In Proceedings of the European Conference on Computer Vision (ECCV), 2018

Attributes as Operators: Factorizing Unseen Attribute-Object Compositions

Tushar Nagarajan[1] and Kristen Grauman[2]

[1] The University of Texas at Austin
[2] Facebook AI Research
tushar@cs.utexas.edu, grauman@fb.com*

Abstract. We present a new approach to modeling visual attributes. Prior work casts attributes in a similar role as objects, learning a latent representation where properties (e.g., *sliced*) are recognized by classifiers much in the way objects (e.g., *apple*) are. However, this common approach fails to separate the attributes observed during training from the objects with which they are composed, making it ineffectual when encountering new attribute-object compositions. Instead, we propose to model attributes as *operators*. Our approach learns a semantic embedding that explicitly factors out attributes from their accompanying objects, and also benefits from novel regularizers expressing attribute operators' effects (e.g., *blunt* should undo the effects of *sharp*). Not only does our approach align conceptually with the linguistic role of attributes as modifiers, but it also generalizes to recognize unseen compositions of objects and attributes. We validate our approach on two challenging datasets and demonstrate significant improvements over the state of the art. In addition, we show that not only can our model recognize unseen compositions robustly in an open-world setting, it can also generalize to compositions where objects themselves were unseen during training.

1 Introduction

Attributes are semantic descriptions that convey an object's properties—such as its materials, colors, patterns, styles, expressions, parts, or functions. Attributes have proven to be an effective representation for faces and people [26, 36, 44, 49, 29, 45, 32], catalog products [4, 24, 57, 17], and generic objects and scenes [28, 11, 27, 37, 19, 1]. Because they are expressed in natural language, attributes facilitate human-machine communication about visual content, *e.g.*, for applications in image search [26, 24], zero-shot learning [1], narration [25], or image generation [55].

Attributes and objects are fundamentally different entities: objects are physical things (nouns), whereas attributes are properties of those things (adjectives). Despite this fact, existing methods for attributes largely proceed in the same manner as state-of-the-art object recognition methods. Namely, image examples labeled according to the attributes present are used to train discriminative models, *e.g.*, with a convolutional neural network [49, 29, 45, 32, 57, 47].

The latent vector encoding learned by such models is expected to capture an *object-agnostic* attribute representation. Yet, achieving this is problematic, both in terms of

**On leave from University of Texas at Austin (grauman@cs.utexas.edu).*

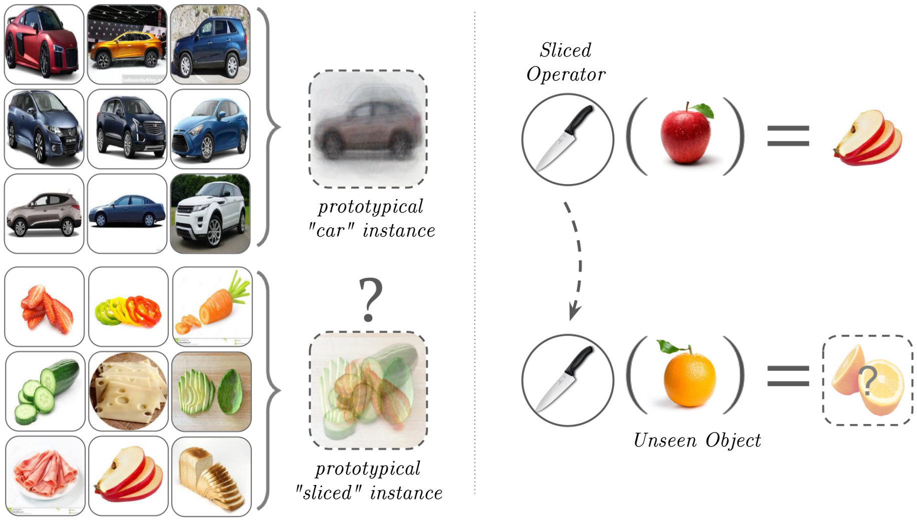


Fig. 1: Conceptual overview of our idea. Left: Unlike for objects, it is difficult to learn a predictable visual prototype for an attribute (*e.g.*, "sliced" as shown here). Furthermore, standard visual recognition pipelines are prone to overfit to those object-attribute pairings observed during training. Right: We propose to model attributes as operators, learning how they *transform objects* rather than what they themselves look like. Once learned, the effects of the attribute operators are generalizable to new, unseen object categories.

data efficiency and generalization. Specifically, it assumes during training that 1) the attribute has been observed in combination with all potential objects (unrealistic and not scalable), and/or 2) an attribute's influence is manifested similarly across all objects (rarely the case, *e.g.*, "old" influences church and shoe differently). We observe that with the attribute's meaning so intrinsically tied to the object it describes, an ideal attribute vector encoding may not exist. See Figure 1, left.

In light of these issues, we propose to model attributes as *operators* — with the goal of learning a model *for attribute-object composition itself* capable of explicitly factoring out the attributes' effect from their accompanying object representations.

First, rather than encode an attribute as a point in some embedding space, we encode it as a (learned) transformation that, when applied to an object encoding, modifies it to appropriately transform its appearance (see Figure 1, right). In particular, we formulate an embedding objective where compositions and images project into the same semantic space, allowing recognition of unseen attribute-object pairings in novel images.[3]

Second, we introduce novel regularizers during training that capitalize on the attribute-as-operator concept. For example, one regularizer requires that the effect of applying an attribute and then its antonym to an object should produce minimal change in the object encoding (e.g., *blunt* should "undo" the effects of *sharp*); another requires commutativ-

[3] We stress that this differs from traditional zero-shot object recognition [28, 21, 1], where an *unseen object* is defined by its (previously learned and class-agnostic) attributes. In our case, we have *unseen compositions* of objects and attributes.

ity when pairs of attributes modify an object (e.g., a *sliced red* apple is equivalent to a *red sliced* apple).

