patcog.2023.109916.

3. Aditya Panda, Bikash Santra and Dipti Prasad Mukherjee, "Isolating features of object and its state for compositional zero-shot learning". IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 5, pp. 1571-1583, Oct. 2023, doi: 10.1109/TETCI.2022.3232816.

Conference papers:

1. Aditya Panda, Bikash Santra and Dipti Prasad Mukherjee, "Bi-modal compositional network for feature disentanglement". In 2022 IEEE International Conference on Image Processing (ICIP), pp. 3051–3055. IEEE, 2022, doi: 10.1109/ICIP46576.2022.9897457.

List of works under review:

1. Aditya Panda and Dipti Prasad Mukherjee, "Prompt-driven multi-branch disentanglement network for compositional zero-shot learning", Submitted to IEEE Transactions on Pattern and Machine Intelligence, 2024.

2. Aditya Panda and Dipti Prasad Mukherjee, "Partially Supervised Unseen State-object Image Classification by Discriminative Context Aggregation", Submitted to IEEE Transactions on Image Processing, 2025.

List of publication not included in the thesis

1. Aditya Panda and Dipti Prasad Mukherjee. "Monocular 3d human pose estimation by multiple hypothesis prediction and joint angle supervision". In 2021 IEEE International Conference on Image Processing (ICIP), pp. 3243–3247. IEEE, 2021

References

S. Alfasly, C. K. Chui, Q. Jiang, J. Lu, and C. Xu. An effective video transformer with synchronized spatiotemporal and spatial self-attention for action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 83

K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences*, 117(47):29302–29310, 2020. 1

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017. 14

Y. Atzmon and G. Chechik. Probabilistic and-or attribute grouping for zero-shot learning. *arXiv preprint arXiv:1806.02664*, 2018. 14

Y. Atzmon, J. Berant, V. Kezami, A. Globerson, and G. Chechik. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*, 2016. 1

Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik. A causal view of compositional zero-shot recognition. In *Advances in Neural Information Processing Systems*, volume 33, pages 1462–1473, 2020a. 14, 25, 27, 45, 57

Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 33:1462–1473, 2020b. 70, 116

J. Bai, L. Yuan, S.-T. Xia, S. Yan, Z. Li, and W. Liu. Improving vision transformers by revisiting high-frequency components. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 86, 87, 129

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 10, 112

Y. Bengio and J.-S. Senécal. Quick training of probabilistic neural nets by importance sampling. In *International Workshop on Artificial Intelligence and Statistics*, pages 17–24. PMLR, 2003. 19

D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a. 79, 82, 89

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019b. 79, 82, 89

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017. 13, 17, 24, 44, 74, 99, 133

S. Bose, A. Jha, E. Fini, M. Singha, E. Ricci, and B. Banerjee. Stylip: Multi-scale style-conditioned prompt learning for clip-based domain generalization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5542–5552, 2024. 78

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 35, 37

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. In *Conference on Learning Representations, ICLR 2014—Conference Track Proceedings*, 2014. 35

Q. Bu, D. Huang, and H. Cui. Towards building more robust models with frequency bias. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 4402–4411, 2023. 86, 129

C. Cai and Y. Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020. 52

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 77, 83

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 *European Conference on Computer Vision*, pages 52–68. Springer, 2016. 25

Cover, Thomas M. *Elements of information theory*. John Wiley & Sons, 1999. 21, 123

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, pages 702–703, 2020. 87

Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 88

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 29, 2016. 35

M. M. Derakhshani, E. Sanchez, A. Bulat, V. G. T. da Costa, C. G. Snoek, G. Tzimiropoulos, and B. Martinez. Bayesian prompt learning for image-language model generalization. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 15237–15246, 2023. 100, 103

K. Do and T. Tran. Theory and evaluation metrics for learning disentangled representations. In *International Conference on Learning Representations*, 2019. 6, 21

P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014. 80

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 55, 56, 58, 59, 60, 63, 64, 65, 75, 77, 83

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. URL https://doi.org/10.1080/01969727308546046. 85

Eastwood, Cian and Williams, Christopher KI. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. 6, 21

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *Advances in Neural Information Processing Systems*, 2009. 1

J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, and H. Lu. Scene segmentation with dual relation-aware attention network. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6): 2547–2560, 2020. 83

K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989. 62

P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024. ix, 133, 134

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 80, 81

C. Gong, D. Wang, and Q. Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13683–13692, 2021. 79, 82, 89

B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021. 63

C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan. Augfpn: Improving multi-scale feature learning for object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12595–12604, 2020. 80, 81

