Disentangling Visual Embeddings for Attributes and Objects

Nirat Saini Khoi Pham Abhinav Shrivastava
University of Maryland, College Park

Abstract

We study the problem of compositional zero-shot learning for object-attribute recognition. Prior works use visual features extracted with a backbone network, pre-trained for object classification and thus do not capture the subtly distinct features associated with attributes. To overcome this challenge, these studies employ supervision from the linguistic space, and use pre-trained word embeddings to better separate and compose attribute-object pairs for recognition. Analogous to linguistic embedding space, which already has unique and agnostic embeddings for object and attribute, we shift the focus back to the visual space and propose a novel architecture that can disentangle attribute and object features in the visual space. We use visual decomposed features to hallucinate embeddings that are representative for the seen and novel compositions to better regularize the learning of our model. Extensive experiments show that our method outperforms existing work with significant margin on three datasets: MIT-States, UT-Zappos, and a new benchmark created based on VAW. The code, models, and dataset splits are publicly available at https://github.com/nirat1606/OADis.

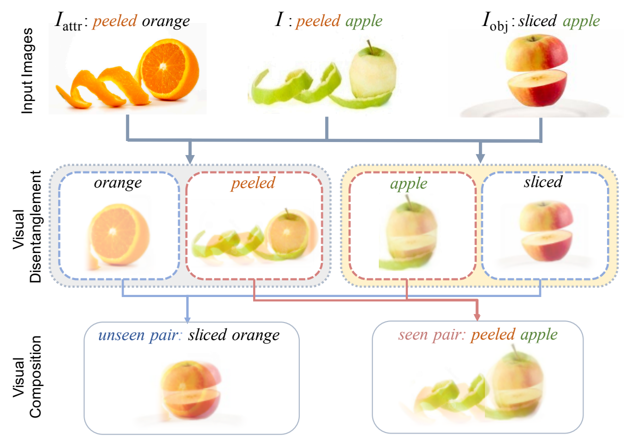

Figure 1. **Method illustration**: Given an input image I of `peeled apple`, we use two other images: (1) one with same object, different attribute I_{obj} - `sliced apple`, (2) one with same attribute, different object I_{attr} - `peeled orange`. We propose a novel architecture that takes I and I_{attr}, and extracts their visual similarity features for `peeled` and visual dissimilarity features for `orange`. Similarly, using I and I_{obj}, the visual similarity features for `apple`, and the dissimilarity features for `sliced` can be extracted. We compose these primitive visual features to hallucinate a seen pair `peeled apple`, and a novel unseen pair `sliced orange` to be used for regularizing our embedding space. Note that this is a visualization of embedding space composition, we do not generate images.

1. Introduction

Objects in the real world can appear with different properties, *i.e.*, different color, shape, material, etc. For instance, an apple can be red or green, cut or peeled, raw or ripe, and even dirty or clean. Understanding object properties can greatly benefit various applications, *e.g.*, robust object detection [5, 14, 15, 26], human object interaction [7, 49, 51], and activity recognition [1, 3, 4, 16, 18, 34]. Since the total number of possible attribute-object pairs in the real world is prohibitively large, it is impractical to collect image examples and train multiple classifiers. Prior works proposed compositional learning, *i.e.*, learning to compose knowledge of known attributes and object concepts to recognize a new attribute-object composition. Datasets such as MIT-States [24] and UT-Zappos [56] are commonly used to study this task, with joint attribute-object recognition for a diverse, yet limited set of objects and attributes.

Compositional learning refers to combining simple primitive concepts to understand a complex concept. This idea dates back to Recognition and Composition theory by Biederman [6], and early work in the visual domain by Hoffman [22], which proposed recognition by parts for pose estimation. Prior works explore compositionality to a certain degree, *e.g.*, via feature sharing and shared embeddings space. Among them, most works use linguistically inspired losses to separate attributes and objects in the shared embedding space, then use that primitive knowledge to compose new complex pairs. Using linguistic embeddings is helpful since: (1) there is a clear distinction between attribute and object in the embedding space, and (2) these embeddings already contain semantic knowledge of similar objects and attributes, which is helpful for composition. However, unlike word embedding, it is difficult to discriminate the object and attribute in the visual embedding space. This is due to the fact that image feature extractor is usually pre-trained for object classification, often along with image augmentation (*e.g.*, color jitter) that tends to produce attribute-invariant image representation, thus does not learn objects and attributes separately. In this paper, we propose a new direction that focuses on ***visual cues***, instead of using linguistic cues explicitly for novel compositions.

Analogous to linguistic embedding, our work focuses on disentangling attribute and object in the visual space. Our method, Object Attribute Disentanglement (OADis), learns distinct and independent visual embeddings for `peeled` and `apple` from the visual feature of `peeled apple`. As shown in Figure 1, for image I of `peeled apple`, we use two other images: one with same object and different attribute I_{obj} (*e.g.*, `sliced apple`), and one with same attribute and different object I_{attr} (*e.g.*, `peeled orange`). OADis takes I and I_{obj} and learns the similarity (`apple`) and dissimilarity (`sliced`) of the second image with respect to the first one. Similarly, using I and I_{attr}, the commonality between them (`peeled`) and the left out dissimilarity (`orange`) can also be extracted. Further, composition of these extracted visual primitives are used to hallucinate seen and unseen pair, `peeled apple` and `sliced orange` respectively.

For compositional learning, it is necessary to decompose first before composing new unseen attribute-object pairs. As humans, we have the ability to imagine an unseen complex concept using previous knowledge of its primitive concepts. For example, if someone has seen a `clown` and a `unicycle`, they can imagine `clown on a unicycle` even if they have never seen this combination in real life [20, 43]. This quality of *imagination* is the basis of various works such as GANs [12], CLIP [47] and DALL-E [48]. However, these works rely on larger datasets and high computation power for training. We study this idea of *imagination* for a smaller setup by composing newer complex concepts using disentangled attributes and object visual features. Our work focuses on answering the question, *can there be visual embedding of* `peeled` *and* `apple`*, disentangled separately from visual feature of* `peeled apple`*?* Our contributions are as follows:

- We propose a novel approach, OADis, to disentangle attribute and object visual features, where visual embedding for `peeled` is distinct and independent of embedding for `apple`.
- We compose unseen pairs in the visual space using the disentangled features. Following Compositional Zero-shot Learning (CZSL) setup, we show competitive improvement over prior works on standard datasets [24, 56].
- We propose a new large-scale benchmark for CZSL using an existing attribute dataset VAW [45], and show that OADis outperforms existing baselines.

