Pattern Recognition Letters
journal homepage: www.elsevier.com

Multi-Stage Semantic Graph Embeddings for Compositional Zero-Shot Learning

Hitesh Kandala[a,], Ruchika Chavhan[b], Ushasi Chaudhuri[c], Biplab Banerjee[a]

[a] *Indian Institute of Technology Bombay*
[b] *University of Edinburgh*
[c] *University of Illinois Urbana Champaign*

ABSTRACT

We tackle the problem of compositional zero-shot learning (CZSL) where the task is to recognize novel composite semantic concepts (like red-tomato, wet-dog) consisting of states (red, wet) and objects (tomato, dog) characterizing the visual primitives. CZSL is a complex task given the fact that similar states may appear to be visually distinct, thus hampering the recognition performance. Recently, graph convolution networks (GCN) have been used to learn a compatibility function among the visual, state, and object embeddings. However, we postulate that these techniques do not fully explore the compositional nature of the semantic space where states, objects, and pairs follow different neighborhood topologies. To this end, we introduce the concept of multi-stage graph embeddings where separate GCNs are used to initially model the pairwise interactions for the states, objects, and the composition label embeddings, respectively. A composite GCN subsequently combines this information to learn a discriminative and neighborhood preserving latent semantic space ensuring the strong coupling of a composition label with its respective state and object. Further, we use a vision transformer for projecting the images onto the same latent space where the cross-modal information can be compared. An adaptive margin based cross-entropy loss is introduced to train the model end-to-end while ensuring enriched discrimination among the categories. Results on the MIT-States, UT-Zappos and C-GQA datasets confirm the superiority of the proposed approach.

Keywords: Image Classification; Compositional Zero-Shot Learning; Deep Learning; Graph Convolutional Networks;

1. Introduction

The machine learning and computer vision community have predominantly directed its focus to object detection and description, by providing human-intelligible interpretation using different modalities like text and speech. In the same spirit, Compositional Learning is a problem that describes objects in an image in conjunction with adjectives referred to as *states*. Learning the semantic meaning behind objects/states from the composition and associate them with novel and unobserved states/objects is indispensably an important aspect of human intelligence. Humans can effortlessly associate the words `red tomato` if they have prior knowledge of the state `red` and the object `tomato`. Thus, generalizing to unseen compositions by learning abstract semantic concepts of seen compositions and visual features is a rapidly emerging field in artificial intelligence, resulting in the initiation of the new problem definition of compositional zero-shot learning (CZSL) (1; 2).

Given a training set consisting of state-object compositions, we aim to learn a mapping between visual features and compositions and further predict unseen compositions of these states and objects at test time. Due to a vast number of possible unseen compositions, it is important to notice the existence of unreal and invalid compositions called *distractors* (eg. `bright salad`, `closed dog`), and encourage the learner against predicting these distractor combinations. Therefore, learning a semantically correct and globally valid mapping is of utmost importance in CZSL. Early works in CZSL (1; 2; 3; 4) have treated objects and states as independent entities on different manifolds to learn topological information using transformation functions. (5) proposes to predict unseen compositions at test time while

e-mail: `khitesh2000@gmail.com` (Hitesh Kandala)

designing loss functions to discourage the models from generating distractor compositions explicitly. Despite designing models that learn a rich shared latent space using visual features and textual attributes, they tend to disregard the rich semantic dependency of states, objects, and compositions. Humans instinctively learn dependencies between objects and states and translate this knowledge to describe objects using novel states. However, previous works in CZSL fail to replicate these pathways of learning as they tend to disentangle the state and object feature learning.

Recently, (6) proposes a model that learns the dependency relationship among states, objects, and compositions by constructing a single compositional graph learned using visual and textual embeddings. By definition, the compositional graph establishes relationships among pairs of state-object, state-composition, object-composition but neglects to learn an association between two compositions. But, it is possible for two distinct composition (for example, `small cat` and `cute kitten`) to be highly related. Thus, ignoring this crucial relationship between distinct compositions causes incomplete semantic relationships among all possible compositions. Furthermore, we argue that forming a composition for a visual feature is, in fact, hierarchically choosing the state and object by establishing semantic associations among them. To form a valid composition, one first needs to learn the meaning behind states and objects individually and then relate them with visual features. Therefore, we aim to utilize neighbourhood information learned from states and objects as 'prior knowledge' to form meaningful compositions.