K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 63

S. Hao, K. Han, and K.-Y. K. Wong. Learning attention as disentangler for compositional zero-shot learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15315–15324, 2023. 100, 105

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1026–1034, 2015. 44

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 24, 38, 44, 55, 80, 94, 99, 115

D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 114

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 62

C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced

classification. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5375–5384, 2016. 88

S. Huang, B. Gong, Y. Feng, Y. Lv, and D. Wang. Troika: Multi-path cross-modal traction for compositional zero-shot learning. *arXiv preprint arXiv:2303.15230*, 2023. 10, 99, 100, 101, 102, 104, 108, 114, 115, 118

F. Huo, W. Xu, S. Guo, J. Guo, H. Wang, Z. Liu, and X. Lu. Procc: Progressive cross-primitive compatibility for open-world compositional zero-shot learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12689–12697, Mar. 2024. doi: 10.1609/aaai.v38i11.29164. URL https://ojs.aaai.org/index.php/AAAI/article/view/29164. 8, 80, 92, 94

D. Huynh and E. Elhamifar. Interactive multi-label cnn learning with partial labels. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9423–9432, 2020. 87

H. Hwang, G.-H. Kim, S. Hong, and K.-E. Kim. Variational interaction information maximization for cross-domain disentanglement. *Advances in Neural Information Processing Systems*, 33:22479–22491, 2020. 109, 110

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 38, 44

P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1383–1391, 2015. 9, 23, 30, 34, 40, 44, 45, 50, 68, 70, 79, 93, 116, 122

C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 99, 102, 122

S. Karthik, M. Mancini, and Z. Akata. Revisiting visual product for compositional zero-shot learning. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 3, 77, 82

S. Karthik, M. Mancini, and Z. Akata. Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2022. 1, 6, 8, 14, 68, 78, 80, 83, 87, 92, 94, 114

M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan. Maple: Multi-modal prompt learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 100, 102, 103, 106, 108, 118

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (Poster)*, 2015. 24, 44, 68, 93, 114

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 108

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 14, 33, 35, 37, 58

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 1

H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. 1

D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, International Conference on Machine Learning*, 3(2):896, 2013. 81

N. Li, Y. Chen, W. Li, Z. Ding, D. Zhao, and S. Nie. Bvit: Broad attention-based vision transformer. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a. 83

Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018. 52

X. Li, X. Yang, K. Wei, C. Deng, and M. Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9335, 2022. 6, 58, 68, 78, 83, 114

Y. Li, Z. Liu, S. Jha, and L. Yao. Distilled reverse attention network for open-world compositional zero-shot learning. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 1782–1791, 2023b. 57, 68, 92, 94, 114

Y. Li, Z. Liu, H. Chen, and L. Yao. Context-based and diversity-driven specificity in compositional zero-shot learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17037–17046, 2024. 10, 102

Y.-L. Li, Y. Xu, X. Mao, and C. Lu. Symmetry and group in attribute-object compositions. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11316–11325, 2020. 14, 19, 20, 24, 25, 26, 45, 54, 55, 61, 68, 92, 94, 114

Y.-L. Li, Y. Xu, X. Xu, X. Mao, and C. Lu. Learning single/multi-attribute of object with symmetry and group. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9043–9055, 2021. 20

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 80, 81, 84

Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 77, 81, 83, 92, 94

Z. Liu, Y. Li, L. Yao, X. Chang, W. Fang, X. Wu, and A. El Saddik. Simple primitives with feasibility-and contextuality-dependence for open-world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 68, 92

A. Liutkus, O. Cıfka, S.-L. Wu, U. Simsekli, Y.-H. Yang, and G. Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pages 7067–7079. PMLR, 2021. 85

M. Loog and A. C. Jensen. Semi-supervised nearest mean classification through a constrained log-likelihood. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):995–1006, 2014. 81

X. Lu, S. Guo, Z. Liu, and J. Guo. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *Proceedings of IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, pages 23560–23569, 2023. 9, 99, 100, 101, 102, 104, 110, 113, 114, 118, 132

Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian. Prompt distribution learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 103

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 129

X. Man, D. Ouyang, X. Li, J. Song, and J. Shao. Scenario-aware recurrent transformer for goal-directed video captioning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(4):1–17, 2022. 58

M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata. Open world compositional zero-shot learning. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5222–5230, 2021. 4, 7, 14, 35, 36, 55, 57, 68, 74, 78, 79, 92, 114, 132