2. Related Work

Visual Attributes. Visual attributes have been studied widely to understand visual properties and low-level semantics of objects. These attributes help further improve on various downstream tasks such as object detection [5, 11, 14, 15, 26, 36], action recognition [1, 3, 4, 16, 18, 34], image captioning [25, 40], and zero-shot and semi-supervised classification [2, 10, 11, 27, 39, 41, 50]. Similar to multi-class classification for objects, initial work for attribute understanding used discriminative models [26, 42], without understanding attributes. Other works [8, 15, 23, 32] explored the relation between the same attributes and different objects, to learn visual attributes. Particularly, disentangling object features from attribute features are explored in [17, 19]. Although, these works use clustering and probabilistic models to learn the attributes of objects.

Compositional Zero-shot Learning. Concept of compositional learning was first introduced in Recognition by Parts [22]. Initially, [35] employed this concept for objects and attributes. Unlike zero-shot learning (ZSL), CZSL requires the model to learn to compose unseen concepts from already learned primitive components. [8, 35] proposed separate classifiers for primitive components, and merged all into a final classifier. Most prior works use linguistically inspired auxiliary loss terms to regularize training for embedding space, such as: [38] models attributes as a linear transformation of objects, [30] uses rules of symmetry for understanding states, and [55] learns composition and decomposition of attributes hierarchically. Another set of studies uses language priors to learn unseen attribute-object pairs, either in feature space or with multiple networks [31, 46, 52]. Other recent works use graph structure to leverage information transfer between seen to unseen pairs using Graph Convolutional Networks [33, 37], and [54] uses key-query based attention, along with modular network with message passing for learning relation between primitive concepts.

3. Object Attribute Disentanglement (OADis)

Contrary to prior works [30, 37, 38, 55], we explicitly focus on separating attributes and object features in the visual space. More precisely, TMN [46] uses word embeddings to generate attention layers to probe image features corresponding to a given pair, GraphEmbedding [37] exploits the dependency between word embeddings of the labels, and HiDC [55] mainly uses word embeddings to compose novel pairs and generate more examples for their triplet loss. To the best of our knowledge, none of the existing works have explored visual feature disentanglement of attributes and objects. We hypothesize that attribute and object visual features can be separated when considering visual feature similarities and differences between image pairs. Composing these disentangled elements help regularize the common embedding space to improve recognition performance.

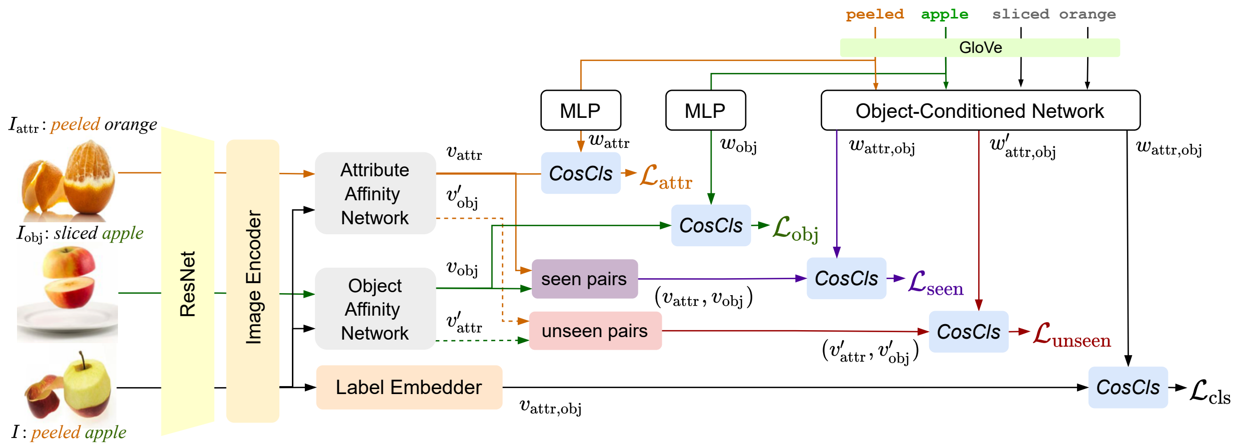

Figure 2. **System Overview**: Given an image I, for `peeled apple`, we consider two images:, one with same object: I_{obj}, `sliced apple`, and one with same attribute, I_{attr} `peeled orange`. (1) The Object-Conditioned Network composes pair word embedding, using GloVe word embeddings for labels. (2) Label Embedder uses the image I and embeds visual feature $v_{\text{attr,obj}}$ along with word embedding $w_{\text{attr,obj}}$, using loss $\mathcal{L}_{\text{cls}}$. (3) Attribute Affinity Network and Object Affinity Network, disentangles the same attribute and object from the pair of images I, I_{attr} and I, I_{obj} respectively. Disentangled visual features for `peeled` (v_{attr}) and `apple` (v_{obj}) are used along with word embeddings of attribute (w_{attr}) and objects (w_{obj}), to compute $\mathcal{L}_{\text{attr}}$ and $\mathcal{L}_{\text{obj}}$. (4) Using disentangled features, we compose seen pair `peeled apple` ($v_{\text{attr}}, v_{\text{obj}}$) and unseen pair `sliced orange` ($v'_{\text{attr}}, v'_{\text{obj}}$), for composition losses $\mathcal{L}_{\text{seen}}$ and $\mathcal{L}_{\text{unseen}}$.

More concretely, we take cues from [17] and [35, 55], to learn to compose unseen attribute-object pairs leveraging visual attributes based on auxiliary losses.

3.1. Task Formulation

We follow the conventional Compositional Zero-shot Learning (CZSL) setup, where distinct attribute-object compositions are used at training and testing. Each image I is labeled with $y = y_{\text{attr,obj}} \in Y$, where y_{attr} and y_{obj} are respectively the attribute and object label. The dataset is divided into two parts, seen pairs $y^s \in Y^s$ and unseen pairs $y^u \in Y^u$, such that $Y = Y^s \cup Y^u, Y^s \cap Y^u = \emptyset$. Although $y^u = y_{\text{attr,obj}} \in Y^u$ consists of attribute y_{attr} and object y_{obj} that are never seen together in training, they are separately seen. We employ the Generalized CZSL setup defined in [46], which has seen Y^s and unseen pairs Y^u in the validation and test sets as detailed in Table 1. As shown in Figure 2, for image I, with label `peeled apple`, we choose two additional images: one with same object and different attribute I_{obj} (*e.g.*, `sliced apple`), and another image with same attribute and different object I_{attr} (*e.g.*, `peeled orange`). Note that the subscript of image symbol, *e.g.*, attr in I_{attr}, shows similarity with I, whereas superscript denotes seen and unseen sets.