In order to ensure a meaningful and discriminative latent space where the relevance between the state-object pairs is preserved, we propose a multi-stage semantic graph embedding model that enhances the latent compositional space by learning semantic relationships among all the states, objects, and pairs separately and then combining this information into a meaningful and efficient compositional graph. The induction of knowledge from states and object facilitates the prediction of semantically aware compositions. We demonstrate that the multi-stage graph embedding model preserves neighbourhood topologies and inherently predicts fewer distractor compositions. Furthermore, we employ vision-based transformers to project images onto a shared latent space where the cross-modal information can be used to predict compositions. Finally, we consider an adaptive feature margin-based softmax formulation to define the loss function, which maximizes the compatibility margin between the visual and fine-grained compositions. We summarize our significant contributions as: i) We introduce a novel multi-stage graph embedding network to solve the problem of CZSL, based on exploiting dependency relationships among states, objects, and compositions separately, further to be combined with learning a shared latent space of compositions. ii) We introduce a novel adaptive margin based softmax formulation to define the compatibility loss function between the visual and label embeddings. iii) Extensive experiments are performed on all the benchmark CZSL datasets. In particular, a significant improvement $5 - 24\%$ in the AUC score on MIT states, UT Zappos, and C-GQA datasets can be observed.

2. Related Works

Compositional learning: Compositionality is a crucial feature of human intelligence, which involves combining simple concepts logically to create higher-level concepts. Early works in this field revolved around learning robust object representations by forming low-level compositions of local image features and learning relations between them to build higher-level compositions subsequently. In this context, (7) described the higher level compositions by probability distributions over coupled local object parts given a categorization. Similarly, (8) introduced an incremental design that can learn flexible compositions in an unsupervised manner while being indifferent to newly observed categories, and (9) proposed a multi-modal recursive compositional model that considers a specific object and different viewpoints as a single composition. While these early research papers used compositions of only visual features, the notion of compositional learning has also been utilized recently in several other cross-modal tasks (10).

CZSL: As mentioned earlier, CZSL aims to recognize unseen compositions of states and objects at test time. Given an image, a learner is trained to generate a composition by assessing similarities between visual features and word embeddings (1; 5). Moreover, (5) considered an open-world setting where the model encounters significantly fewer compositions while training as compared to testing and includes loss functions that discourage the prediction of distractor compositions. Earlier, (11) trained an SVM in the compositional space to create a tensor of margin weights for each pair of seen compositions and used Bayesian probabilistic tensor factorization for unseen compositions. (3) treated the objects as vectors and states as operators that modify the visual meaning behind the composition of state and object. (2) trained fully-connected layers to produce compatible features which are selectively activated by a gating function, taking as input an object-state composition. In (12), CZSL has been presented with a causal perspective to tackle generalizing problems in long-tailed visual distributions by deducing the features associated with states and objects that caused the image. Very recently. (6; 13) introduced a graph based formulation to learn improved composition embeddings.

Our work is closely related to (6) in the sense that we use graphs to generate a latent compositional space. However, we would like to point out that (6) considered the states, objects, and compositions to be nodes of a single graph where the node features are obtained from a pre-trained word embedding space. This paradigm leads to partial exploration of the hierarchical structure of states acting on objects to change visual features. To combat this issue and to fully explore the neighborhood structure of the three spaces (states, objects, and pairs), we introduce the notion of multi-stage GCNs to capture the essence of states and objects and map them onto a more efficient latent space.

3. Methodology

Preliminaries: Formally, let $\mathcal{D}^{\text{train}} = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in C_s\}_{i=1}^{n}$, where $\mathcal{D}^{\text{train}}$ denotes the training set, $\mathcal{X}$ denotes the input data, and C_s denotes the corresponding label space of seen compositions. Every label is a tuple of states and objects $y = (s, o) : s \in \mathcal{S}, o \in \mathcal{O}$, where $\mathcal{S}$ and $\mathcal{O}$ denote the set of states and objects, respectively. As discussed earlier, the goal of CZSL is to generalize to unseen compositions C_u such that $C_s \cap C_u = \phi$. We denote the test set by $\mathcal{D}^{\text{test}}$ consisting of both seen and unseen compositions from $C = C_s C_u$, similar to the generalized zero-shot setting (2).

Broadly, our model consists of two streams: a vision transformer $\mathcal{F}$ to encode the image information and a semantic branch $\mathcal{G}$ consisting of the combination of four graph CNNs and multi-layer perceptrons, respectively. Both the streams aim to project the visual and compositions onto a joint embedding space where an adaptive margin based softmax formulation is used to maximize the compatibility between the correct image-composition pairs while minimizing the same for the incorrect pairs. In addition, the margin helps to ensure enriched discrimination for classification. In the following, we first discuss about the projection networks for the compositions and images, following which the loss function is detailed (Fig. 1).

