Improving Closed and Open-Vocabulary Attribute Prediction using Transformers

Khoi Pham[1⋆], Kushal Kafle[2], Zhe Lin[2], Zhihong Ding[2], Scott Cohen[2], Quan Tran[2], and Abhinav Shrivastava[1]

[1] University of Maryland, College Park
{khoi,abhinav}@cs.umd.edu
[2] Adobe Research
{kkafle,zlin,zhding,scohen,qtran}@adobe.com

Abstract. We study recognizing attributes for objects in visual scenes. We consider attributes to be any phrases that describe an object's physical and semantic properties, and its relationships with other objects. Existing work studies attribute prediction in a closed setting with a fixed set of attributes, and implements a model that uses limited context. We propose TAP, a new Transformer-based model that can utilize context and predict attributes for multiple objects in a scene in a single forward pass, and a training scheme that allows this model to learn attribute prediction from image-text datasets. Experiments on the large closed attribute benchmark VAW show that TAP outperforms the SOTA by 5.1% mAP. In addition, by utilizing pretrained text embeddings, we extend our model to OpenTAP which can recognize novel attributes not seen during training. In a large-scale setting, we further show that OpenTAP can predict a large number of seen and unseen attributes that outperforms large-scale vision-text model CLIP by a decisive margin. The project page is available at https://vkhoi.github.io/TAP

Keywords: Attribute prediction; Open-vocabulary; Transformer

1 Introduction

Accurately describing object attributes plays a key role in a variety of computer vision challenges. Among many others, some uses of attribute prediction include image retrieval from text [25], referring expression and object selection [29, 59]. They also form arguably the central part of vision and language problems such as visual question answering (VQA) [2, 26, 27], and image captioning [4].

While implicitly required and tackled by numerous downstream tasks, research in attribute for objects in the wild is still under-explored. Existing work is mostly limited to attributes in specific domains such as scenes [69], animals [62], clothing [16, 36], and humans [33, 30, 37]. In recent years, several datasets provide explicit annotations of object attributes, such as [31, 46]. However, they

⋆ A portion of this work was done during Khoi Pham's internship at Adobe Research.

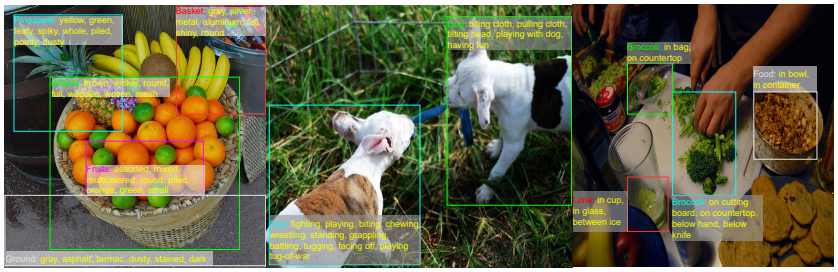

Fig. 1. Attributes in LSA. Attributes in LSA cover a wide-range of words/phrases that describe an object, including (a) adjective, (b) verb to describe action, (c) verb-object pairs to describe interaction, and (c) preposition-object to describe location.

are still limited in terms of their coverage of objects and unique attributes, with even the largest datasets only consisting of a few hundreds of attributes.

Additionally, existing work considers attributes to only include adjective properties, and exclude their interactions with other objects in the scene. The latter is often classified as *visual relationship* and is dedicated to an entirely different research topic [38, 63, 66] which *requires* localization of both subject and object in a subject-predicate-object triplet. We believe this distinction is unnecessarily limiting, *e.g.*, *person wearing hat* conveys information about the property of *person* that is useful even if exact grounding of *hat* is unknown. Hence, we expand the definition of attributes to include adjective- as well as action- and interaction-based properties from the point-of-view of an object.