3.2. Disentangling Visual Features

We extract image and label embedding features from pre-trained networks (ResNet [21] and GloVe [44]). As seen in Figure 2, we use *Image Encoder (IE)* and *Object Conditioned Network (OCN)*, for image and word embedding features respectively. Similar to [38], we use *Label Embedder (LE)* as an additional FC-Layer for the image feature. *LE* and *OCN* learn image and word embeddings and embed those in a common pair embedding space. Next, visual similarity between I and I_{obj} is computed using *Object Affinity Network*, which extracts visual features for object, v_{obj}. Whatever is not similar is considered dissimilar. Hence, visual features of I_{obj} that are least similar to visual features of I are considered as the attribute feature v'_{attr} in I_{obj}, which is `sliced` in this example. Similarly, *Attribute Affinity Network* takes I and I_{attr}, and extracts visual similarity feature v_{attr} for `peeled`, and dissimilar visual features of I_{attr}, as object feature v'_{obj} for `orange`. The disentangled features are then used to compose seen and unseen pairs. We discuss the details in the following sections:

Image Encoder (IE). We use the second last layer before AveragePool of an ImageNet-pretrained ResNet-18 [13, 21] to extract features for all images. *IE* is a single convolutional layer that is shared across images I, I_{attr} and I_{obj} to generate their image features, represented as f, f_{attr} and f_{obj} respectively, where each $f \in \mathbb{R}^{n \times 49}$ and n is the output dimension of IE.

Label Embedder (LE). Inspired by [38], our *LE* inputs spatial feature from ResNet [21], AveragePools and passes through a linear layer to extract final feature $v_{\text{attr,obj}}$ for pair embedding, which has same dimension as the word embedding final feature $w_{\text{attr,obj}}$, extracted from *Object Conditioned Network (OCN)* (Figure 2). This is the main branch, and is used for input image I only.

Object Conditioned Network (OCN). This takes word embeddings of attribute emb_{attr} and object emb_{obj}, concatenates the features and passes through multiple layers. Object-conditioned is named because a residual connection for the object feature is concatenated with the final attribute feature, and the output feature is $w_{\text{attr,obj}} \in Y$. We discuss

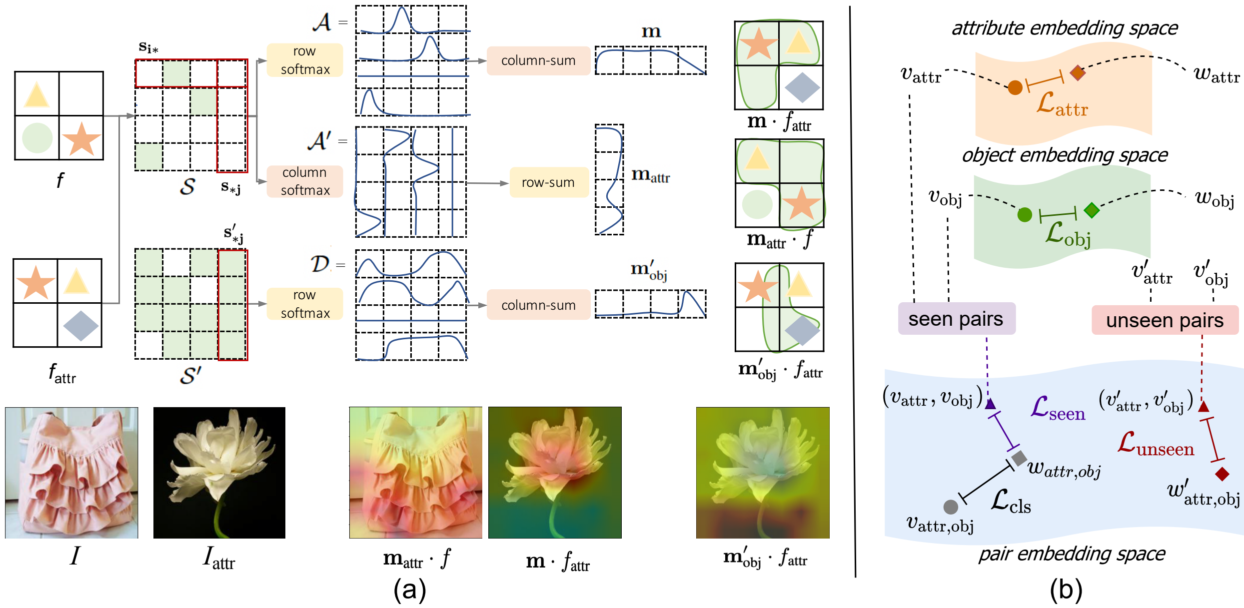

Figure 3. (a) Attribute Affinity Module: We compute the cosine similarity between blocks in f and f_{attr} ($\mathcal{S}$ in Eq. 3), then apply row-wise and column-wise softmax ($\mathcal{A}$ and $\mathcal{A}'$), followed by a respective column-sum and row-sum to obtain $\mathbf{m}$ and $\mathbf{m}_{\text{attr}}$. $\mathbf{m}$ represents regions where f_{attr} is highly similar to f (hence, we reshape and multiply $\mathbf{m}$ with f_{attr}) and $\mathbf{m}_{\text{attr}}$ represents regions where f is highly similar to f_{attr} (thus, $\mathbf{m}_{\text{attr}} \cdot f$). Similarly, $\mathcal{S}'$ represents the regions where feature f_{attr} is not similar to feature f (more details in Section 3.2). The last row shows real samples and generated attention maps overlayed on images. Give image `ruffled bag` and `ruffled flower`, we show that attribute `ruffle` is highlighted in the center $\mathbf{m}_{\text{attr}} \cdot f$ and $\mathbf{m} \cdot f_{\text{attr}}$. Whereas, $\mathbf{m}'_{\text{obj}} \cdot f_{\text{attr}}$ shows the dissimilar regions of I_{attr} w.r.t I. (b) Shows the three embedding spaces learnt with different losses. Same notation is used as Figure 2.

the motivation for this in Section 4.3.