Multi-stage Graph Embeddings. Our method consists of three stages of graph convolutional networks (GCNs) to preserve and improve the transfer of topological information from the original state, object, and composition space to the embedding space. Initially, we learn the pairwise similarity measure for the objects, states, and the compositions separately using GCNs. Subsequently, the embeddings obtained from these graphs are accumulated in a joint adjacency matrix which learns two things: the relation between a composition $y = (s, o)$ and the respective state s and object o labels and the neighborhood relation among the compositions. The output of this final GCN is treated as the latent space, which ensures improved discrimination among the compositions while preserving the semantics of the space. Let $G_{s/o} = (V_{s/o}, E_{s/o})$ be the graph corresponding to the state/object space, where vertices $V_{s/o}$ correspond to word embeddings of each state/object. Subsequently, we represent the input node feature matrix $V_{s/o} \in \mathbb{R}^{|S|/|O|} \times P$, where P denotes the feature dimension of each node (e.g. 300 for word2vec). Edges $E^{ij}_{s/o}$ are defined as follows:

$$E^{ij}_{s/o} = \begin{cases} 0 & \cos(s_i/o_i, s_j/o_j) \leq t \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Here, t denotes a threshold which we typically set to 0.5 and cos denotes the cosine similarity metric. Thus, our adjacency matrices for both states and objects, $\mathcal{A}_s \in \mathbb{R}^{|S|\times|S|}$ and $\mathcal{A}_o \in \mathbb{R}^{|O|\times|O|}$ respectively, are symmetrical, and the graphs are undirected and unweighted.

Next, we construct two graph convolutional networks (GCNs) for states and objects separately to learn the feature representation of nodes. Let us denote the GCN for states by $\mathcal{G}_s$ and the GCN for object by $\mathcal{G}_o$, and the obtained GCN embeddings corresponding to states and objects by H_s/H_o, respectively. The input to the GCNs consist of their respective node feature matrices V_s/V_o and adjacency matrices $\mathcal{A}_s/\mathcal{A}_o$. Moreover, the GCN also consists of a diagonal matrix $D_s \in \mathbb{R}^{|S|\times|S|}$ and $D_o \in \mathbb{R}^{|O|\times|O|}$ such that it normalizes the rows of the adjacency matrix to preserve the magnitude of the feature vectors. Finally, each GCN convolves over the adjacency matrix using the iterative propagation rule L times for a GCN of depth L, where for a given $l \in L$, the transformation is defined as,

$$H_s^{(l+1)} = \sigma(D_s^{-1} \mathcal{A}_s H_s^{(l)} \theta_s^{(l)}) \tag{2}$$

$$H_o^{(l+1)} = \sigma(D_o^{-1} \mathcal{A}_o H_o^{(l)} \theta_o^{(l)}) \tag{3}$$

Here, σ denotes the ReLU activation function and $\theta_{s/o}$ is the learnable filters operating over the inputs. $H^{(l)}_{s/o}$ represents the hidden representation of the l^{th} layer, given $H^{(0)}_{s/o} = V_{s/o}$. At this stage, the output of the GCNs contains useful neighbourhood information that conveys how different states and objects are related to one another. Further, we concatenate these feature representations and pass the concatenated quantity through a fully connected layer $\mathcal{N}$ to reduce its dimensionality. We represent this quantity as $\mathcal{H}^{(L)}_{s,o} = \mathcal{N}([H^{(L)}_s, H^{(L)}_o])$, where $[\cdot, \cdot]$ denotes the concatenation operation which makes input to $\mathcal{N}$ of the dimension $\mathbb{R}^{|S|+|O|\times P}$. This final feature vector $\mathcal{H}^{(L)}_{s,o}$ is implicitly learnt by the model and can be treated as prior information or bias about the compositions. This stage is given by the block (a) in Fig. 1

For the second stage, we create a graph operating over the total compositional space $C = C_s C_u$. Let us denote the compositional graph by $G_c = \{V_c, E_c\}$, where V_c and E_c are the vertices and edges, respectively. The feature node matrix of vertices $V_c \in \mathbb{R}^{|C|\times 2P}$ of the compositional graph are computed by horizontally concatenating the word embeddings of the respective states and objects (each with P dimensions). Similar to $G_{s/o}$, we deduce the edge weights E_c for for G_c by applying a threshold on the cosine similarity on the embeddings as per Eq. 1. Similarly, the compositional adjacency matrix $\mathcal{A}_c$ is symmetrical, unweighted and un-directed. The compositional GCN $\mathcal{G}_c$ produces feature representations $H^{(L)}_c$ that contain the pairwise similarity information for the compositions. This GCN is formulated as per Equation 4 for a given $l \in L$. Here, θ_c denotes the learnable weights while other notations correspond to the same quantities as before. This stage is given by the block (b) in Fig. 1

$$H_c^{(l+1)} = \sigma(D_c^{-1} \mathcal{A}_c H_c^{(l)} \theta_c^{(l)}) \tag{4}$$