We validate our approach on two challenging datasets: MIT-States [33] and UT-Zappos [56]. Together, they span hundreds of objects, attributes, and compositions. The results demonstrate the advantages of attributes as operators, in terms of the accuracy in recognizing unseen attribute-object compositions. We observe significant improvements over state-of-the-art methods for this task [5, 33], with absolute improvements of 3%-12%. Finally, we show that our method is similarly robust whether identifying unseen compositions on their own or in the company of seen compositions—which is of great practical value for recognition in realistic, open world settings.

2 Related Work

Visual attributes. Early work on visual attributes [26, 28, 11, 36] established the task of inferring mid-level semantic descriptions from images. The research community has since explored many applications for attributes, including image search [26, 24, 44], zero-shot object categorization [28, 21, 1], sentence generation [25] and fashion image analysis [4, 18, 17]. Throughout, the standard approach to learn attributes is very similar to that used to learn object categories: discriminative classifiers with labeled examples. In particular, today's best accuracies are obtained by training a deep convolutional neural network to classify attributes [49, 29, 45, 32, 47]. Multi-task attribute training methods account for correlations between different attributes [32, 23, 19, 44]. Our approach is a fundamental departure from all of the above: rather than consider attribute instances as points in some high-dimensional space that can be classified, we consider attributes as *operators* that transform visual data from one condition to another.

Composition in language and vision. In natural language processing, the composition of adjectives and nouns is modeled as single compositions [13, 34] or transformations (*i.e.*, an adjective transformation applied to the noun vector) [3, 46]. Bridging such linguistic concepts to visual data, some work explores the correlation between similarity scores for color-object pairs in the language and visual domains [35].

Composition in vision has been studied in the context of modeling compound objects [39] (clipboard = clip + board), verb-object interactions [42, 58] (riding a horse = person + riding + horse), and adjective-noun combinations [5, 33, 9] (fluffy towel = towel modified by fluffy). All these approaches leverage the key insight that the characteristics of the composed entities could be very different from their constituents; however, they all subscribe to the traditional notion of representing constituents as vectors, and compositions as black-box modifications of these vectors. Instead, we model compositions as unique operators conditioned on the constituents (*e.g.*, for attribute-object composition, a different modification for each attribute).

Limited prior work on attribute-object compositions considers *unseen compositions*, that is, where each constituent is seen during training, but new unseen compositions are seen at test time [5, 33]. Both methods construct classifiers for composite concepts using pre-trained linear classifiers for the "seen" primitive concepts, either with tensor completion [5] or neural networks [33]. Recent work extends this notion to expressions connected by logical operators [9]. We tackle unseen compositions as well. However, rather than treat attributes and objects alike as classifier vectors and place the burden of learning on a single network, we propose a factored representation of the constituents, modeling attribute-object composition as an attribute-specific invertible *transformation* on object vectors. Our formulation also enables novel regularizers based on the attributes' linguistic meaning. Our model naturally extends to compositions where the objects themselves are unseen during training, unlike [5, 33] which requires an SVM classifier to be trained for every new object. In addition, rather than exclusively predict unseen compositions as in [33], we also study the more realistic scenario where *all* compositions are candidates for recognition.

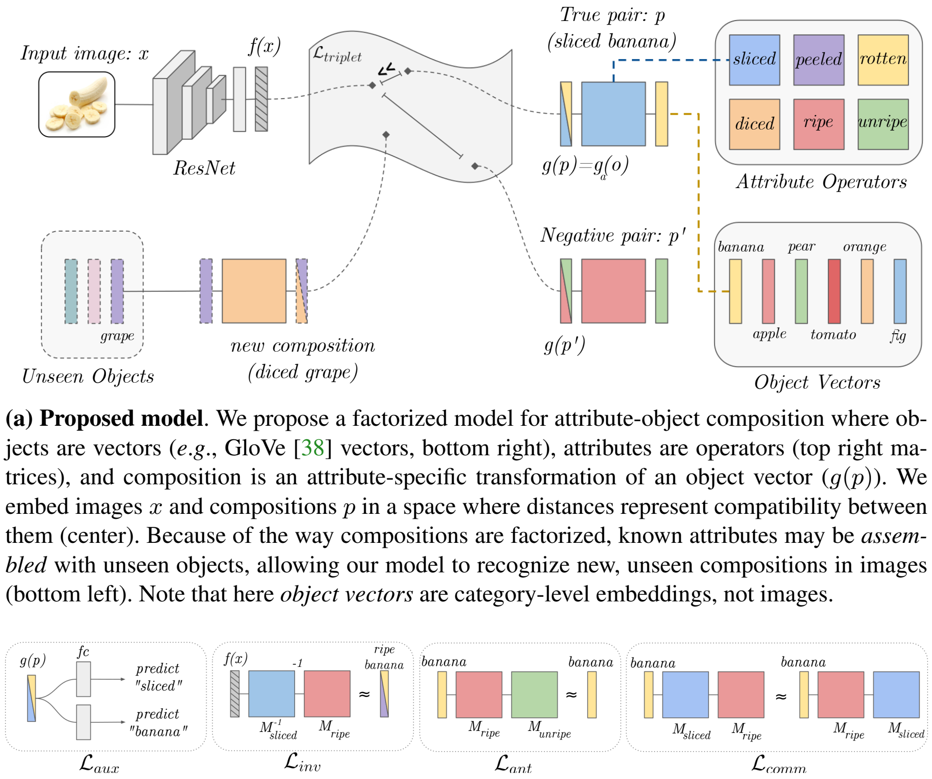
Visual transformations. The notion of visual "states" has been explored from several angles. Given a collection of images [20] or time-lapse videos [60, 27], methods can discover transformations that map between object states in order to create new images or visualize their relationships. Given video input, action recognition can be posed as learning the visual state transformation, *e.g.*, how a person manipulates an object [12, 2] or how activity preconditions map to postconditions [51]. Given a camera transformation, other methods visualize the scene from the specified new viewpoint [22, 59]. While we share the general concept of capturing a visual transformation, we are the first to propose modeling attributes as operators that alter an object's state, with the goal of recognizing unseen compositions.