Cosine Classifier (CosCls). Analogous to compatibility function used in [33, 37], we use cross-entropy along with cosine similarity to get the final score for each pair. For visual features $v_{\text{attr,obj}}$ (from *LE*), and composed word embeddings $w_{\text{attr,obj}}$ (from *OCN*), *CosCls* provides logits for an image I. For instance, let us assume $v : X \rightarrow Z$ and $w : Y \rightarrow Z$. Z is the common embedding space for word embeddings w and visual embeddings v. Then classifier unit *CosCls* gives the score for label $y \in Y^s$ is C:

$$h(v, w) = \cos(v, w) = \delta \cdot \frac{v^T w}{\|v\| \, \|w\|} \tag{1}$$

$$C(v, w) = \frac{e^{h(v,w)}}{\sum_{y \in Y^s} e^{h(v,y)}} \tag{2}$$

where δ is the temperature variable. Each loss function uses same CosCls score evaluator, with different inputs.

Object and Attribute Similarity Modules. Our main contribution is the proposed affinity modules and compositional losses. Inspired by image captioning [9,28,29], OADis uses image similarities and differences to identify visual features corresponding to attributes and objects. *Object Affinity Network (OAN)* uses f and f_{obj}, whereas *Attribute Affinity Network (AAN)* uses f and f_{attr}. For brevity, we explain the *AAN*, while the OAN follows the same architecture. Reminded that both f and $f_{\text{attr}} \in \mathbb{R}^{n \times 49}$.

Similar to [53], which computes attention between word concepts with corresponding visual blocks, we compute attention between two images I and I_{attr}. Since both images have the same attribute, *i.e.*, `peeled`, our affinity network learns visual similarity between the images, which represents the attribute. Similarity matrix $\mathcal{S}$ is the cosine similarity between f and f_{attr}, such that $\mathcal{S} \in \mathbb{R}^{49 \times 49}$ as:

$$\mathcal{S} = \frac{f^T f_{\text{attr}}}{\|f\|_2 \, \|f_{\text{attr}}\|_2} \tag{3}$$

where element s_{ij} represents the similarity between i^{th} element of f with j^{th} element of f_{attr}. Moreover, let $\mathbf{s_{i*}}$ and $\mathbf{s_{*j}}$ represent the i^{th} row and j^{th} column of $\mathcal{S}$ respectively. Then, $\mathbf{s_{i*}}$ captures the similarity of all the elements in f_{attr} with respect to i^{th} element of f. To know the most similar element among f_{attr} with respect to i^{th} element of f, we can take a row-wise `softmax` over $\mathcal{S}$. Similarly, for j^{th} element of f_{attr}, column $\mathbf{s_{*j}}$ represents the similarity with all the elements of f. Using a column-wise `softmax`, we can interpret the most similar and least similar element of f with respect to j^{th} element of f_{attr}, as shown in Figure 3. Therefore, by applying column-wise and row-wise `softmax`, we get two matrices, $\mathcal{A}$ and $\mathcal{A}'$ ($\mathcal{A}, \mathcal{A}' \in \mathbb{R}^{d \times d}, d = 49$),

$$\mathcal{A}_i = \frac{e^{\lambda \mathbf{s_{i*}}}}{\sum_{j=1}^{d} e^{\lambda s_{ij}}} \quad \text{and} \quad \mathcal{A}'_j = \frac{e^{\lambda \mathbf{s_{*j}}}}{\sum_{i=1}^{d} e^{\lambda s_{ij}}}, \tag{4}$$

where λ is the inverse temperature parameter. We compute row and column sum for $\mathcal{A}$ and $\mathcal{A}'$ respectively, to get final similarity maps, $\mathbf{m}$ and $\mathbf{m}_{\text{attr}}$,

$$m_j = \sum_{i=1}^{d} \mathcal{A}_{ij} \quad \text{and} \quad m_{\text{attr}\,i} = \sum_{j=1}^{d} \mathcal{A}'_{ij}. \tag{5}$$

Similarly, the difference between these two images f and f_{attr} is the object label, y_{obj}. Hence, we use the negative of $\mathcal{S}$ as the image difference, denoted as $\mathcal{S}'$. Then, difference of f_{attr} with respect to f would be row-wise `softmax` of difference matrix, denoted by $\mathcal{D}$. Hence, by performing column-sum over $\mathcal{D}$, we get difference map, $\mathbf{m}'_{\text{obj}}$,

$$\mathcal{D}_j = \frac{e^{\gamma \mathbf{s}'_{*\mathbf{j}}}}{\sum_{i=1}^{d} e^{\gamma s'_{ij}}} \quad \text{and} \quad m'_{\text{obj}\,i} = \sum_{j=1}^{d} \mathcal{D}_{ij}. \tag{6}$$

The final disentangled features for attribute v_{attr} and object v'_{obj}, for both AAN and OAN, can be computed as:

$$\begin{aligned} v_{\text{attr}} &= \mathbf{m} \cdot f_{\text{attr}} + \mathbf{m}_{\text{attr}} \cdot f \quad \text{and} \quad v'_{\text{obj}} = \mathbf{m}'_{\text{obj}} \cdot f_{\text{attr}} \\ v_{\text{obj}} &= \mathbf{m} \cdot f_{\text{obj}} + \mathbf{m}_{\text{obj}} \cdot f \quad \text{and} \quad v'_{\text{attr}} = \mathbf{m}'_{\text{attr}} \cdot f_{\text{obj}}. \end{aligned} \tag{7}$$

More details using a toy example can be seen in Figure 3. Using concatenation of v_{attr} and v_{obj} along with a single Linear layer, composes the pair `peeled apple`, represented by $(v_{\text{attr}}, v_{\text{obj}})$. Similarly, the disentangled visual features v'_{attr} and v'_{obj}, are used to compose unseen pair `sliced orange`, and is represented as $(v'_{\text{attr}}, v'_{\text{obj}})$.

3.3. Embedding Space Learning objectives

As shown in Figure 3b, we learn three embedding spaces: (1) attributes space, (2) object space, and (3) attribute-object pair space. The attribute and object spaces are used for disentangling the two, whereas pair embedding is used for final pair composition and inference. OADis has separate loss functions for disentangling and composing. All loss functions are expressed in terms of *CosCls* defined previously.