All the $\mathcal{G}_s$, $\mathcal{G}_o$, and $\mathcal{G}_c$ are made up by stacking multiple graph convolution layers. We now aim to combine the final embeddings $\mathcal{H}^L_s$, $\mathcal{H}^L_o$, and H^L_c, which are constrained to have identical feature dimensions of $2P$, through the composition GCN $\mathcal{G}_f$. Let the graph for the final stage be $G_f = \{V_f, E_f\}$, where V_f and E_f denote the vertices and edges of the graph. Each vertex of this graph represents either a state, object, or a compositions obtained from $\mathcal{G}_s + \mathcal{N}$, $\mathcal{G}_o + \mathcal{N}$, and $\mathcal{G}_c$, respectively: $V_f \in \mathbb{R}^{|S|+|O|+|C|\times 2P}$. While constructing the adjacency matrix $\mathcal{A}_f$ for G_f, we seek to ensure the followings: a given composition should have high similarity with the respective state and object nodes. Hence for a given $y = (s, o)$, we consider $E_f(y, s) = E_f(y, o) = 1$. On the other hand, we aim to preserve the neighborhood interaction among the compositions, hence for two compositions y_1 and y_2, we calculate $E_f(y_1, y_2)$ as per Eq. 1. The working principle of the third stage is shown below.

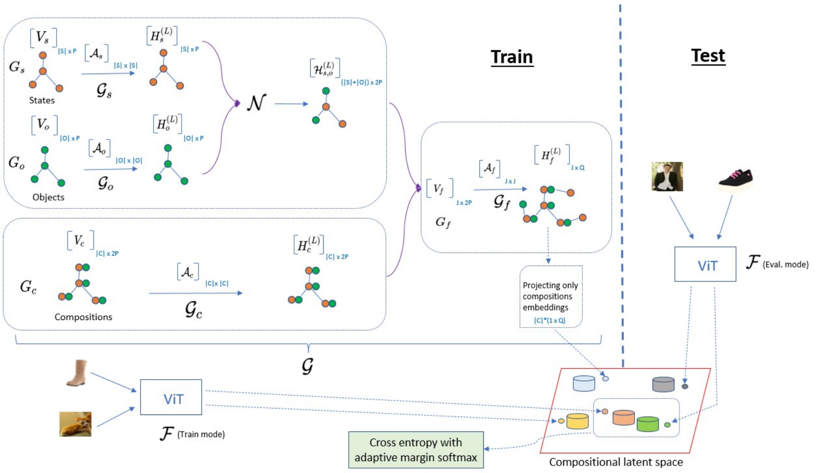

Fig. 1: Working principle of the Multi-stage semantic graph embedding model. Blue arrow denotes forward pass through a function and purple arrow represents the concatenation operation. The quantities H_s^L, H_o^L, and H_f^L are the final feature embeddings from the respective graphs of the multi-stage model, while $\mathcal{H}_{s,o}^L$ is the output of the fully connected layer $\mathcal{N}$. Here, J denotes $(|S| + |O| + |C|)$

$$\begin{aligned} \hat{H}_f^{(0)} &= [\mathcal{H}_{s,o}^{(L)}, H_c^{(L)}, H_f^{(L)}] \\ H_f^{(l+1)} &= \sigma(D_f^{-1} \mathcal{A}_f \hat{H}_f^{(l)} \theta_f^{(l)}) \end{aligned} \tag{5}$$

Here, the trainable filter weights are denoted by θ_f and other quantities correspond to the definitions mentioned for previous GCNs. The final stage is given by the block (c) in Fig. 1 We represent the final compositional embedding obtained for a given y as $\mathcal{G}(y)$.

In principle, our semantic projection module is distinctively different from (6). While (6) considers only one graph for learning the interactions between $y = (s, o)$ and s/o relying on the embeddings obtained from a pre-trained word embedding model (word2vec), we propose to separately learn the topology of the object, state, and composition spaces through $(\mathcal{G}_c, \mathcal{G}_o, \mathcal{G}_s)$, following which we integrate these information using $\mathcal{G}_f$. This leads to improved concept learning.

Visual Features: Existing works in CZSL (6; 5; 1) have all used pre-trained CNNs as de-facto feature extractors on visual data. However, recently visual transformers (ViT) (14) have demonstrated rapidly increasing potential in the field of image recognition. In fact, it has also been claimed that ViTs have surpassed current state-of-the-art CNNs in visual tasks with four times fewer computational resources when trained on sufficient data. Similar to ResNets (15), ViTs also contain skip-connections throughout added on a self-attention layer and an MLP layer. Moreover, ViT perceive highly similar representations throughout the model, while the ResNet models show much lower similarity between lower and higher layers. This implies that lower-level information is propagated effectively through the skip-connections even into the deeper layers of the transformer to form higher-level abstract representations. Using the same analogy, we would like to describe the task of CZSL as learning to express higher-level *objects* by passing information about lower-level properties like shape, texture, color as *states*. Thus, we hypothesize that by virtue of their construction, ViTs are more suitable for the task of CZSL as compared to CNNs and incorporate more global information for strong performance. Let us denote this ViT by $\mathcal{F}$. It projects the image onto the latent space shared with the multi-stage graph network.