Low-shot learning with sample synthesis. Recent work explores ways to generate synthetic training examples for classes that rarely occur, either in terms of features [10, 14, 31, 52, 61] or entire images [57, 8]. One part of our novel regularization approach also involves hypothetical attribute-transformed examples. However, whereas prior work explicitly generates samples offline to augment the dataset, our feature generation is an implicit process to regularize learning and works in concert with other novel constraints like inverse consistency or commutativity (see Section 3.3).

3 Approach

Our goal is to identify attribute-object compositions (*e.g.*, sliced banana, fluffy dog) in an image. Conventional classification approaches suffer from the long-tailed distribution of complex concepts [42, 30] and a limited capacity to generalize to unseen concepts. Instead, we model the composition process itself. We factorize out the underlying primitive concepts (attributes and objects) seen during training, and use them as building blocks to identify unseen combinations during inference. Our approach is driven by the fundamental narrative: *if we've seen a sliced orange, a sliced banana, and a rotten banana, can we anticipate what a rotten orange looks like?*

We model the composition process around the functional role of attributes. Rather than treat objects and attributes equally as vectors, we model attributes as invertible operators, and composition as an attribute-conditioned transformation *applied* to object vectors. Our recognition task then turns into an embedding learning task, where we project images and compositions into a common semantic space to identify the composition present. We guide the learning with novel regularizers that are consistent with the linguistic behavior of attributes.



(a) Proposed model. We propose a factorized model for attribute-object composition where objects are vectors (*e.g.*, GloVe [38] vectors, bottom right), attributes are operators (top right matrices), and composition is an attribute-specific transformation of an object vector ($g(p)$). We embed images x and compositions p in a space where distances represent compatibility between them (center). Because of the way compositions are factorized, known attributes may be *assembled* with unseen objects, allowing our model to recognize new, unseen compositions in images (bottom left). Note that here *object vectors* are category-level embeddings, not images.

(b) Proposed regularizers. We propose several regularizers that conform with the linguistic meaning of attributes. $\mathcal{L}_{aux}$ ensures that the identity of the attribute/object is not lost during composition; $\mathcal{L}_{inv}$ *swaps out* attributes to implicitly synthesize new compositions for training; $\mathcal{L}_{ant}$ models antonyms ("unripe" should *undo* the effects of "ripe"); and $\mathcal{L}_{comm}$ models the commutative property of attributes (a *ripe sliced* banana is the same as a *sliced ripe* banana).

Fig. 2: Overview of proposed approach. Best viewed in color

In the following, we start by formally describing the embedding learning problem in Section 3.1. We then describe the details of our embedding scheme for attributes and objects in Section 3.2. We present our optimization objective and auxiliary loss terms in Section 3.3. Finally, we describe our training methodology in Section 3.4.

3.1 Unseen pair recognition as embedding learning

We train a model that learns a mapping from a set of images $\mathcal{X}$ to a set of attribute-object pairs $\mathcal{P} = \mathcal{A} \times \mathcal{O}$. For example, "old-dog" is one attribute-object pairing. We divide the set of pairs into two disjoint sets: $\mathcal{P}_s$, which is a set of pairs that is seen during training and is used to learn a factored composition model, and $\mathcal{P}_u$, which is a set of pairs unseen during training, yet perfectly valid to encounter at test time. While $\mathcal{P}_s$ and $\mathcal{P}_u$ are completely disjoint, their constituent attributes and objects are observed in some (other) composition during training. Our images contain objects with a single attribute label associated with them, *i.e.*, each image has a unique pair label $p \in \mathcal{P}$.

During training, given an image $x \in \mathcal{X}$ and its corresponding pair label $p \in \mathcal{P}_s$, we learn two embedding functions $f(x)$ and $g(p)$ to project them into a *common semantic space*. For $f(x)$, we use a pretrained ResNet18 [15] followed by a linear layer. For $g(p)$, we introduce an attribute-operator model, described in detail in Section 3.2.

We learn the embedding functions such that in this space, the Euclidean distance between the image embedding $f(x)$ and the correct pair embedding $g(p)$ is minimized, while the distance to all incorrect pairs is maximized. Distance in this space represents *compatibility*—*i.e.*, a low distance between an image and pair embedding implies the pair is present in the image. Critically, once $g(p)$ is learned, even an unseen pair can be projected in this semantic space, and its compatibility with an image can be assessed. See Figure 2a.

During inference, we compute and store the pair embeddings of *all* potential pair candidates from $\mathcal{P}$ using our previously learned composition function $g(.)$. When presented with a new image, we embed it as usual using $f(.)$, and identify which of the pair embeddings is closest to it. Note how $\mathcal{P}$ includes both pairs seen in training as well as unseen attribute-object compositions; recognizing the latter would not be possible if we were doing a simple classification among the previously seen combinations.

3.2 Attribute-operator model for composition

As discussed above, the conventional approach treats attributes much like objects, both occupying some point/region in an embedding space [44, 49, 29, 45, 32, 47, 23, 19].

On the one hand, it is meaningful to conjure a latent representation for an "attribute-free object"—for example, *dog* exists as a concept before we specialize it to be a spotted or fluffy dog. In fact, in the psychology of perception, one way to characterize a so-called basic-level category is by its affordance of a single mental prototype [40]. On the other hand, however, it is problematic to conjure an "object-free attribute". What does it mean to map "fluffy" as a concept in a semantic embedding space? What is the visual prototype of "fluffy"? See Figure 1.

We contend that a more natural way of describing attributes is in how they *modify* the objects they refer to. Images of a "dog" and a "fluffy dog" help us estimate what the concept "fluffy" refers to. Moreover, these modifications are strongly conditioned on the object they describe ("fluffy" exhibits itself significantly differently in "fluffy dog" compared to "fluffy pillow"). In this sense, attribute behavior bears some resemblance to geometric transformations. For example, rotation can be perfectly represented as an orthogonal matrix acting on a vector. Representing rotation as a vector, and its action as some additional function, would be needlessly complicated and unintuitive.