To this end, we first describe a pipeline to extract object-centric attributes and interactions from large quantities of grounded, weakly grounded, and ungrounded image-text pairs. Then, we propose a novel attribute prediction model called Transformer for Attribute Prediction (TAP). TAP can predict an order of magnitude larger number of unique attributes than previous methods, matching performance of supervised baselines when directly transfer to the VAW benchmark [46]. After finetuning, we outperform prior art by 5.1% mAP and 5.0% mean recall. Furthermore, our model design allows an easy extension to an open-ended attribute prediction branch which we call OpenTAP, by using pretrained text embeddings. OpenTAP can recognize seen attributes, or unseen attributes described by arbitrary text. In our large-scale benchmark, for previously seen attributes, OpenTAP achieves 12.27% mAP higher than CLIP [50], a large-scale image-text matching and zero-shot image classification method, and maintains its superior performance even on attributes that are not seen during training.

In summary, our major contributions are:

– We extend attribute recognition from predicting solely adjectival and action attributes to predicting a larger set that also comprises object interaction. To this end, we propose a new Large-Scale Attribute (LSA) dataset comprising attributes extracted from multiple image-text datasets.

- We propose TAP, a Transformer-based Attribute Prediction model that can effectively utilize the scene context and efficiently make attribute prediction for all objects within an image in a single forward pass, even at the absence of strong grounding information (*i.e.*, object bounding box).
- We propose OpenTAP, a simple extension of TAP to allow open-vocabulary prediction of arbitrary attributes, including those not seen during training.
- We demonstrate state-of-the-art performance on the closed-set attribute prediction dataset (VAW [46]), human-object interaction classification (HICO [6]), as well as a superior performance in our open-vocabulary attribute prediction experiment compared to the recent CLIP model.

2 Related Work

Our work is related to a variety of visual attribute prediction works [12, 11, 3, 44, 55, 37, 45, 31, 16, 46]. While these often target domain-specific attribute (for constrained set of objects, *e.g. clothing*) or small set of attribute classes, our work differs in three points. First, we extract a large number of attributes from public image-text datasets to be used for training. Second, we propose a training scheme that allows our model to make truly large-scale attribute prediction (orders of magnitude larger than prior works) for unconstrained set of objects. Third, our model can be extended to predict unseen attributes, making it also a zero-shot attribute prediction model. Note that this is different from compositional zero-shot [40, 42, 49, 41, 51] which tackles unseen object-attribute composition.

Our work shares background with vision-language (VL) models [39, 56, 8, 32, 50, 67, 28] that need to encode object properties and interaction for learning language-grounded visual information. While the goal of these works is to achieve better performance on downstream VL tasks (*e.g.*, VQA, phrase grounding), our goal is solely on accurate large-scale prediction of attributes. In this work, we compare against CLIP, a large scale image-text matching and zero-shot image classifier trained on 400 million images and alt-text from the internet.

Our model architecture is related to the end-to-end object detection Transformer DETR and its language modulated MDETR [5, 28]. In DETR, a Transformer encoder is used to contextualize input image features before localizing objects. In the localization step, a Transformer decoder takes in N *object queries*, and decodes them into bounding box and category by cross-attending to the image features from the encoder. Our model also takes after this *object query* approach with a Transformer decoder, but instead of decoding into object category and bounding box, we decode them into attributes.

Open-vocabulary methods have been studied for object recognition and detection using natural language [15, 68, 65, 14, 50, 13, 22]. Even though attributes can be part of the text query, these works often neglect attributes in their proposal and evaluation, and only focus on object nouns. Earlier works have used object hierarchy from WordNet [68], which is unsuitable for attributes since adjectives/verbs do not have such hierarchy predefined, or search engine to retrieve web text description of object [15] which is costly. Recent works have attempted

pretrained text embeddings (e.g., BERT) [64, 22] or vision-text embeddings (*e.g.*, CLIP) [14, 13] to detect novel objects that are semantically similar to the text in the embedding space. However, their focus is still on objects (nouns) and not their descriptions. [14] attempts to include attributes with their object detector, but it only consists of basic colors. To our knowledge, our work is the first to utilize pretrained text embeddings for large-scale attribute prediction on an unconstrained set of object categories.

3 Attribute Data Preparation

Attribute Extraction: Our goal is to build an image understanding system that can recognize object-centric attributes as well as its immediate interaction with nearby objects. We refer to these as attribute phrases (or just 'attributes', used interchangeably) as they can be in the form of multiple words (*e.g.*, *wearing hat*). We select the following prominent image-text datasets as our data sources: Visual Genome (VG) [31], GQA [21], COCO-Attributes [45], Flickr30K-Entities [47], MS-COCO [7], and a portion of Localized Narratives (LNar) [48].