The loss function for main branch, $\mathcal{L}_{\text{cls}}$ uses combined visual feature $v_{\text{attr,obj}}$ from *LE* and word embedding feature $w_{\text{attr,obj}}$ from *OCN*. $\mathcal{L}_{\text{cls}}$ is used for the pair embedding space. Similarly, $\mathcal{L}_{\text{attr}}$ and $\mathcal{L}_{\text{obj}}$ are used to learn the visual attribute and object feature, in their respective embedding spaces. $\mathcal{L}_{\text{attr}}$ pushes the visual feature of attribute, closer to the word embedding. $\mathcal{L}_{\text{obj}}$ does the same for objects in object embedding space Figure 3b. These losses cover the concept of disentanglement, and can be represented as:

$$\begin{aligned} \mathcal{L}_{\text{cls}} &= C(v_{\text{attr,obj}}, w_{\text{attr,obj}}) \\ \mathcal{L}_{\text{attr}} &= C(v_{\text{attr}}, w_{\text{attr}});\ \mathcal{L}_{\text{obj}} = C(v_{\text{obj}}, w_{\text{obj}}) \end{aligned} \tag{8}$$

For composition, we use $\mathcal{L}_{\text{seen}}$ and $\mathcal{L}_{\text{unseen}}$. Among the images seen (I, I_{attr}, and I_{obj}), disentangled features v_{obj} and v_{attr}, composes the same pair as $(v_{\text{attr}}, v_{\text{obj}})$, which we refer to as the seen composition. Note that $(v_{\text{attr}}, v_{\text{obj}})$ is different from $v_{\text{attr,obj}}$, as the former is hallucinated feature with combination of disentangled attribute and object visual features, and latter is the combined visual feature extracted with *LE*. Here, we use $\mathcal{L}_{\text{seen}}$ loss which takes the composition of disentangled features and learns to put the composition closer to $w_{\text{attr,obj}}$. Moreover, the dissimilarity aspect from *OAN* and *AAN* extracts v'_{attr} and v'_{obj}, which composes an unseen pair $(v'_{\text{attr}}, v'_{\text{obj}})$. We use $\mathcal{L}_{\text{unseen}}$ as unseen loss since the hallucinated composition is never seen among I , I_{attr}, and I_{obj}.

$$\begin{aligned} \mathcal{L}_{\text{seen}} &= C((v_{\text{attr}}, v_{\text{obj}}), w_{\text{attr,obj}}) \\ \mathcal{L}_{\text{unseen}} &= C((v'_{\text{attr}}, v'_{\text{obj}}), w'_{\text{attr,obj}}) \end{aligned} \tag{9}$$

The combined loss function $\mathcal{L}$ is minimized over all the training images, to train OADis end-to-end. The weights for each loss (α) are empirically computed:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha_1 \mathcal{L}_{\text{attr}} + \alpha_2 \mathcal{L}_{\text{obj}} + \alpha_3 \mathcal{L}_{\text{seen}} + \alpha_4 \mathcal{L}_{\text{unseen}}.$$

Table 1. This table shows dataset splits. Y^s and Y^u are seen and unseen compositions respectively. We propose a new benchmarck, VAW-CZSL [45], which has more than $10\times$ compositions in each split compared to other datasets.

	Train set			Val set	Test set
Datasets	attr.	obj.	Y^s	Y^s/Y^u	Y^s/Y^u
MIT-states [24]	115	245	1262	300 / 300	400 / 400
UT-Zappos [56]	16	12	83	15 / 15	18 / 18
VAW-CZSL [45]	440	541	11175	2121 / 2322	2449 / 2470

4. Experiment

4.1. Datasets and Metrics

We show results on three datasets: MIT-states [24], UT-Zappos [56], and a new benchmark for evaluating CZSL on images of objects in-the-wild, referred as VAW-CZSL. VAW-CZSL is created based on images with object and attribute labels from the VAW dataset [45]. Both MIT-states [24] and UT-Zappos [56] are common datasets used for this task in previous studies. MIT-states covers wide range of objects (*i.e.*, laptop, fruits, fish, room, *etc.*) and attributes (*i.e.*, mossy, dirty, raw, *etc.*), whereas UT-zappos has fewer objects (*i.e.*, shoes type: boots, slippers, sandals) and fine-grained attributes (*i.e.*, leather, fur, *etc.*).

Proposed New Benchmark. While experimenting with MIT-states [24] and UT-Zappos [56], we found several shortcomings with these datasets and discovered issues across all baselines using these datasets:

- Both datasets are small, with a maximum of 2000 attribute-object pairs and 30k images, leading to overfitting fairly quickly.
- Random seed initialization makes performance fluctuate significantly (0.2-0.4% AUC). Moreover, [2] found 70% noise in human-annotated labels on MIT-States [24].
- A new dataset C-GQA was introduced in [37], but the dataset is still small and we found a lot of discrepancies (kindly refer to the suppl.).

Table 2. We show results on MIT-states [24] and UT-Zappos [56]. Following [37, 46], we use AUC in % between seen and unseen compositions with different bias terms, along with Val, Test, attribute and object accuracy. HM is Harmonic Mean. OADis consistently outperforms on most categories with significant increment.

	MIT-States							UT-Zappos						
Model	Val@1	Test@1	HM	Seen	Unseen	Attribute	Object	Val@1	Test@1	HM	Seen	Unseen	Attribute	Object
AttrOpr [38]	2.5	2.0	10.7	16.6	18.4	22.9	24.7	29.9	22.8	38.1	55.5	54.4	38.6	70.0
LabelEmbed+ [38]	3.5	2.3	11.5	16.2	21.2	25.6	27.5	35.5	22.6	37.7	53.3	58.6	40.9	69.1
TMN [46]	3.3	2.6	11.8	22.7	17.1	21.3	24.2	35.9	28.4	44.0	58.2	58.0	40.8	68.4
Symnet [30]	4.5	3.4	13.8	24.8	20.0	26.1	25.7	27.4	27.7	42.5	56.7	61.6	44.0	70.6
CompCos [33]	6.9	4.8	16.9	26.9	24.5	28.3	31.9	**40.8**	26.9	41.1	57.7	62.8	43.3	73.0
GraphEmb [37]	7.2	5.3	18.1	28.9	25.0	27.2	32.5	33.9	24.7	38.9	58.8	61.0	44.0	72.6
OADis	**7.6**	**5.9**	**18.9**	**31.1**	**25.6**	**28.4**	**33.2**	**40.8**	**30.0**	**44.4**	**59.5**	**65.5**	**46.5**	**75.5**

Table 3. We show results on VAW-CZSL. Since it is a much more challenging dataset, with significantly large number of compositions, to discriminate performance among different baseline, we show top-3 and top-5 AUC (in %) for Val and Test sets.