M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata. Learning graph embeddings for open world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 92, 113, 114, 132

X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, and H. Xue. Towards robust vision transformer. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 60

G. F. Marcus. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press, 2003. 1

M. Menéndez, J. Pardo, L. Pardo, and M. Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. 125

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 56, 60, 68, 74, 99, 133

I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1792–1801, 2017. 6, 7, 13, 24, 33, 45, 55, 61, 68, 74, 78, 92, 114, 120

F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017. 35, 37

M. F. Naeem, Y. Xian, F. Tombari, and Z. Akata. Learning graph embeddings for compositional zero-shot learning. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 953–962, 2021. 6, 7, 9, 14, 23, 24, 25, 27, 34, 35, 39, 40, 44, 45, 46, 54, 55, 58, 68, 69, 70, 74, 78, 79, 80, 92, 93, 114, 122

T. Nagarajan and K. Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *European Conference on Computer Vision*, pages 169–185, 2018. 6, 7, 9, 13, 14, 20, 24, 27, 33, 45, 55, 61, 68, 74, 80, 92, 114

Z. Nan, Y. Liu, N. Zheng, and S.-C. Zhu. Recognizing unseen attribute-object pair with generative model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8811–8818, 2019. 1, 7, 14, 57, 61, 88, 99

N. V. Nayak, P. Yu, and S. Bach. Learning to compose soft prompts for compositional zero-shot learning. In *The Eleventh International Conference on Learning Representations*, 2022. 6, 9, 99, 100, 101, 102, 104, 110, 114, 118

T. Nguyen, M. Raghu, and S. Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020. 66

H. J. Nussbaumer and H. J. Nussbaumer. *The fast Fourier transform*. Springer, 1982. 86

K. Oono and T. Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. 52

W. Ouyang, X. Wang, C. Zhang, and X. Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 864–873, 2016. 88

M. M. Palatucci, D. A. Pomerleau, G. E. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. *Advances in Neural Information Processing Systems*, 2009. 1

A. Panda and D. P. Mukherjee. Compositional zero-shot learning using multi-branch graph convolution and cross-layer knowledge sharing. *Pattern Recognition*, page 109916, 2023. 10

A. Panda and D. P. Mukherjee. Compositional zero-shot learning using multi-branch graph convolution and cross-layer knowledge sharing. *Pattern Recognition*, 145:109916, 2024a. 7, 10

A. Panda and D. P. Mukherjee. Knowledge guided transformer network for compositional zero-shot learning. *ACM Transactions on Multimedia Computing Communications and Applications*, 2024b. 7, 11

A. Panda and D. P. Mukherjee. Prompt-driven multi-branch disentanglement network for compositional zero-shot learning. *Submitted to IEEE Transactions on Pattern and Machine Intelligence*, 2024c. 9, 11

A. Panda, B. Santra, and D. P. Mukherjee. Bi-modal compositional network for feature disentanglement. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3051–3055. IEEE, 2022. 10, 88, 99

A. Panda, B. Santra, and D. P. Mukherjee. Isolating features of object and its state for compositional zero-shot learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023. 5, 12, 99

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 93, 114

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *2014 Conf. on Empirical Methods in Natural Language Proceedings (EMNLP)*, pages 1532–1543, 2014. 13, 14, 17, 56, 74, 99, 113, 132, 133

J. Pennington, S. Schoenholz, and S. Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in Neural Information Processing Systems*, 30, 2017. 37

F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019. 102

B. Póczos and J. Schneider. On the estimation of alpha-divergences. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 609–617. JMLR Workshop and Conference Proceedings, 2011. 78, 82, 87, 89, 96, 125

S. Purushwalkam, M. Nickel, A. Gupta, and M. Ranzato. Task-driven modular networks for zero-shot compositional learning. In *in Proceedings of the IEEE/CVF International*

Conference on Computer Vision, pages 3593–3602, 2019. 6, 9, 14, 15, 17, 19, 24, 25, 31, 40, 45, 46, 55, 58, 61, 68, 74, 80, 92, 114, 120

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 9, 99, 100, 101, 102, 105, 106, 114, 117, 122

M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. Do vision transformers see like convolutional neural networks? *in Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 66

A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961. 78, 79, 82, 87

E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, 2003. 81

H. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. *Advances in Neural Information Processing Systems*, 8, 1995. 80

F. Ruis, G. Burghouts, and D. Bucur. Independent prototype propagation for zero-shot compositionality. *Advances in Neural Information Processing Systems*, 34:10641–10653, 2021. 7, 35, 57