VG, GQA, and COCO-Attrs contain object-level attribute labels and bounding boxes, which we directly use. VG-DenseCap, Flickr30K-Entities, MS-COCO, and LNar, on the other hand, contain attributes in their captions. Hence, we rely on language dependency parser [19, 53, 60] and derive rules to detect attributes, including adjectives, verbs, verb-object and preposition-object pairs. Some of these datasets contain grounding information (bounding box, mouse trace) for each caption that we also extract. We convert LNar mouse trace into bounding box (refer to supplementary). Several examples of these attributes are illustrated in Fig. 1. For the remaining captions, we extract objects without any grounding.
Large-Scale Object Attribute Dataset (LSA): We aggregate all images, their parsed objects and attributes into our dataset that we call Large-Scale Attribute (LSA). The overall statistics is in Table 1. From 420k images, we split into 379k images for training, 8k for validation, and 33k for testing. For training, we construct a vocabulary set of attributes that we deem to be common, determined by frequency thresholding for adjective and verb attributes (*e.g.*, adjectives appear $\geq$ 75 times), and keeping only those that involve common object categories for interaction and location attributes. More details about this construction and the attribute statistics can be found in the supplementary. Ultimately, this results in a training (or seen) attribute set $\mathcal{C}_\text{s}$ with $|\mathcal{C}_\text{s}| = 5526$.

4 TAP - Transformer for Attribute Prediction

There are two common approaches in attribute prediction in the wild, differing in how multiple objects in an image are processed. First approach, often used with object detector (*e.g.* Faster-RCNN), extracts features for the whole image and pools regions that contain the objects [1, 23]. The pooled features are used to predict attributes for each object. These models are normally not used as standalone attribute prediction models but rather as pre-training target

Table 1. Statistics of attributes extracted for LSA. Note that LNar contains 32k and 122k images from Flickr30K and COCO respectively. Among all instances, 7.1M are grounded (bounding box), 1.4M weakly grounded (mouse trace), and 975k ungrounded.

Datasets	# images	# instances	# attr annotations	Type of grounding
VG + GQA	108k	6.5M	10.1M	Box
Flickr30K-Entities	32k	285k	503k	Box
MS-COCO + COCO-Attrs	122k	1.2M	2.2M	Ungrounded + Box
Localized Narratives	312k	1.4M	1.7M	Mouse trace
Total	420k	9.5M	14.6M	

for downstream vision-language tasks. Second approach uses crops of each object and processes them separately and independently. While the former encodes more context and is more computationally efficient as it can predict attributes for multiple objects in one forward pass, it suffers from lower accuracy since the feature resolution for each object is lower. Here, we introduce TAP, a new tranformer-based model for attribute prediction that achieves many desirable properties: 1) Use of context information, 2) attribute prediction of multiple objects in a single pass, and 3) easily extendable to unseen attributes.

Problem setting: Let I be an input image consisting of N objects with named categories $\{o_i\}_{i=1}^{N}$ and potentially bounding boxes $\{b_i\}_{i=1}^{N}$. If an object does not have bounding box, its top-left and bottom-right can be set to the image corners. Let $\mathcal{C}_\text{s}$ be the set of training attribute classes, then each object has a ground-truth label vector $Y_i = [y_{i,1}, ..., y_{i,\mathcal{C}_\text{s}}]$, $y_{i,c} \in \{1, 0\}$ denoting whether attribute c is positive or negative. In our work, we treat the unlabeled classes as negatives. Our goal is to train a multi-label classifier to predict these $\mathcal{C}_\text{s}$ attributes on all N objects. Additionally, we also train a final layer to ensure proximity of the phrase embedding for an attribute, which makes it capable of predicting open-world attributes. We call this extension to our TAP model as **OpenTAP**.

4.1 Model Architecture

Fig. 2 illustrates the architecture of TAP. TAP takes in two input modalities: (1) a *visual sequence*, and (2) a *query sequence*.