	Val. Set		Test Set						
Model	V@3	V@5	V@3	V@5	HM	Seen	Unseen	Attr.	Obj.
AttrOpr [38]	1.4	2.5	1.4	2.6	9.1	16.4	11.7	13.7	34.9
LabelEmbed+ [38]	1.5	2.8	1.6	2.8	9.8	16.2	13.2	13.4	35.1
Symnet [30]	2.3	3.9	2.3	3.9	12.2	19.1	15.8	**18.6**	40.9
TMN [46]	2.2	3.9	2.3	4.0	11.9	19.9	15.4	15.9	38.3
CompCos [33]	3.1	5.6	3.2	5.6	14.2	23.9	18.0	16.9	41.9
GraphEmb [37]	2.7	5.3	2.9	5.1	13.0	23.4	16.8	16.9	40.8
OADis	**3.5**	**6.0**	**3.6**	**6.1**	**15.2**	**24.9**	**18.7**	17.5	**43.3**

To address these limitations, we propose a new benchmark **VAW-CZSL**, a subset of VAW [45], which is a multi-label attribute-object dataset. We sample one attribute per image, leading to much larger dataset in comparison to previous datasets as shown in Table 1 (details in the suppl.).

Evaluation. We use Generalized CZSL setup, defined in [46], with dataset statistics presented in Table 1. As observed in prior works [37, 46], a model trained on a set of labels Y^s, does not generalize well on unseen pairs Y^u. Therefore, [37, 46] use a scalar term for overcoming the negative bias for unseen pairs. We use the same evaluation protocol, which computes Area Under the Curve (AUC) (in %) between the accuracy on seen and unseen compositions with different bias terms [46]. Larger bias term leads to better results for unseen pairs whereas smaller bias leads to better results for seen pairs. Harmonic mean is reported, to balance the bias. We also report the attribute and object accuracy for unseen pairs, to show improvement due to visual disentanglement of features. Our new benchmark subset for VAW [45], follows the similar split as other datasets. In addition, we conduct all experiments with image augmentation for all methods (discussed in Section 4.3).

4.2. Results and Discussion

Baselines. We compare with related recent and prominent prior works: AttrOp [38], LabelEmbed+ [38], TMN [46], Symnet [30], CompCos [33] and GraphEmb [37]. We do not compare with BMP [54], since it uses the concatenation of features from all four ResNet blocks (960-d features), resulting in higher input features and the number of network parameters than all other setups. Moreover, GraphEmb [37] is state-of-the-art; hence, comparing with that makes our work comparable to other baselines that [37] already outperforms. To be consistent, we state the performance of all models (including GraphEmb [37]) using frozen backbone ResNet without fine-tuning the image features, and using GloVe [44] for the object and attribute word embeddings. Before passing through backbone, training images are augmented with horizontal flip and random crop. Compared to other baselines, OADis uses convolutional features rather than AvgPooled, since it is easier to segregate visual features in the spatial domain for attributes and objects. Moreover, other studies [33, 37] have also used additional FC layers on top of *IE*, which we argue makes it fair for us to use pre-pooled features for OADis.

Results on MIT-States. MIT-states has considerable label noise [2], but still is a standard dataset for this task. We show significant improvement on this dataset (reported in Table 2), from previous state-of-the-art GraphEmb, which has 7.2 Val AUC and 5.3 Test AUC. Note that we do not report GraphEmb results with fine-tuning backbone, as we find it incomparable with other baselines that did not incorporate fine-tuning as part of their proposed methods. Overall, our model performs significantly better than GraphEmb on all metrics.

Results on UT-Zappos. Similar improvement trends hold for UT-Zapopos as well (see Table 2). Although, as explained for GraphEmb, it is difficult to balance the best performance for Val and Test set in this dataset. The problem is that 7/36 ($\sim$20%) attributes in Test set do not appear in Val set. Hence, improving Val set AUC, does not necessarily improve Test AUC for UT-Zappos. Similar trend can be seen for other baselines: CompCos has best Val AUC, but does not perform well on Test set, compared to TMN and Symnet. Even GraphEmb in their final table show the frozen backbone network has much lower performance than TMN. However, OADis performs well on UT-Zappos overall, with $\sim$4.0 improvement for Val and Test AUC, HM, unseen and object accuracy.

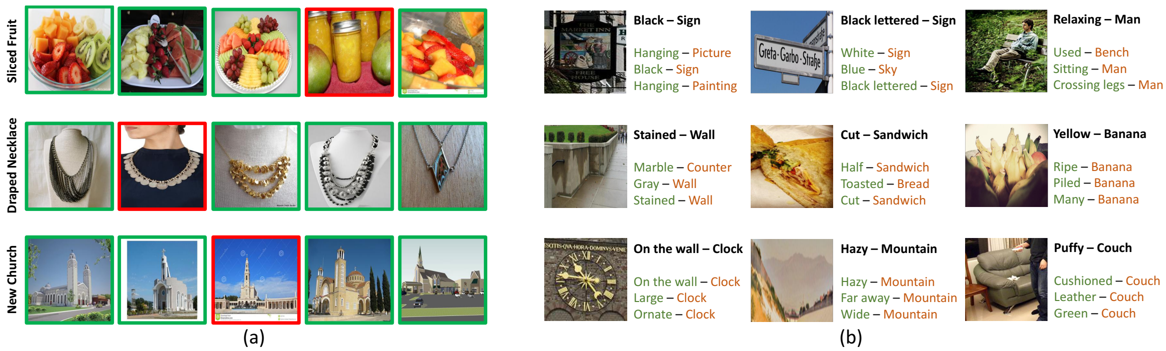

Figure 4. **Qualitative Results**: We show the nearest neighbors using the hallucinated unseen composition features for MIT-states and UT-Zappos. Although, all the neighbors are not correct (represented with red outline), they look very similar to true class labels: (a) First row: pureed fruit, Second row: engraved coin, Third row: huge tower. (b) We show top-3 predictions for images in VAW-CZSL.

Table 4. We quantitatively show that the proposed architecture and different losses help in disentanglement and composition of unseen pairs. The experiments are conducted on MIT-States [24], where change in accuracy is shown with green and red based on increment or decrement respectively from the previous row. A dash (-) represents no change more than ($\pm$ 0.1). Refer to Section 4.2 for details.

Losses	Val AUC@1	Test AUC@1	Seen	Unseen	Attribute	Object
$\mathcal{L}_{\text{cls}}$	7.24	5.43	29.92	25.33	28.03	33.10
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{attr}}$	-	-	31.09 (+2.0)	-	28.30 (+0.3)	-
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{obj}}$	-	-	-	25.50 (+0.2)	-	33.38 (+0.2)
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{attr}} + \mathcal{L}_{\text{obj}}$	7.49 (+0.2)	5.73 (+0.2)	-	-	28.50 (+0.2)	-
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{attr}} + \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{seen}}$	-	5.44 (-0.5)	31.21 (+0.2)	-	28.18 (-0.4)	-
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{attr}} + \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{unseen}}$	-	5.73 (+0.3)	-	25.80 (+0.4)	28.51 (+0.4)	-
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{attr}} + \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{seen}} + \mathcal{L}_{\text{unseen}}$	7.62 (+0.2)	5.94 (+0.2)	31.64 (+0.4)	25.60 (-0.2)	28.51	33.20

Table 5. Results with different networks for word-embeddings. Object-conditioning with attribute performs the best, and is therefore used for OADis (Section 4.3).