With this in mind, we represent each object category $o \in \mathcal{O}$ as a D-dimensional vector, which denotes a prototypical object instance. Specifically, we use GloVe word embeddings [38] for the object vector space. Each attribute $a \in \mathcal{A}$ is a parametrized function $g_a : \mathcal{R}^D \rightarrow \mathcal{R}^D$ that modifies an object representation to exhibit that attribute, and brings it to the semantic space where images reside. For simplicity, we consider a linear transform for g_a, represented by a $D \times D$ matrix M_a:

$$g(p) = g_a(o) = M_a o, \tag{1}$$

though the proposed framework (excluding the inverse consistency regularizer) naturally supports more complex functions for g_a as well. See Figure 2a, top right.

Interesting properties arise from our attribute-operator design. First, factorizing composition as a matrix-vector product facilitates transfer: an unseen pair can be represented by applying a learned attribute operator to an appropriate object vector (Figure 2a, bottom left). Secondly, since images and compositions reside in the same space, it is possible to *remove* attributes from an image by applying the inverse of the transformation; multiple attributes can be applied consecutively to images; and the structure of the attribute space can be coded into how the transformations behave. Below we discuss how we leverage these properties to regularize the learning process (Sec. 3.3).

3.3 Learning objective for attributes as operators

Our training set consists of n images and their pair labels, $\{(x_1, p_1), \ldots, (x_n, p_n)\}$. We design a loss function to efficiently learn to project images and composition pairs to a common embedding space. We begin with a standard triplet loss. The loss for an image x with pair label $p = (a, o)$ is given by:

$$\mathcal{L}_{triplet} = \max\left(0, d(f(x), M_a o) - d(f(x), M_{a'} o') + m\right), \forall\, a' \neq a \vee o' \neq o, \quad (2)$$

where d denotes Euclidean distance, and m is the margin value, which we keep fixed at 0.5 for all our experiments. In other words, the embedded image ought to be closer to its object transformed by the specified attribute a than other attribute-object pairings.

Thus far, the loss is similar in spirit to embedding based zero-shot learning methods [54], and more generally to triplet-loss based representation learning methods [7, 16, 43]. We emphasize that our focus is on learning a model for the composition operation; a triplet-loss based embedding is merely an appropriate framework that facilitates this. In the following, we extend this framework to effectively accommodate attributes as operators and inject our novel linguistic-based regularizers.

Object and attribute auxiliaries. In our model, both the attribute operator and object vector, and thereby their composition, are learnable parameters. It is possible that one element of the composition (either attributes or objects) will dominate during optimization, and try to capture all the information instead of learning a factorized model. This could lead to a composition representation, where one component does not adequately feature. To address this, we introduce an auxiliary loss term that forces the composed representation to be discriminative, *i.e*., it must be able to predict both the attribute and object involved in the composition:

$$\mathcal{L}_{aux} = -\sum_{i \in \mathcal{A}} \delta_{ai}\, log(p_a^i) - \sum_{i \in \mathcal{O}} \delta_{oi}\, log(p_o^i), \quad (3)$$

where $\delta_{yi} = 1$ iff $y = i$, and p_a and p_o are the outputs of softmax linear classifiers trained to discriminate the attributes and objects, respectively. This auxiliary supervision ensures that the identity of the attribute and the object are not lost in the composed representation—in effect, strongly incentivizing a factorized representation.

Inverse consistency. We exploit the invertible nature of our attributes to implicitly synthesize new training instances to regularize our model further. More specifically, we *swap out* an actual attribute a from the training example for a randomly selected one a', and construct another triplet loss term to account for the new composition:

$$\begin{aligned} &f(x') := M_{a'} M_a^{-1} f(x) \\ &\mathcal{L}_{inv} = \max\left(0, d(f(x'), M_{a'}o) - d(f(x'), M_a o) + m\right), \end{aligned} \tag{4}$$

where the triplet loss notation is in the same form as Eq 2.

Here $M_{a'} M_a^{-1}$ represents the removal of attribute a to arrive at the "prototype object" description of an image, and then the application of attribute a' to imbue the object with a new attribute. As a result, $f(x')$ represents a pseudo-instance with a new attribute-object pair, helping the model generalize better.

The pseudo-instances generated here are inherently noisy, and factoring them in directly (as a new instance) may obstruct training. To mitigate this, we select our negative example to target the more direct, and thus simpler consequence of this swapping. For example, when we swap out "sliced" for "ripe" from a *sliced banana* to make a *ripe banana*, we focus on the more obvious fact—that it is no longer "sliced"—by picking the original composition (*sliced banana*) as the negative, rather than sampling a completely new one.

Commutative attribute operators. Next we constrain the attributes to respect the commutative property. For example, applying the "sliced" operator after the "ripe" operator is the same as applying "ripe" after "sliced", or in other words a *ripe sliced* banana is the same as a *sliced ripe* banana. This commutative loss is expressed as:

$$\mathcal{L}_{comm} = \sum_{a,b \in \mathcal{A}} \left\| M_a(M_b o) - M_b(M_a o) \right\|_2 . \tag{5}$$

This loss forces the attribute transformations to respect the notion of *attribute composability* we observe in the context of language.

Antonym consistency. The final linguistic structure of attributes we aim to exploit is antonyms. For example, we hypothesize that the "blunt" operator should *undo* the effects of the "sharp" operator. To that end, we consider a loss term that operates over pairs of antonym attributes (a, a'):

$$\mathcal{L}_{ant} = \sum_{a,a' \in \mathcal{A}} \left\| M_{a'}(M_a o) - o \right\|_2 . \tag{6}$$

For the MIT-States dataset (cf. Sec. 4), we manually identify 30 antonym pairs like ancient/modern, bent/straight, blunt/sharp. Figure 2b recaps all the regularizers.

3.4 Training and inference

We minimize the combined loss function ($\mathcal{L}_{triplet} + \mathcal{L}_{aux} + \mathcal{L}_{inv} + \mathcal{L}_{comm} + \mathcal{L}_{ant}$) over all the training images, and train our network end to end. The learnable parameters are: the linear layer for $f(x)$, the matrices for every attribute M_a, $\forall a \in \mathcal{A}$, the object vectors $\forall o \in \mathcal{O}$ and the two fully-connected layers for the auxiliary classifiers.