Visual sequence: The first part of our model is a CNN backbone (ResNet-50 in our model) that takes in an input image of size $w \times h$ and returns a grid of features with size $L_v = w/32 \times h/32$. We flatten this feature grid to get a list of feature vectors $\{x_i\}_{i=1}^{L_v} \in \mathbb{R}^{2048}$, which we refer as visual tokens. These visual tokens are then added with a learned 2-D positional encodings similar to [28] so that spatial information can be preserved. Let r_i and c_i be the row and column index of x_i in the feature grid, its final representation v_i is obtained as

$$row_i = \text{RowEmbed}(r_i), \quad col_i = \text{ColumnEmbed}(c_i), \tag{1}$$

$$v_i = x_i + \text{concat}([row_i, col_i]) \tag{2}$$

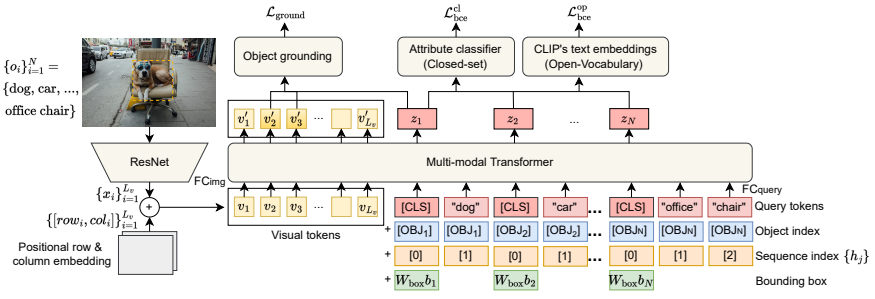

Fig. 2. Model architecture. Sequence of ResNet encodings form the input visual tokens. This is processed jointly with the query token which consists of object query tokens (red), their object index embedding (blue), a sequence index embedding (orange), and a bounding box embedding (green). Contextualized representation z_i of the `[CLS]` token of all objects are decoded into attributes. In addition, an object grounding loss is used to train object localization (shown here for *dog*).

Query sequence: Based on the list of N objects, a sequence of *object queries* is created so that our model can decode them into attributes. For every object i, we add to our query sequence a `[CLS]` token, which is the object query that shall be decoded into attributes at the final stage. For every `[CLS]`, we use the following information: (1) the object category name, (2) the image location of the object, (3) the object instance index that the query corresponds to. Specifically, for every object i, we tokenize its category o_i using the WordPiece tokenizer [61] into word tokens and append them into the sequence. To provide image location: we encode the object's bounding box [39, 20] as a 5-d vector $b_i = \left(\frac{x_0}{W}, \frac{y_0}{H}, \frac{x_1}{W}, \frac{y_1}{H}, \frac{(x_1-x_0)(y_1-y_0)}{H\times W}\right)$ where (x_0, y_0) and (x_1, y_1) are respectively the top-left and bottom-right coordinates of the box, and H and W specify the image size; b_i is then projected via a learned linear layer W_{box} and added with the `[CLS]` token. Next, to indicate which object instance a given `[CLS]` token belongs to, we add to it a cardinal *object index* embedding $\texttt{[OBJ}_i\texttt{]}$. Finally, a cardinal *sequence index* is added for every object category tokens to denote the order of tokens within each object category and account for objects comprising multiple words (*e.g.*, *office chair*). This *sequence index* resets for every new object instance. In summary, given object i, its token w_j (at index j w.r.t. object i) has its final representation h_j computed as follows:

$$\hat{w}_j = \text{WordEmbed}(w_j), w_{\texttt{[OBJ}_i\texttt{]}} = \text{WordEmbed}(\texttt{[OBJ}_i\texttt{]}), \tag{3}$$

$$p_j = \text{SequenceIndexEmbed}(j), \tag{4}$$

$$h_j = \hat{w}_j + w_{\texttt{[OBJ}_i\texttt{]}} + p_j + W_{\text{box}} b_i \mathbb{1}_{w_j=\texttt{[CLS]}}, \tag{5}$$

where W_{box} is the learnable linear layer that transform b_i to have the same dimension as the word embeddings.