	Linear	MLP	Obj-cond. Network
Val@1	6.6	7.0	**7.6**
Test@1	5.0	5.2	**5.9**

Results on VAW-CZSL. Our model performs well on VAW-CZSL, and is consistently better than other methods across almost all metrics. As shown in Table 1, VAW-CZSL has $\sim$6-8 times more pairs in each split than MIT-States, which shows how challenging the benchmark is. Due to top-1 AUC being too small to quantify any learning and comparing between methods, we report top-3 and top-5 AUC instead. This is also because objects in-the-wild tend to depict multiple possible attributes; hence, evaluating only the top-1 prediction is insufficient. We provide qualitative results of how our model makes object-attribute composition prediction on VAW-CZSL in the suppl.

Is disentangling and hallucinating pairs helpful? Prior works rely heavily on word embeddings for this task, but to improve the capabilities of visual systems, it is imperative to explore what is possible in the visual domain. We do an extensive study to understand if our intuition aligns with OADis (Table 4). Here are some takeaways:

- Using only $\mathcal{L}_{\text{cls}}$, we get a benchmark performance based on the architectural contributions, such as *LE* and *ONC*. When $\mathcal{L}_{\text{attr}}$ is added, significant performance boost for attribute accuracy can be seen in Table 4.
- Adding object loss $\mathcal{L}_{\text{obj}}$ with $\mathcal{L}_{\text{cls}}$, makes object accuracy better but no change in Val and Test AUC. This indicates the need of both losses to balance the effects. Using both $\mathcal{L}_{\text{attr}}$ and $\mathcal{L}_{\text{obj}}$ gives improvement in all measures.
- Adding $\mathcal{L}_{\text{seen}}$ results in boost for seen AUC, but drop in Test AUC, which has unseen pairs along with seen pairs. Using unseen loss $\mathcal{L}_{\text{unseen}}$ leads to increase in both Test and attribute accuracy.
- Finally adding unseen composition loss $\mathcal{L}_{\text{unseen}}$ along with seen loss $\mathcal{L}_{\text{seen}}$, the model improves on most metrics. Each loss plays a role and regularizes effects from other losses.

Is visual disentangling actually happening? Visual disentanglement in feature space is challenging to visualize since: (a) parts of an image for attributes and objects are hard to distinguish, as *attributes are aspects of an object*; (b) OADis is end-to-end trained with losses to disentangle features for attribute and object embeddings, which is separate from pair embedding space. Inspired by [30, 38], we show a few qualitative results in Figure 5. Using all training images, prototype features $\mathcal{V}_{\text{attr}}$ for each attribute can be computed by averaging features for all images containing that attributes v_{attr} using *AAN*. Similarly, with *OAN*, prototype object features are also computed. For each test image, we find top-3 nearest neighbors from these prototype fea-

Figure 5. Qualitative results showing top 3 attributes and objects from test images, using prototype disentangled features computed on training data.

tures (Figure 5). Hence, the disentangled prototype features of attributes and objects are used for classifying unseen images. Note that results reported in Table 1 use pair embedding space for attribute and object classification, whereas here we use auxiliary attribute and object embedding spaces (in Figure 3b) for the same task. If disentanglement features are not robust, then composition features will also not be efficient. We also show that using the composition of disentangled features for unseen pairs, relevant images from the test set can be found in suppl.

Limitations. Despite OADis outperforming prior works on all benchmarks, we still notice some outstanding deficiencies in this problem domain. First, similar to [37], OADis often struggles on images containing multiple objects, where it does not know which object to make prediction on. One possible solution is to utilize an object-conditioned attention that allows the model to focus and possibly output attribute for multiple objects. Second, from qualitative studies on VAW-CZSL, we notice there are multiple cases where OADis makes the correct prediction but is considered incorrect by the image label. This is due to the fact that objects in-the-wild are mostly multi-label (containing multiple attributes), which none of the current single-label benchmarks have attempted to address.

4.3. Ablation Studies

In this section, we show experiments to support our design choices for OADis. All the ablations are done for MIT-states [24], for one random seed initialization, and are consistent for other datasets as well. Empirical results for λ, δ and different word embeddings can be found in suppl.

Why Object-Conditioned Network? Label Embedder [38] uses a linear layer and concatenates word embeddings for attributes and objects. We experiment with other networks: MLP with more parameters with two layers and ReLU and Object-conditioned network that uses a residual connection for object embedding. Our intuition is that same attribute contributes differently to each object, *i.e.*, `ruffled bag` is very different from `ruffled flower`. Hence, attributes are conditioned on object. Adding a residual connection for object embeddings to the final attribute embedding helps condition the attribute. We empirically demonstrate that object-conditioning helps in Table 5 (refer to the suppl.).

To augment or not to augment? Augmentation is a common technique to reduce over-fitting and improve generalization. Surprisingly, prior works do not use any image augmentation. OADis without augmentation gives 6.7% AUC on Val and 5.1% AUC on Test set for MIT-states. Hence, we use augmentation for OADis and re-implemented rest of the baselines in Table 2, showing that augmentation helps improving all methods $\sim$1.0-1.5% AUC. We use horizontal flip and random crop as augmentation.

4.4. Qualitative results

To qualitatively analyze our hallucinated compositions, we perform a nearest neighbor search on all three datasets. We pick the unseen compositions composed using the disentangled features, and find their top-5 nearest neighbors from the validation and test set. Figure 4(a) illustrates a few of our results. Note that these pairs are never seen in training. Based on the hallucinated compositions of disentangled attributes and objects, we are able to retrieve samples from these unseen compositions.

In Figure 4(b), we show the top-3 predictions of OADis on VAW-CSZL. Column 1 shows results for seen, and columns 2 and 3 show unseen compositions, with the ground-truth label on top (bold black). In all examples, our top-3 predictions describe the visual content of the images accurately, even though in many cases the ground-truth label is not predicted in top-1. For column 3, we purposely show examples where our model predictions totally differ from the ground-truth label, but still correctly describe the visual information in each image. Similar to [37], this explains the multi-label nature of object-attribute recognition, and why we report top-3 and top-5 metrics for the VAW-CZSL benchmark.