Loss Functions: The loss function for CZSL aims to maximize the compatibility between the image and the correct composition embeddings while simultaneously minimizing the same for all the negative compositions. It is reasonably intuitive to realize that several compositions may be closely related in the embedding space due to a shared state or object. However, it is also highly likely that two compositions that lie farther in the embedding space may correspond to images with high visual similarities. From the perspective of word embeddings, `cute cat` is far away from `small kitten`, but their images have almost identical features. This requires a loss function that implicitly guides the model to distinguish between such compositions while separating semantically different ones. To accomplish the same, we add an adaptive margin in the softmax function as shown in Eq. 6, which defines the probability of a ground truth composition y belonging to image x. The degree of adaption of the softmax function is decided by a margin $m(y, k)$ which depends on the cosine similarity between embeddings of ground truth seen composition y and other seen compositions $k \in C_s$. The high similarity between k and y implies that it is more probable that y and k lie in the same neighbourhood, thus ensuring increase in margin that separates the k from the ground truth is desired. Similarly, low cosine similarity between k and y implies that the model learns a stronger de-correlation between k and y, hence, the margin value can be small. Thus, our model learns a more efficient and robust margin between compositions in the latent space. To the best of knowledge, ours is one of the primitive approaches to adaptively increase the intra-class separation through the margin-based softmax function.

$$\hat{p}(y|x) = \frac{e^{D(\mathcal{F}(x), \mathcal{G}(y))}}{e^{D(\mathcal{F}(x), \mathcal{G}(y))} + \sum_{k \in C_s \setminus y} e^{D(\mathcal{F}(x), \mathcal{G}(k)) + m_{y,k}}} \tag{6}$$

Here, $D(.)$ is the compatibility score between the image embedding and the composition embedding (Eq. 7) and the adaptive margin given by $m_{y,k}$ is calculated by an appropriately scaled version of the cosine similarity between the composition embeddings $\mathcal{G}(y)$ and $\mathcal{G}(k)$ as per Eq. 8,

$$D(\mathcal{F}(x), \mathcal{G}(k)) = \mathcal{F}(x) \cdot \mathcal{G}(k) \tag{7}$$

$$m_{y,k} = \alpha \cos(\mathcal{G}(y), \mathcal{G}(k)) \tag{8}$$

where we consider $\alpha = 2$ all through. Finally, after introducing the adaptive margin, we minimize the cross entropy as follows,

$$\min_{\mathcal{G}_s,\mathcal{G}_o,\mathcal{G}_c,\mathcal{G}_f,\mathcal{N},\mathcal{F}} \frac{1}{|\mathcal{C}_s|} \sum_{(x,y)\in\mathcal{C}_s} -\log(\hat{p}(y|x)) \tag{9}$$

4. Experiments

Datasets: We consider three benchmark datasets to validate our model (2): MIT-States (16), UT Zappos50k (17), and C-GQA (6), respectively. MIT-States is a collection of 53k natural images with 245 object classes and 115 attribute classes. UT Zappos50k is a collection of catalog images of shoe with around 29k images used for CZSL but has few attribute and object classes, 16 and 12, respectively. Compositional-GQA (C-GQA) is the most extensive dataset for CZSL, consisting of around 38k images and over 9.5k compositions.

Evaluation metrics: We adopt the evaluation protocol proposed by (2) wherein the generalized setting considers both seen and unseen compositions in the evaluation. Henceforth, we report the Area Under the Curve (AUC) (in %) at different operating points of the curve, accuracy when prediction is made on only unseen and seen compositions respectively, and their Harmonic mean (HM).

Implementation: For the image feature extractor $\mathcal{F}$, we use the vision-image transformer (ViT) (14) backbone pretrained on the ImageNet dataset. Note that we train our feature extractor end-to-end along with the GCNs. We train a total of four $L = 2$-layer GCNs, one each on the states $\mathcal{G}_s$, objects $\mathcal{G}_o$, compositions $\mathcal{G}_c$ and the last one on the combined graph $\mathcal{G}_f$. In each layer of GCN, we employ a Dropout layer, a linear layer with a hidden dimension of 4096 followed by a ReLU non-linearity. Similarly, $\mathcal{N}$ is designed as a single dense layer coupled with ReLU activation function. This layer converts the P-dimensional state and object embeddings to $2P$ dimensions to match the output dimensionality of $\mathcal{G}_c$. We use the pretrained word2vec (18) model to initialize our word embeddings in the case of UT-Zappos50k, whereas for C-GQA and MIT-States we use a concatenation of word embeddings from the pretrained fasttext (19) and word2vec models, as suggested by other recent works (6).