Multi-modal Transformer: Both visual ($\{v_i\}$) and query ($\{h_j\}$) embeddings are mapped to the same embedding space with the help of two fully-connected

(FC) layers, FC_{img} and FC_{query}. Both sequences are then concatenated into a single, long sequence and fed to the Transformer. Doing so allows an object's attribute prediction to properly account of its context and surrounding objects, which is crucial for predicting attributes that denote an object property in context of others (*e.g.*, *wearing glasses*). We denote the output visual embeddings to be $\{v_i'\}$ which will be used for the object grounding loss discussed later. For the output query embeddings, we only care about those belong to the `[CLS]` tokens, which we denote as z_i for the output `[CLS]` embedding of object i.

Closed-vocabulary classifier (TAP): We apply a linear classifier on every z_i to obtain $\mathcal{C}_\text{s}$ logit values for the attribute classes $[r_{i,1}, ..., r_{i,\mathcal{C}_\text{s}}]$.

Open-vocabulary classifier (OpenTAP): We propose an open-vocabulary classifier head that extends TAP to recognize novel attributes it has not seen during training. As CLIP is a SOTA zero-shot image classifier that has been trained on 400M image-text pairs and potentially seen an enormous amount of attributes, we propose to use pretrained CLIP text embeddings for our open-vocabulary classifier. We train a linear layer on top of z_i's to project them close to the CLIP text embeddings of the ground-truth attributes, while keeping the text embeddings fixed. By fixing the text embeddings, we expect our model to generalize to unseen attributes represented by arbitrary text inputs thanks to the structure in the CLIP embedding space.

Formally, let q_j be the CLIP text embedding of attribute class j. To compute similarity between z_i and q_j, we use the scaled cosine similarity

$$s_{i,j} = \frac{z_i^T q_j / \tau}{\|z_i\| \|q_j\|}, \tag{6}$$

where τ is a temperature hyperparameter. Details on how we generate text embeddings of the attributes are presented in section 5. Note that our open-vocabulary classifier head is not limited to CLIP text, but can be used with any pretrained text encoders (*e.g.*, BERT [10]). We select CLIP mainly because it is more representative of the visual world and as determined by empirical results. For example, BERT that is only trained on text corpus is not expected to capture well object appearance characteristics, such as color and texture.

4.2 Training and loss functions

Attribute classification: We apply a reweighted binary cross entropy loss for our closed-set prediction branch

$$\mathcal{L}_{\text{bce}}^{\text{cl}}(Y, r) = \sum_{i=1}^{N} \sum_{c=1}^{\mathcal{C}_s} - \mathbb{1}_{[y_{i,c}=1]} p_c \log(\sigma(r_{i,c})) - \mathbb{1}_{[y_{i,c}=0]} n_c \log(1 - \sigma(r_{i,c})), \tag{7}$$

where p_c and n_c are the positive and negative weights for attribute c computed in the same way as [46] to handle data imbalance. Similarly, the open-vocabulary branch is also trained with the same BCE loss and we denote it as $\mathcal{L}_{\text{bce}}^{\text{op}}(Y, s)$.

Object grounding As our training data also contains ungrounded image-text

pairs, we employ a grounding loss that trains the model to attend to the correct image regions for objects with known grounding. By providing grounding supervision when available, the model can learn to ground object and transfer that knowledge to softly localize any object of interest when training/testing on ungrounded objects. Specifically, for a query embedding z_i of object i, we enforce an alignment between z_i and the output visual embeddings $\{v'_j\}_{j \in O_i^+}$ where O_i^+ denotes the indices of those visual tokens that locate inside the bounding box of object i in the image feature grid. Our grounding loss is as follows:

$$\mathcal{L}_{\text{ground}} = \sum_{i=1}^{N} \frac{1}{|O_i^+|} \sum_{j \in O_i^+} -\log \left(\frac{\exp\left(z_i^T v'_j / \tau\right)}{\sum_{k=0}^{L_v - 1} \exp\left(z_i^T v'_k / \tau\right)} \right), \tag{8}$$

which is similar to contrastive loss in [28], but instead of using it to strongly supervise a phrase grounding model, we use it to equip TAP with object grounding ability so that it can also learn and predict attributes from ungrounded objects. Our final loss is the sum of the BCE and the object grounding loss

$$\mathcal{L} = \mathcal{L}_{\text{bce}}^{\text{cl}} + \lambda_{\text{op}} \mathcal{L}_{\text{bce}}^{\text{op}} + \lambda_{\text{ground}} \mathcal{L}_{\text{ground}}. \tag{9}$$