During training, we embed each labeled image x in a semantic space using $f(x)$, and apply its attribute operator g_a to its object vector o to get a composed representation

$g_a(o)$. The triplet loss pushes these two representations close together, while pushing incorrect pair embeddings apart. Our regularizers further make sure compositions are discriminative; attributes obey the commutative property; they undo the effects of their antonyms; and we implicitly synthesize instances with new compositions.

For inference, we compute and store the embeddings for all candidate pairs, $g_a(o)$, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$. When a new image q arrives, we sort the pre-computed embeddings by their distance to the image embedding $f(q)$, and identify the compositions with the lowest distances. The distance calculations can be performed quickly on our dataset with a few thousand pairs. Intelligent pruning strategies may be employed to reduce the search space for larger attribute/object vocabularies. We stress that the novel image can be assigned to an unseen composition absent in training images.We evaluate accuracy on the nearest composition $\hat{p_q} = (o_q, a_q)$ as our datasets support instances with single attributes.

4 Experiments

Our experiments explore the impact of modeling attributes as operators, particularly for recognizing unseen combinations of objects and attributes.

4.1 Experimental setup

Datasets. We evaluate our method on two datasets:

- **MIT-States** [20]: This dataset has 245 object classes, 115 attribute classes and $\sim$53K images. There is a wide range of objects (*e.g.*, *fish*, *persimmon*, *room*) and attributes (*e.g.*, *mossy*, *deflated*, *dirty*). On average, each object instance is modified by one of the 9 attributes it affords. We use the *compositional* split described in [33] for our experiments, resulting in disjoint sets of pairs—about 1.2K pairs in $\mathcal{P}_s$ for training and 700 pairs in $\mathcal{P}_u$ for testing.
- **UT-Zappos50k** [57]: This dataset contains 50K images of shoes with attribute labels. We consider the subset of $\sim$33K images that contain annotations for material attributes of shoes (*e.g.*, *leather*, *sheepskin*, *rubber*); see Supp. The object labels are shoe types (*e.g.*, *high heel*, *sandal*, *sneaker*). We split the data randomly into disjoint sets, yielding 83 pairs in $\mathcal{P}_s$ for training and 33 pairs in $\mathcal{P}_u$ for testing, over 16 attribute classes and 12 object classes.

The datasets are complementary. While MIT-States covers a wide array of everyday objects and attributes, UT-Zappos focuses on a fine-grained domain of shoes. In addition, object annotations in MIT-States are very sparse (some classes have just 4 images), while the UT-Zappos subset has at least 200 images per object class.

Evaluation metrics. We report top-1 accuracy on recognizing pair compositions. We report this accuracy in two forms: (1) Over only the unseen pairs, which we refer to as the **closed world setting**. During test time, we compute the distance between our image embedding and only the pair embeddings of the unseen pairs $\mathcal{P}_u$, and select the nearest one. The closed world setting artificially reduces the pool of allowable labels at test time to *only* the unseen pairs. This is the setting in which [33] report their results.

(2) Over both seen and unseen pairs, which we call the **open world setting**. During test time, we consider all pair embeddings in $\mathcal{P}$ as candidates for recognition. This is more realistic and challenging, since no assumptions are made about the compositions present. We aim for high accuracy in both these settings. We report the *harmonic mean* of these accuracies given by $\textit{h-mean} = 2 * (\textit{open} * \textit{closed})/(\textit{open} + \textit{closed})$, as a consolidated metric. Unlike the arithmetic mean, it penalizes large performance discrepancies between settings. The harmonic mean is recommended to handle a similar discrepancy between seen/unseen accuracies in "generalized" zero-shot learning [54], and is now widely adopted as an evaluation metric [48, 53, 6, 50].

Implementation details. For all experiments, we use an ImageNet [41] pretrained ResNet-18 [15] for $f(x)$. For fair comparison, we do not finetune this network. We project our images and compositions to a $D = 300$-dim. embedding space. We initialize our object and attribute embeddings with GloVe [38] word vectors where applicable, and initialize attribute operators with the identity matrix as this leads to more stable training. All models are implemented in PyTorch. ADAM with learning rate $1e-4$ and batch size 512 is used. The attribute operators are trained with learning rate $1e-5$ as they encounter larger changes in gradient values. Our code is available at github.com/attributes-as-operators.

Baselines and existing methods. We compare to the following methods:

- **VisProd** uses independent classifiers on the image features to predict the attribute and object. It represents methods that do not explicitly model the composition operation. The probability of a pair is simply the product of the probability of each constituent: $P(a, o) = P(a)P(o)$. We report two versions, differing in the choice of the classifier used to generate the aforementioned probabilities: VisProd(SVM) uses a Linear SVM (as used in [33]), and VisProd(NN) uses a single layer softmax regression model.
- **AnalogousAttr** [5] trains a linear SVM classifier for each seen pair, then uses Bayesian Probabilistic Tensor Factorization (BPTF) to infer classifier weights for unseen compositions. We use the same existing code[4] as [5] to recreate this model.
- **RedWine** [33] trains a neural network to transform linear SVMs for the constituent concepts into classifier weights for an unseen combination. Since the authors' code was not available, we implement it ourselves following the paper closely. We train the SVMs with image features consistent with our models. We verify we could reproduce their results with VGG (network they employed), then upgrade its features to ResNet to be more competitive with our approach.
- **LabelEmbed** is like the RedWine model, except it composes word vector representations rather than classifier weights. We use pretrained GloVe [38] word embeddings. This is the LabelEmbed baseline designated in [33].
- **LabelEmbed+** is an improved version of LabelEmbed where (1) We embed both the constituent inputs *and* the image features using feed-forward networks into a semantic embedding space of dimension D, and (2) We allow the input representations to be optimized during training. See Supp. for details.