Training details: The model is trained with the Adam optimizer (20) with learning rates of $5e^{-6}$ and $5e^{-5}$ for $\mathcal{F}$ and $\mathcal{G}$, respectively and use some of the boilerplate code provided by (6) to implement our model using PyTorch (21). We use a batch size of 32 for all the datasets and train our experiments on an 11 GB GeForce RTX 2080 Ti graphics card. We reach our best results in MIT-States, UT-Zappos50k and C-GQA datasets within 20, 30 and 20 epochs, respectively.

5. Results & Discussions

Comparison to the literature: In Table 1, we compare our results with the current state of the art models and demonstrate that the proposed method provides significantly better results. To compare the performace of ViTs and ResNets, we also compare our model's performance with a fixed ResNet-18 backbone based version of (CGE) denoted by CGE$_{ff}$.

For the MIT-States dataset, we outperform our closest competitor (CGE) (6) by 37% by obtaining an AUC of 8.93%. As seen in Table 1, our model also performs significantly better than a ResNet feature extractor model CGE$_{ff}$. At this stage, it is crucial to point out that this dataset consists of natural objects in different states collected using an older search engine with limited human annotation resulting in significant label noise. Thus, the robustness of our model to high label noise is evident from our results. Apart from quantitative results, we also show that our model predicts fewer *distractor* compositions compared to (6). Fig 3 demonstrates that our model predicts semantically accurate labels when CGE predicts a distractor. This discrepancy is most apparent in the case of the third image, where CGE only predicts the size of the `phone` as `small`. But, our model predicts a more meaningful state `shattered`, which is more useful information about a `phone` in real life. We would like to point out that CGE predicts the correct object but a distractor state due to incomplete or biased exploration of semantic space. Moreover, our ViT based backbone aids in recognizing lower-level features important to predict the states.

In Fig. 5, we present the adjacency matrices corresponding to embedding features of composition learned by CGE (6) and our model for MIT-States. Ideally, this matrix should be sparse i.e. majority of their elements should be equal to zero. From Fig. 5, we observe that this requirement is closely satisfied by learned embeddings of our model, at the same time the adjacency matrix corresponding to CGE shows a high correlation between a high number of compositions. Thus, we verify that our model predicts fewer distractors than CGE.

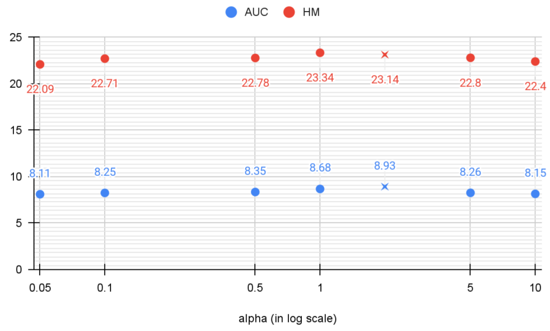

Fig. 6: Ablation study on parameter α for MIT-States. x denotes the data point for $\alpha = 2$, which we have used for our results.

Model ablation: Here we analyze the effects of different model components: i) the effectiveness of ViT by comparing it against a ResNet-18 based visual feature encoder, ii) ablation on the different GCN components used, and iii) study of the effectiveness of the adaptive margin based softmax over softmax with-

Table 1: Comparison to the literature for all the three datasets. In CGE$_{\text{ff}}$, ResNet-18 backbone is not trained.

Method	**MIT-States**				**UT-Zappos50k**				**C-GQA**			
	AUC	Unseen	Seen	HM	AUC	Unseen	Seen	HM	AUC	Unseen	Seen	HM
AttOp (3)	1.6	17.4	14.3	9.9	25.9	54.2	59.8	40.8	0.3	3.9	11.8	2.9
LE+ (1)	2.0	20.1	15.0	10.7	25.7	61.9	53.0	41.0	0.6	5.0	16.1	5.3
TMN (2)	2.9	20.1	20.2	13.0	24.7	49.4	58.8	40.7	1.1	6.3	21.6	7.7
SymNet (4)	3.0	25.2	24.4	16.1	23.9	57.9	53.3	39.2	1.8	9.2	25.2	9.8
CompCos (5)	4.5	24.6	25.3	16.4	28.7	62.5	59.8	43.1	-	-	-	-
CGE$_{\text{ff}}$ (6)	5.1	25.3	28.7	17.2	26.4	63.6	56.8	41.2	2.5	11.7	27.5	11.9
CGE (6)	6.5	28.0	32.8	21.4	32.5	**65.9**	61.0	47.1	3.6	14.0	31.4	14.5
ProtoProp (13)	-	-	-	-	34.7	65.5	62.1	50.2	-	-	-	-
Ours	**8.93**	**31.22**	**38.36**	**23.14**	**35.14**	65.26	**67.74**	**50.62**	**4.31**	**14.34**	**36.13**	**16.03**