5 Experiments

In this section, we describe our main experiments: (1) closed-set attribute prediction on VAW [46], (2) open-vocabulary attribute prediction on LSA, and (3) human-object interaction classification on HICO [6]. Results on VAW and HICO demonstrate our model's understanding of adjective, verb, and interaction classes, while results on LSA shows its ability to predict large number of unique attributes, and even recognize unseen attributes in the open world.

Architecture: We use the ImageNet-pretrained ResNet-50 [18] for the image backbone. For word embeddings, we use the pretrained BERT-base [10, 58]. Our multi-modal Transformer takes in both visual and query features at once and has 5 self-attention layers with 8 attention heads each. Further implementation details, including hyperparameters, are presented in the supplementary.

OpenTAP: As mentioned in Sec. 4.1, we extend TAP to recognize unseen attributes by using CLIP-RN50 to generate text embeddings for the attribute classes. Given an attribute, we extract its embedding using an ensemble of multiple prompts [50], such as '*A photo of something that is* `<attr>`.'. Since object information is already present in the input query (Fig. 2), using object agnostic prompts allows us to pre-compute all attribute embeddings which significantly improves training speed. The supplementary contains all prompts that we use.

5.1 Closed-set attribute prediction

Dataset: We evaluate TAP in a closed setting on VAW [46], a large-scale attribute in the wild dataset that contains positive and negative labels for 620 attributes across multiple types (*e.g.*, *color, material, shape, size*). With explicit

Table 2. Results on VAW. The top box reports results of methods trained only on VAW, while the bottom box shows our newly introduced baseline RN50-Context and TAP on VAW after pre-trained on LSA. LSA-pretrained and VAW-supervised denote whether a model is trained with attribute labels from LSA and VAW respectively

Methods	LSA pretrained	VAW supervised	mAP	mR@15	mA	F1@15
RN50-Baseline [46]		✓	63.0	52.1	68.6	63.9
ML-GCN [9, 46]		✓	63.0	52.8	69.5	64.1
Sarafianos et al. [52, 46]		✓	64.6	51.1	68.3	64.6
SCoNE [46]		✓	68.3	58.3	71.5	70.3
TAP [**Ours**]		✓	65.4	54.2	67.2	66.4
RN50-Context	✓	✓	67.3	54.1	69.3	66.1
TAP [**Ours**]	✓		67.2	53.8	65.5	61.5
TAP [**Ours**]	✓	✓	**73.4**	**63.3**	**73.5**	**71.1**

negative labels, the VAW dataset allows for reporting better evaluation metrics on this problem, albeit on a much smaller scale than what TAP is capable of.
Setup: We report 3 versions of TAP: one that is trained only on VAW, one that is trained only on LSA, and one that is first pretrained on LSA and then finetuned on VAW. The 620-attribute set of VAW is also a subset of $\mathcal{C}_s$. When pretraining on LSA, we make sure to exclude VAW test images from LSA.
Baselines: We compare with ResNet-50 baseline, ML-GCN [9], Sarafianos et al. [52], and the SOTA model SCoNE [46]. These models predict attributes on each cropped object independently, and use ground-truth segmentation mask (provided in VAW) to improve accuracy. Because these models require accurate object box as input for cropping, they cannot be trained on LSA - a dataset with noisy or even no bounding boxes in many cases (refer to Table 1).

Because it can be argued that TAP achieves better results by simply using more context, we introduce another baseline, RN50-Context, which is the RN50-Baseline but takes in the whole image and uses RoIAlign to extract object feature for classification. Because RN50-Context does not perform cropping, we can pretrain it on LSA. More details about this can be found in the supplementary.
Results: Following [46], we report mean Average Precision (mAP), mean recall at top-15 (mR@15), mean balanced accuracy (mA), and overall F1 at top-15 (F1@15). The result is presented in Table 2.