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. *IJCV*, 115(3):211–252, 2015. 24, 27, 44, 47, 48

N. Saini, K. Pham, and A. Shrivastava. Disentangling visual embeddings for attributes and objects. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13658–13667, 2022. 6, 57, 78, 83, 99

R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR 2011*, pages 1481–1488. IEEE, 2011. 2

A. Sanborn and T. Griffiths. Markov chain monte carlo with people. *in Advances in Neural Information Processing Systems*, 20, 2007. 67

T. L. Scao and A. M. Rush. How many data points is a prompt worth? *arXiv preprint arXiv:2103.08493*, 2021. 102

K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. 79, 82, 89, 97

R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI*, pages 4444–4451, 2017. 8, 14, 74, 80, 94

M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81:73–205, 1995. 112

Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019. 62

R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. *IEE Proceedings -Vision, Image and Signal Processing*, 141(4):245–250, 1994. 80

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 58, 64, 83, 84, 99, 102, 104, 115, 131, 133

B. Wang, R. Ji, L. Zhang, and Y. Wu. Bridging multi-scale context-aware representation for object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022a. 78, 81, 83, 86, 97

H. Wang, M. Yang, K. Wei, and C. Deng. Hierarchical prompt learning for compositional zero-shot recognition. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023a. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/163. URL https://doi.org/10.24963/ijcai.2023/163. 10, 99, 101, 102, 114, 116, 118

J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin. Carafe: Content-aware reassembly of features. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 3007–3016, 2019a. 84

P. Wang, W. Zheng, T. Chen, and Z. Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022b. 86

Q. Wang, L. Liu, C. Jing, H. Chen, G. Liang, P. Wang, and C. Shen. Learning conditional attributes for compositional zero-shot learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2023b. 57, 68

X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez. Tafe-net: Task-aware feature embeddings for low shot learning. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, June 2019b. 14

X. Wang, S. Zhang, Z. Yu, L. Feng, and W. Zhang. Scale-equalizing pyramid convolution for object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13359–13368, 2020. 80, 81, 84

Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019c. 86, 87

Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2022c. 97

Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. In *31st Advances in Neural Information Processing Systems*, volume 30, pages 7032–7042, 2017. 2, 88

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 102

K. Wei, M. Yang, H. Wang, C. Deng, and X. Liu. Adversarial fine-grained composition learning for unseen attribute-object recognition. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3741–3749, 2019. 14

Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017. 1

G. Xu, P. Kordjamshidi, and J. Chai. Prompting large pre-trained vision-language models for compositional concept learning. *arXiv preprint arXiv:2211.05077*, 2022. 9, 100, 102

Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021a. 97

Z. Xu, G. Wang, Y. Wong, and M. S. Kankanhalli. Relation-aware compositional zero-shot learning for attribute-object pair recognition. *IEEE Transactions on Multimedia*, 2021b. 1, 6, 14, 25, 35, 61, 78, 88, 94

L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7236–7246, 2023. 97

M. Yang, C. Deng, J. Yan, X. Liu, and D. Tao. Learning unseen concepts via hierarchical decomposition and composition. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10248–10256, 2020. 57, 99

M. Yang, C. Xu, A. Wu, and C. Deng. A decomposable causal view of compositional zero-shot learning. *IEEE Transactions on Multimedia*, 2022. 57

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 81

A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 9, 23, 24, 25, 34, 44, 68, 79, 93, 122

A. Yu and K. Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 5570–5579, 2017. 9, 23, 24, 25, 34, 44, 68, 79, 93, 122

J. Yu, J. Yao, J. Zhang, Z. Yu, and D. Tao. Sprnet: single-pixel reconstruction for one-stage instance segmentation. *IEEE Transactions on Cybernetics*, 51(4):1731–1742, 2020. 84

K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021a. 63, 131

L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021b. 55, 56, 57, 58, 63, 64, 68, 75, 77, 83

B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11304–11314, 2022a. 58

H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 86, 87

J. Zhang, Z. Fang, H. Sun, and Z. Wang. Adaptive semantic-enhanced transformer for image captioning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022b. 83

T. Zhang, K. Liang, R. Du, X. Sun, Z. Ma, and J. Guo. Learning invariant visual representations for compositional zero-shot learning. In *European Conference on Computer Vision*, pages 339–355. Springer, 2022c. 100

K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Conditional prompt learning for vision-language models. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022a. 100, 102, 103, 106, 108

K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b. 100, 102, 103, 106, 108