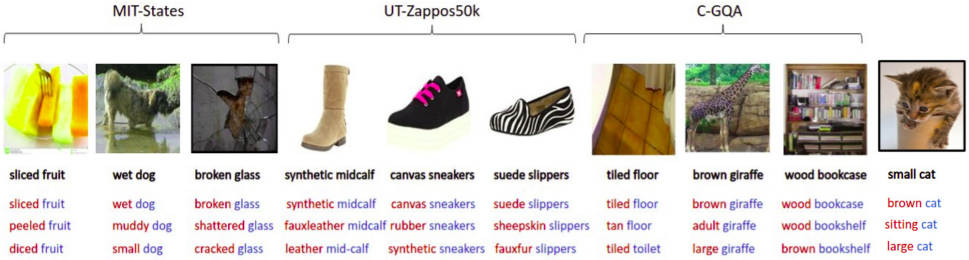

Fig. 2: Qualitative results of the proposed multi-stage graph embedding model where we show the top three predicted compositions. Our model always predicts the semantically similar compositions.

Table 2: Ablation study on UT-Zappos50K and C-GQA.

Method	**UT-Zappos50K**		**C-GQA**	
	AUC	HM	AUC	HM
ResNet18$_{\text{ff}}$ + $\mathcal{G}$	28.7	43.21	0.68	5.18
ViT$_{\text{ff}}$ + $\mathcal{G}$	30.68	46.11	2.37	11.46
ResNet18 (trainable) + $\mathcal{G}$	33.51	48.11	1.99	10.05
ViT (trainable) + $\mathcal{G}_s + \mathcal{G}_o + \mathcal{G}_c$	4.95	14.60	0.01	0.11
ViT (trainable) + $\mathcal{G}_c$	4.15	13.4	0.01	0.09
ViT (trainable) + $\mathcal{G}_f$	29.1	43.57	3.24	13.46
ViT (L = 4) (trainable) + $\mathcal{G}$	21.92	38.47	1.94	9.87
ViT (L = 1) (trainable) + $\mathcal{G}$	17.84	32.55	1.45	8.43
ViT (trainable) + Softmax	33.02	47.24	4.16	15.94
ViT (trainable) + Softmax-fixed (0.4)	**35.20**	49.39	4.18	15.78
ViT (trainable) + Softmax-(22)	34.87	48.36	4.21	16.01
ViT (trainable) + Softmax-adaptive (Ours)	35.14	**50.62**	**4.31**	**16.03**

out any margin and softmax with fixed margin, respectively, for UT-Zappos and C-GQA.

For both ResNet-18 and ViT, we consider training with the ImageNet pre-trained versions (ResNet18$_{ff}$ and ViT$_{ff}$) and the end-to-end trainable models (ResNet18 and ViT) respectively in Table 2. It can be seen that a trained model performs better, while the ViT model beats the ResNet18 counterparts by 1.98% for the fixed backbone and 1.63% for the trainable backbone in terms of the AUC values and by more than 2.5% for the HM values between the seen and unseen class accuracies.

In order to showcase the effects of the proposed graph embedding, we compare our model against two baselines, a) when $\mathcal{G}_c$ and $\mathcal{G}_f$ are individually used, b) when we do not combine the state and object latent features, i.e., a GCN consisting of $\mathcal{G}_s + \mathcal{G}_o + \mathcal{G}_c$. Table 2 clearly suggests that while the baselines perform poorly, indicating the importance of jointly modeling the state and object spaces, the complete $\mathcal{G}$ beats the baseline only with $\mathcal{G}_c$ by more than 20% and the baseline with $\mathcal{G}_f$ by at least 3% in the HM score for UT-Zappos and at least 5% for C-GQA. Concretely, $\mathcal{G}_s$+$\mathcal{G}_o$+$\mathcal{G}_c$ performs poorly as their is no interaction and exchange of knowledge among these graphs. Similarly, using $\mathcal{G}_c$ hinders performance due to inadequate knowledge interactions between states and objects.

We compare the effects of the adaptive margin against three cases: a) softmax with no margin, b) a fixed margin, c) adaptive margin of (22), respectively. It can be seen from Table 2 that the proposed margin learning produces superior performance in the HM score by at least 3% from the model without margin, by > 1.5% from the fixed margin, and by > 1.3 from (22), respectively. Further Fig. 6 shows the sensitivity analysis of the hyper-parameter α. Finally, we study the effect of varying the depth of the GCN L on the performance of the proposed method. It can be observed that employing GCNs with only two layers is optimal.

Qualitative results: Fig. 2 presents examples of images and top-3 predicted compositions for all three datasets. The corresponding ground truth is mentioned in black. For the MIT-States dataset and UT-Zappos50k dataset, we observe that the top-3 predictions contain information about the distinct visual features. Our model predicts states that accurately perceive information related to shape, texture, form, and size. For the C-GQA (6) dataset, we observe that our model also classifies the image into a different yet accurate object classes.