[4] https://www.cs.cmu.edu/~lxiong/bptf/bptf.html

	MIT-States				UT-Zappos			
	closed	open	+obj	h-mean	closed	open	+obj	h-mean
Chance	0.1	0.05	0.9	0.1	3.0	0.9	6.3	1.3
VisProd(SVM)	11.1	2.4	21.6	3.9	46.8	4.1	17.8	7.5
VisProd(NN)	13.9	2.8	22.6	4.7	**49.9**	4.8	18.1	8.8
AnalogousAttr [5]	1.4	0.2	22.4	0.4	18.3	3.5	16.9	5.9
RedWine [33]	12.5	3.1	18.3	5.0	40.3	2.1	10.5	4.0
LabelEmbed	13.4	3.3	18.8	5.3	25.8	5.2	11.1	8.7
LabelEmbed+	**14.8**	5.7	27.2	8.2	37.4	9.4	19.4	15.0
Ours	12.0	**11.4**	**49.3**	**11.7**	33.2	**23.4**	**38.3**	**27.5**

Table 1: Accuracy (%) on unseen pair detection. Our method outperforms all previous methods in the open world setting. It also is strongest in the consolidated harmonic mean (h-mean) metric that accounts for both the open and closed settings. Our method's gain is significantly wider when we eliminate the pressure caused by scarce object training data, by providing oracle object labels during inference to all methods ("+obj"). The harmonic mean is calculated over the open and closed settings only (it does not factor in +obj).

To our knowledge [5, 33] are the most relevant methods for comparison, as they too address recognition of unseen object-attribute pairs. For all methods, we use the same ResNet-18 image features used in our method; this ensures any performance differences can be attributed to the model rather than the CNN architecture. For all neural models, we ensure that the number of parameters and model capacity are similar to ours.

4.2 Quantitative results: recognizing object-attribute compositions

Detecting unseen compositions. Table 1 shows the results. Our method outperforms all previously reported results and baselines on both datasets by a large margin—around 6% on MIT-States and 14% on UT-Zappos in the open world setting—indicating that it learned a strong model for visual composition.

The absolute accuracies on the two datasets are fairly different. Compared to UT-Zappos, MIT-States is more difficult owing to a larger number of attributes, objects, and unseen pairs. Moreover, it has fewer training examples for primitive object concepts, leading to a lower accuracy overall.

Indeed, if an oracle provides the true object label on a test instance, the accuracies are much more consistent across both datasets ("+obj" in Table 1). This essentially trims the search space down to the attribute afforded by the object in question, and serves as an upper bound for each method's accuracy. On MIT-States, without object labels, the gap between the strongest baseline and our method is about 6%, which widens significantly to about 22% when object labels are provided (to all methods). On UT-Zappos, all methods improve with the object oracle, yet the gap is more consistent with and without (14% vs. 19%). This is consistent with the datasets' disparity in label distribution; the model on UT-Zappos learns a good object representation by itself.

AnalogousAttr [5] varies significantly between the two datasets; it relies on having a partially complete set of compositions in the form of a tensor, and uses that information to "fill in the gaps". For UT-Zappos, this tensor is 43% complete, making completion a relatively simpler task compared to MIT-States, where the tensor is only

	MIT-States			UT-Zappos		
	closed	open	h-mean	closed	open	h-mean
BASE	**14.2**	2.1	3.7	**46.2**	13.1	20.4
+INV	14.0	2.7	4.5	45.7	14.2	21.7
+AUX	10.3	9.5	9.9	33.2	26.5	29.5
+AUX+INV	10.4	9.8	10.1	33.1	26.2	29.2
+AUX+COMM	11.4	10.8	11.1	38.1	**29.7**	**33.4**
+AUX+ANT	8.9	8.8	8.8	-	-	-
+AUX+INV+COMM	12.0	**11.4**	**11.7**	33.2	23.4	27.5

Table 2: Ablation study of regularizers used. The auxiliary classifier loss is essential to our method. Adding other regularizers that are consistent with how attributes function also produces boosts in accuracy in most cases, highlighting the merit of thinking of *attributes as operators*.

4% complete. We believe that over-fitting due to this extreme sparsity is the reason we observe low accuracies for ANALOGOUSATTR on this dataset.

In the closed world setting, our method does not perform as well as some of the other baselines. However, this setting is contrived and arguably a weaker indication of model performance. In the closed world, it is easy for a method to produce biased results due to the artificially pruned label space during inference. For example, the attribute "young" occurs in only *one* unseen composition during test time—"young iguana". Since all images during test time that contain iguanas *are* of "young iguanas", an attribute-blind model is also perfectly capable of classifying these instances correctly, giving a false sense of accuracy. In practical applications, the separation into seen and unseen pairs arises from natural data scarcity. In that setting, the ability to identify unseen compositions *in the presence of known compositions*, *i.e.*, the open world, is a critical metric.

The lower performance in the closed world appears to be a side-effect of preventing overfitting to the subset of closed-world compositions. All models except ours have a large difference between the closed and open world accuracy. Our model operates robustly in both settings, maintaining similar accuracies in each. Our model outperforms the other models in the harmonic mean metric as well by about 3% and 12% on MIT-States and UT-Zappos, respectively.

Effect of regularizers. Table 2 examines the effects of each proposed regularizer on the performance of our model. We see that the auxiliary classification loss stabilizes the learning process significantly, and results in a large increase in accuracy on both datasets. For MIT-States, including the inverse consistency and the commutative operator regularizers provide small boosts and a reasonable increase when used together. For UT-Zappos, the effect of inverse consistency is less pronounced, possibly because the abundance of object training data makes it redundant. The commutative regularizer provides the biggest improvement of 4%. Antonym consistency is not very helpful on MIT-States, perhaps due to the wide visual differences between some antonyms. For example, "ripe" and "unripe" for fruits produce vibrant color changes, and *undoing* one color change does not directly translate to *applying* the other *i.e.*, "ripe" may not be the *visual inverse* of "unripe".[5] These ablation experiments show the merits of pushing our model to be consistent with how attributes operate.

[5] Attributes for UT-Zappos are centered around materials of shoes (*leather*, *cotton*) and so lack antonyms, preventing us from experimenting with that regularizer.