– **Without LSA-pretrained:** After trained only on VAW, TAP achieves better results than RN50-Baseline, ML-GCN [9], and Sarafianos et al. [52]. ML-GCN is not effective since it requires constructing label co-occurrence matrix which is not suited for partially labeled problem such as VAW. Sarafianos et al. has to learn to produce one attention map per attribute, which is costly and redundant because many attributes (*e.g.*, *color*) already share the same attention map that cover the entire object region. However, TAP without LSA-pretrained is lower than SCoNE [46] (-2.9% mAP). Note that SCoNE uses segmentation mask while TAP does not. Transformer is well-known for being

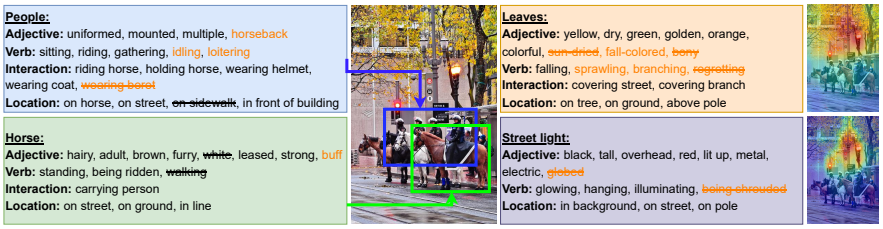

Fig. 3. Qualitative results. Every attribute list is sorted in descending order of the model's confidence. Both **seen attributes** from closed and unseen attributes from open-vocabulary branch are shown. We display the attention mask of TAP for objects without bounding box. Strikethrough represents wrong predictions as judged by us.

data hungry, and VAW consists of 50× less instances than LSA. Hence training only on VAW does not fully utilize TAP's capability that we specifically design it for: large-scale attribute learning from image-text datasets.

– **With LSA-pretrained:** TAP without finetuning achieves +4.2% mAP than RN50-Baseline, even though it is only trained on sparse attributes parsed from captions and is not trained on VAW densely annotated data. After finetuning, TAP achieves a new SOTA with a substantial improvement of +5.1% mAP and +5.0% mR@15 over SCoNE. RN50-Context, our redesigned RN50-Baseline that uses context, is almost comparable with SCoNE, showing the effectiveness of using context and the LSA data. However, even though TAP and RN50-Context both use context, TAP is clearly better. The impressive performance of TAP is attributed to the effective usage of context, multi-modal Transformer, and our training algorithm that allows to learn attribute from image-caption datasets. In the supplementary, we provide qualitative results and detailed performance breakdown on each attribute type (*e.g.*, TAP achieves much better accuracy on *action* attributes than the baselines).

More discussion: To demonstrate TAP's efficiency that can predict attributes for multiple objects in a scene in a single pass, we report the inference time on VAW: on average, it takes 18.01ms/img for TAP, while it is 43.71ms for SCoNE and 40.05ms for RN50-Baseline. In addition, thanks to object grounding loss, TAP can also work when bounding box is not given. We demonstrate this qualitatively in Fig. 3, and quantitatively by removing all boxes from VAW and re-evaluate TAP, where we obtain 68.9% mAP which is still better than SCoNE.

5.2 Open-vocabulary attribute prediction

In this section, we evaluate OpenTAP on seen and unseen attributes in LSA. We focus on investigating how a model trained on large number of attributes can generalize to unseen attributes by leveraging fixed text embeddings.

Setup: OpenTAP generalizability to unseen attributes can be studied in 2 ways:

1. **LSA common:** First, we study whether OpenTAP can extrapolate to recognize unseen but common attributes, *e.g.*, can it recognize never-seen *black*

from having seen *white* and *gray*? We perform frequency-based sampling to select 605 attributes from set $\mathcal{C}_s$ of 5526 attributes (refer to sec. 3), and remove them from the training data so that they can be used as unseen attributes. Hence, we have 4921 seen, and 605 unseen attributes. The test set consists of randomly sampled 100k instances from the test images that are labeled with any of these 5526 attributes. By using frequency-based sampling instead of uniform, we ensure the unseen set also contains attributes that are more common (*e.g.*, common colors like *black, orange*).