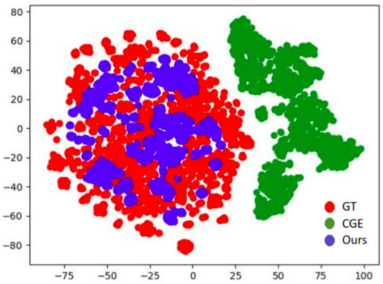

Fig. 7: t-SNE visualization of ground truth embeddings and embeddings learned by CGE and our model

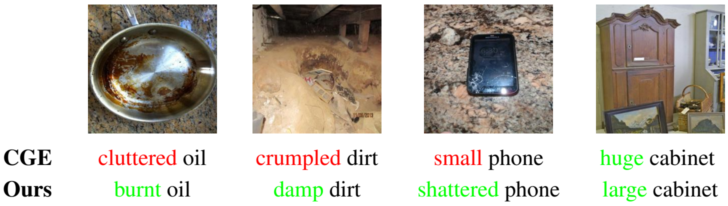

Fig. 3: Qualitative comparison of compositions predicted by our proposed model and CGE (6) on MIT-States. The color red and green has been used to indicate invalid and valid words.

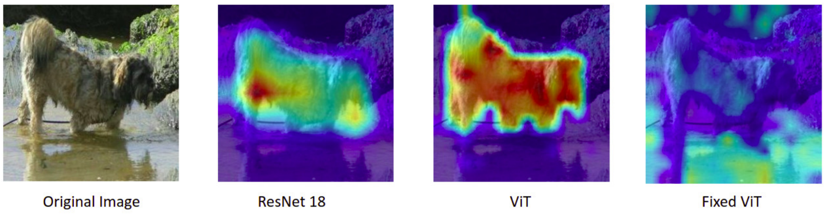

Fig. 4: GradCAM visualization for ResNet18, ViTs, and fixed ViTs

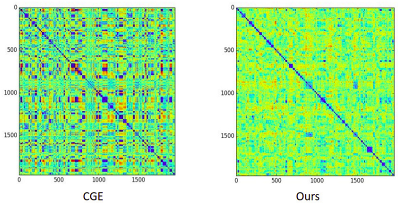

Fig. 5: A visual comparison of adjacency matrices corresponding to embeddings learned by CGE (6) and our method.

6. Conclusions

We tackle the problem of CZSL in this paper, where the task is to recognize unseen composition labels as the visual primitives during testing given a model trained on a disjoint set of visual-composition pairs. We focus on designing a superior semantic embedding space respecting the neighborhood topologies jointly for the states, objects, and composition labels. This accounts for a novel semantic projection network consisting of four GCNs. Further, an adaptive margin based softmax formulation for the cross-entropy loss is introduced to ensure discriminativeness in the shared latent space while limiting the prediction of distractors. Finally, we perform extensive experimentation on all the three benchmark datasets to confirm the superiority of the model.

References

[1] I. Misra, A. Gupta, M. Hebert, From red wine to red tomato: Composition with context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1792–1801.

[2] S. Purushwalkam, M. Nickel, A. Gupta, M. Ranzato, Task-driven modular networks for zero-shot compositional learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3593–3602.

[3] T. Nagarajan, K. Grauman, Attributes as operators, ECCV.

[4] Y.-L. Li, Y. Xu, X. Mao, C. Lu, Symmetry and group in attribute-object compositions, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 11313–11322.

[5] M. Mancini, M. F. Naeem, Y. Xian, Z. Akata, Open world compositional zero-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5222–5230.

[6] M. Naeem, Y. Xian, F. Tombari, Z. Akata, Learning graph embeddings for compositional zero-shot learning, in: 34th IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2021.

[7] B. Ommer, J. M. Buhmann, Learning the compositional nature of visual objects, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[8] S. Fidler, A. Leonardis, Towards scalable representations of object categories: Learning a hierarchy of parts, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[9] P. Ott, M. Everingham, Shared parts for deformable part-based models, in: CVPR 2011, IEEE, 2011, pp. 1513–1520.

[10] B. Zhao, B. Chang, Z. Jie, L. Sigal, Modular generative adversarial networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 150–165.

[11] C.-Y. Chen, K. Grauman, Inferring analogous attributes, 2014 IEEE Conference on Computer Vision and Pattern Recognition.

[12] Y. Atzmon, F. Kreuk, U. Shalit, G. Chechik, A causal view of compositional zero-shot recognition, in: Advances in Neural Information Processing Systems (NeurIPS), 2020.

[13] F. Ruis, G. Burghouts, D. Bucur, Independent prototype propagation for zero-shot compositionality (2021). `arXiv:2106.00305`.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[16] P. Isola, J. J. Lim, E. H. Adelson, Discovering states and transformations in image collections, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1383–1391.

[17] A. Yu, K. Grauman, Fine-grained visual comparisons with local learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 192–199.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[19] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[20] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[21] 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) 8026–8037.

[22] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4690–4699.