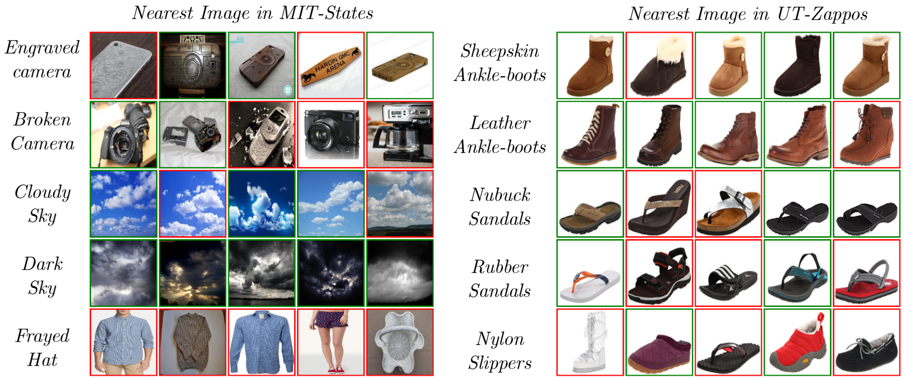


Fig. 3: Top retrieval results for unseen compositions. Unseen compositions are posed as textual queries on MIT-States (left) and UT-Zappos (right). These attribute-object pairs are completely unseen during training; the representation for them is generated using our factored composition model. We highlight correctly retrieved instances with a green border, and incorrect ones with red. Last row shows failure cases.

Overall, the results on two challenging and diverse datasets strongly support our idea to model attributes as operators. Our method consistently outperforms state-of-the-art methods. Furthermore, we see the promise of injecting novel linguistic/semantic operations into attribute learning.

4.3 Qualitative results: retrieving images for unseen descriptions

Next, we show examples of our approach at work to recognize unseen compositions.

Image retrieval for unseen compositions. With a learned composition model in place, our method can retrieve relevant images for textual queries for object-attribute pairs unseen during training. The query itself is in the form of an attribute a and an object o; we embed them, and all the image candidates x, in our semantic space, and select the ones that are nearest to our desired composition. We stress that these compositions are completely new and arise from our model's factored representation of composition.

Figure 3 shows examples. The query is shown in text, and the top 5 nearest images in embedding space are shown alongside. Our method accurately distinguishes between attribute "states" of the same object to retrieve relevant images for the query. The last row shows failure cases. We observe characteristic failures for compositions involving some under-represented object classes in training pairs. For example, compositions involving "hat" are poorly learned as it features in only two training compositions. We also observe common failures involving ambiguous labels (examples of *moldy bread* are also often *sliced* in the data).

Image retrieval for out-of-domain compositions. Figure 4 takes this task two steps further. First, we perform retrieval on an image database disjoint from training to demonstrate robustness to domain shift in the open world setting. Figure 4 (left) shows retrievals from the ImageNet validation set, a set of 50K images disjoint from MIT-States. Even across this dataset, our model can retrieve images with unseen compositions. As

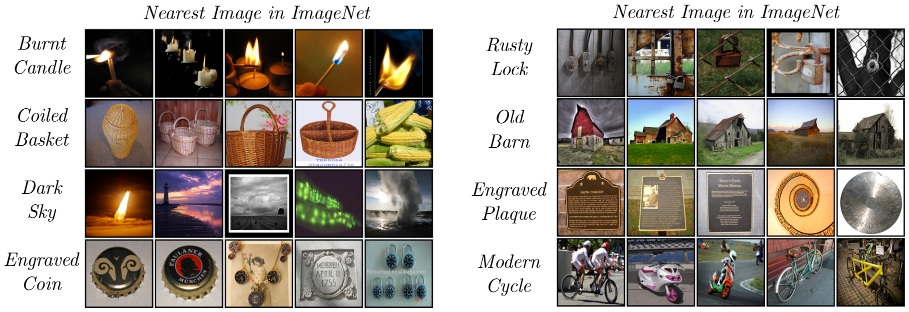


Fig. 4: Top retrieval results in the out-of-domain setting. Images are retrieved from an unseen domain, ImageNet. Left: Our method can successfully retrieve unseen compositions from images in the wild. Right: Retrievals on out-of-domain compositions. Compositions involving objects that are not even present in our dataset (like lock and barn) can be retrieved using our model's factorized representation.

to be expected, there is much more variation. For example, bottle-caps in ImageNet—an object class that is not present in MIT-States—are misconstrued as coins.

Second, we perform retrieval on the disjoint database *and* issue queries for compositions that are in neither the training nor test set. For example, the objects *barn* or *cycle* are never seen in MIT-States, under any attribute composition. We refer to these compositions as *out-of-domain*. Our method handles them by applying attribute operators to GloVe object vectors. Figure 4 (right) shows examples. This generalization is straightforward with our method, whereas it is prohibited by the existing methods RedWine [33] and AnalogousAttr [5]. They rely on having pre-trained SVMs for all constituent concepts. In order to allow an out-of-domain composition with a new object category, those methods would need to gather labeled images for that object, train an SVM, and repeat their full training pipelines.

5 Conclusion

We presented a model of attribute-object composition built around the idea of "attributes as operators". We modeled this composition as an attribute-conditioned transformation of an object vector, and incorporated it into an embedding learning model to identify unseen compositions. We introduced several linguistically inspired auxiliary loss terms to regularize training, all of which capitalize on the operator model for attributes. Experiments show considerable gains over existing models. Our method generalizes well to unseen compositions, in open world, closed world, and even out-of-domain settings. In future work we plan to explore extensions to accommodate relative attribute comparisons and to deal with compositions involving multiple attributes.

Acknowledgments: This research is supported in part by ONR PECASE N00014-15-1-2291 and an Amazon AWS Machine Learning Research Award. We gratefully acknowledge Facebook for a GPU donation.