5. Conclusion

In this work, we demonstrated the ability to disentangle object and attribute in the visual feature space, that are used for hallucinating novel complex concepts, as well as regularizing and obtaining a better object-attribute recognition model. Through extensive experiments, we show the efficacy of our method, and surpass previous methods across three different benchmarks. In addition, we also propose a new benchmark for the compositional zero-shot learning task with images of objects in-the-wild, which we believe can help shift the focus of the community towards images in more complex scenes. Finally, we also highlight limitations of our work, including the notable problem of multi-label in object attributes, which we hope would encourage future works to start tackling CSZL for more realistic scenarios.

Acknowledgements. This work was supported by the Air Force (STTR awards FA865019P6014, FA864920C0010), DARPA SAILON program (W911NF2020009) and gifts from Adobe collaboration support fund.

References

[1] Jean-Baptiste Alayrac, Josef Sivic, I. Laptev, and S. Lacoste-Julien. Joint discovery of object states and manipulation actions. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2146–2155, 2017. 1, 2

[2] Y. Atzmon, F. Kreuk, U. Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. *ArXiv*, abs/2006.14610, 2020. 2, 5, 6

[3] Nachwa Abou Bakr, J. Crowley, and Rémi Ronfard. Recognizing manipulation actions from state-transformations. *ArXiv*, abs/1906.05147, 2019. 1, 2

[4] Nachwa Abou Bakr, Rémi Ronfard, and J. Crowley. Recognition and localization of food in cooking videos. In *CEA/MADiMa '18*, 2018. 1, 2

[5] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video, 2020. 1, 2

[6] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94 2:115–147, 1987. 1

[7] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 381–389, 2018. 1

[8] Chao-Yeh Chen and Kristen Grauman. Inferring analogous attributes. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 200–207, 2014. 2

[9] Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12652–12660, 2020. 4

[10] Hui Chen, Zhixiong Nan, Jingjing Jiang, and Nanning Zheng. Learning to infer unseen attribute-object compositions. *arXiv preprint arXiv:2010.14343*, 2020. 2

[11] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. NEIL: Extracting Visual Knowledge from Web Data. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[12] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2020. 2

[13] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, 2009. 3

[14] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric recognition for cross-category generalization. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2352–2359, 2010. 1, 2

[15] Ali Farhadi, Ian Endres, Derek Hoiem, and David Alexander Forsyth. Describing objects by their attributes. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009. 1, 2

[16] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013. 1, 2

[17] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. In *NIPS*, 2007. 2, 3

[18] A. Fire and S. Zhu. Learning perceptual causality from video. *ACM Trans. Intell. Syst. Technol.*, 7:23:1–23:22, 2015. 1, 2

[19] Jianlong Fu, Jinqiao Wang, Xin-Jing Wang, Yong Rui, and Hanqing Lu. What visual attributes characterize an object class? In *ACCV*, 2014. 2

[20] Liane Gabora. The 'power of then': The uniquely human capacity to imagine beyond the present. *arXiv: Neurons and Cognition*, 2015. 2

[21] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[22] Donald D. Hoffman and Whitman Richards. Parts of recognition. *Cognition*, 18:65–96, 1984. 1, 2

[23] Sung Ju Hwang, Fei Sha, and Kristen Grauman. Sharing features between objects and their attributes. *CVPR 2011*, pages 1761–1768, 2011. 2

[24] Phillip Isola, Joseph J. Lim, and E. Adelson. Discovering states and transformations in image collections. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2015. 1, 2, 5, 6, 7, 8

[25] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35:2891–2903, 2013. 2

[26] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 1, 2

[27] Hao-Wei Lee, Chia-Po Wei, and Yu-Chiang Frank Wang. Learning grassmann manifolds for object state discovery. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1223–1227, 2017. 2

[28] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. *ArXiv*, abs/1803.08024, 2018. 4

[29] Kunpeng Li, Yulun Zhang, K. Li, Yuanyuan Li, and Yun Raymond Fu. Visual semantic reasoning for image-text matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4653–4661, 2019. 4

[30] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11313–11322, 2020. 2, 6, 7

[31] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 2

[32] Dhruv Kumar Mahajan, Sundararajan Sellamanickam, and Vinod Nair. A joint learning framework for attribute models and object descriptions. *2011 International Conference on Computer Vision*, pages 1227–1234, 2011. 2

[33] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. *ArXiv*, abs/2105.01017, 2021. 2, 4, 6

[34] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, 2013. 1, 2

[35] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1160–1169, 2017. 2, 3

[36] Ishan Misra*, Abhinav Shrivastava*, Abhinav Gupta, and Martial Hebert. Cross-stitch Networks for Multi-task Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[37] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. *ArXiv*, abs/2102.01987, 2021. 2, 4, 5, 6, 8

[38] Tushar Nagarajan and K. Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *ECCV*, 2018. 2, 3, 6, 7, 8

[39] Z. Nan, Y. Liu, N. Zheng, and S. Zhu. Recognizing unseen attribute-object pair with generative model. In *AAAI*, 2019. 2

[40] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[41] Devi Parikh and Kristen Grauman. Relative attributes. *2011 International Conference on Computer Vision*, pages 503–510, 2011. 2

[42] Genevieve Patterson and James Hays. Coco attributes: Attributes for people, animals, and objects. In *ECCV*, 2016. 2

[43] Joel Pearson. The human imagination: the cognitive neuroscience of visual mental imagery. *Nature Reviews Neuroscience*, pages 1–11, 2019. 2

[44] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 3, 6

[45] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5, 6

[46] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3592–3601, 2019. 2, 3, 6

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 2

[48] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 2

[49] Liyue Shen, Serena Yeung, Judy Hoffman, Greg Mori, and Li Fei-Fei. Scaling human-object interaction recognition through zero-shot learning. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1568–1576, 2018. 1

[50] Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *European Conference on Computer Vision*, 2012. 2

[51] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, X. Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4115–4124, 2020. 1

[52] Kun-Juan Wei, Muli Yang, H. Wang, Cheng Deng, and Xianglong Liu. Adversarial fine-grained composition learning for unseen attribute-object recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3740–3748, 2019. 2

[53] Guangyue Xu, Parisa Kordjamshidi, and Joyce Yue Chai. Zero-shot compositional concept learning. *ArXiv*, abs/2107.05176, 2021. 4

[54] Ziwei Xu, Guangzhi Wang, Yongkang Wong, and Mohan S. Kankanhalli. Relation-aware compositional zero-shot learning for attribute-object pair recognition. *ArXiv*, abs/2108.04603, 2021. 2, 6

[55] Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10245–10253, 2020. 2, 3

[56] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 1, 2, 5, 6