References

1. Al-Halah, Z., Tapaswi, M., Stiefelhagen, R.: Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In: CVPR (2016) 1, 2, 3
2. Alayrac, J.B., Sivic, J., Laptev, I., Lacoste-Julien, S.: Joint discovery of object states and manipulating actions. ICCV (2017) 4
3. Baroni, M., Zamparelli, R.: Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In: EMNLP (2010) 3
4. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: ECCV (2010) 1, 3
5. Chen, C.Y., Grauman, K.: Inferring analogous attributes. In: CVPR (2014) 3, 4, 10, 11, 14
6. Chen, L., Zhang, H., Xiao, J., Liu, W., Chang, S.F.: Zero-shot visual recognition using semantics-preserving adversarial embedding network. In: CVPR (2018) 10
7. Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N.: Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: CVPR (2016) 7
8. Choe, J., Park, S., Kim, K., Park, J.H., Kim, D., Shim, H.: Face generation for low-shot learning using generative adversarial networks. In: ICCVW (2017) 4
9. Cruz, R.S., Fernando, B., Cherian, A., Gould, S.: Neural algebra of classifiers. WACV (2018) 3
10. Dixit, M., Kwitt, R., Niethammer, M., Vasconcelos, N.: Aga: Attribute-guided augmentation. In: CVPR (2017) 4
11. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009) 1, 3
12. Fathi, A., Rehg, J.M.: Modeling actions through state changes. In: CVPR (2013) 4
13. Guevara, E.: A regression model of adjective-noun compositionality in distributional semantics. In: ACL Workshop on GEometrical Models of Natural Language Semantics (2010) 3
14. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: ICCV (2017) 4
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 6, 10
16. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: SIMBAD (2015) 7
17. Hsiao, W.L., Grauman, K.: Learning the latent look: Unsupervised discovery of a style-coherent embedding from fashion images. ICCV (2017) 1, 3
18. Huang, J., Feris, R., Chen, Q., Yan, S.: Cross-domain image retrieval with a dual attribute-aware ranking network. In: ICCV (2015) 3
19. Huang, S., Elhoseiny, M., Elgammal, A., Yang, D.: Learning hypergraph-regularized attribute predictors. In: CVPR (2015) 1, 3, 6
20. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: CVPR (2015) 4, 9
21. Jayaraman, D., Grauman, K.: Zero-shot recognition with unreliable attributes. In: NIPS (2014) 2, 3
22. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: ICCV (2015) 4
23. Jayaraman, D., Sha, F., Grauman, K.: Decorrelating semantic visual attributes by resisting the urge to share. In: CVPR (2014) 3, 6
24. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Image search with relative attribute feedback. In: CVPR (2012) 1, 3
25. Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Babytalk: Understanding and generating simple image descriptions. TPAMI (2013) 1, 3

26. Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: A search engine for large collections of images with faces. In: ECCV (2008) 1, 3
27. Laffont, P.Y., Ren, Z., Tao, X., Qian, C., Hays, J.: Transient attributes for high-level understanding and editing of outdoor scenes. SIGGRAPH (2014) 1, 4
28. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009) 1, 2, 3
29. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015) 1, 3, 6
30. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: ECCV (2016) 4
31. Lu, J., Li, J., Yan, Z., Zhang, C.: Zero-shot learning by generating pseudo feature representations. arXiv preprint arXiv:1703.06389 (2017) 4
32. Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., Feris, R.: Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: CVPR (2017) 1, 3, 6
33. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: Composition with context. In: CVPR (2017) 3, 4, 9, 10, 11, 14
34. Mitchell, J., Lapata, M.: Vector-based models of semantic composition. ACL: HLT (2008) 3
35. Nguyen, D.T., Lazaridou, A., Bernardi, R.: Coloring objects: adjective-noun visual semantic compositionality. In: ACL Workshop on Vision and Language (2014) 3
36. Parikh, D., Grauman, K.: Relative attributes. In: ICCV (2011) 1, 3
37. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: CVPR (2012) 1
38. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: EMNLP (2014) 5, 6, 10
39. Pezzelle, S., Shekhar, R., Bernardi, R.: Building a bagpipe with a bag and a pipe: Exploring conceptual combination in vision. In: ACL Workshop on Vision and Language (2016) 3
40. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyes-Braem, P.: Basic objects in natural categories. Cognitive psychology (1976) 6
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015) 10
42. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: CVPR (2011) 3, 4
43. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015) 7
44. Siddiquie, B., Feris, R.S., Davis, L.S.: Image ranking and retrieval based on multi-attribute queries. In: CVPR (2011) 1, 3, 6
45. Singh, K.K., Lee, Y.J.: End-to-end localization and ranking for relative attributes. In: ECCV (2016) 1, 3, 6
46. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013) 3
47. Su, C., Zhang, S., Xing, J., Gao, W., Tian, Q.: Deep attributes driven multi-camera person re-identification. In: ECCV (2016) 1, 3, 6
48. Verma, V.K., Arora, G., Mishra, A., Rai, P.: Generalized zero-shot learning via synthesized examples. In: CVPR (2018) 10
49. Wang, J., Cheng, Y., Schmidt Feris, R.: Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In: CVPR (2016) 1, 3, 6
50. Wang, Q., Chen, K.: Alternative semantic representations for zero-shot human action recognition. In: ECML (2017) 10
51. Wang, X., Farhadi, A., Gupta, A.: Actions˜ transformations. In: CVPR (2016) 4

52. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. CVPR (2018) 4
53. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: CVPR (2018) 10
54. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning-the good, the bad and the ugly. CVPR (2017) 7, 10
55. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: ECCV (2016) 1
56. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: CVPR (2014) 3
57. Yu, A., Grauman, K.: Semantic jitter: Dense supervision for visual comparisons via synthetic images. In: ICCV (2017) 1, 4, 9
58. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: CVPR (2017) 3
59. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: ECCV (2016) 4
60. Zhou, Y., Berg, T.L.: Learning temporal transformations from time-lapse videos. In: ECCV (2016) 4
61. Zhu, Y., Elhoseiny, M., Liu, B., Elgammal, A.: Imagine it for me: Generative adversarial approach for zero-shot learning from noisy texts. CVPR (2018) 4