2. **LSA common→rare:** Next, we study whether once trained on common attributes, can OpenTAP generalize to long-tailed unseen attributes. For this, we keep the whole set $\mathcal{C}_s$ of 5526 attributes intact as our seen set. From all attributes in the test images that do not belong in $\mathcal{C}_s$, we construct an unseen set $\mathcal{C}_u$ by selecting those that appear more than 8 times (to filter out noise, typos). We also subsample some types of attributes (*e.g.* location attributes) so that various attribute types are well-balanced in $\mathcal{C}_u$. This results in $|\mathcal{C}_u| = 4012$ classes. We sample 60k instances in LSA test that are labeled with either attributes in $\mathcal{C}_s$ and $\mathcal{C}_u$ for this setup. Since $\mathcal{C}_s$ already contains 5526 most common attributes, the remaining unseen attributes in $\mathcal{C}_u$ are not only unseen, but are also semantically distant from the attributes seen during training because they belong to the long-tail.

For both setups, because we use CLIP for comparison, we make sure all instances in our test set are larger than 25% of the image area in order to not put CLIP at a disadvantage due to small object size.

Baselines: We use CLIP as our baseline. As discussed in Sec. 2, CLIP is a SOTA zero-shot image classifier and has been used successfully for open-vocabulary object detection [14, 13]. However, no existing work have studied CLIP for open-vocabulary attribute recognition. We introduce 3 CLIP baselines based on how the attribute classifiers are constructed from its text encoder:

1. **CLIP (attribute prompt):** Similar to OpenTAP, for every attribute, we create its classifier by ensembling multiple prompts with formats similar to the following 'A photo of something that is `<attr>`'. This model is agnostic to the object present in the image since the object is not mentioned in the prompts. This is done to establish parity with OpenTAP setup.
2. **CLIP (object-attribute prompt):** We ensemble object-aware prompts with formats similar to the following 'A photo of `<obj> <attr>`' (*e.g.*, *A photo of man riding horse*). We observe that this solely object-aware prompt returns drastically low accuracy due to CLIP being unable to detect nonsensical object-attribute pairs, *e.g.*, for an image of a boat with the text *A photo of a boat wearing shirt*, CLIP still returns a high similarity score since CLIP is highly attentive to the object mentioned in the prompt and it is not trained to detect incompatible object-attribute pairs.
3. **CLIP (combined prompt):** To alleviate the above problem, we find that combining the object-aware with the object-agnostic prompts allows CLIP to focus more on the attribute aspect.

Table 3. Evaluation of **LSA common** and **LSA common→rare**

Methods	LSA common			LSA common→rare		
	AP_{seen}	AP_{unseen}	$AP_{overall}$	AP_{seen}	AP_{unseen}	$AP_{overall}$
CLIP (attribute prompt)	2.53	3.37	2.64	2.62	2.52	2.58
CLIP (object-attribute prompt)	0.97	1.56	1.04	1.16	0.73	0.97
CLIP (combined prompt)	2.81	3.67	2.92	3.12	2.63	2.91
OpenTAP	14.34	7.62	13.59	15.39	5.37	10.91

All CLIP baselines use ensemble of multiple prompts within each prompt type [50] (refer to supplementary). Furthermore, given an object with its bounding box, we ensemble its CLIP image embeddings from its 1×, 1.25×, and 1.5× crops to incorporate context similar to [14] (this improves +0.4 mAP). These are our best-faith effort to augment CLIP model to allow for maximum accuracy.

The baselines from the closed experiment cannot be used in this setup because they do not scale to the large number of classes in LSA. For example, Sarafianos et al. [52] produces one attention map per class, ML-GCN [9] builds a graph of all classes as nodes, SCoNE [46] runs supervised contrastive loss iteratively over every class, all of which is expensive when the number of classes is large.

Results: We report in Table 3 the mAP to evaluate the ranklist returned for every attribute by each method. The attributes are categorized into *seen* and *unseen* based on the data OpenTAP is trained on. Unlike OpenTAP, CLIP cannot be entirely zero-shot in these experiments as it presumably has already been trained on these attributes from its 400M image-caption training data [50]. Hence, CLIP shows little difference in performance of seen versus unseen in both experiments. The results show that CLIP ensemble of object-agnostic and object-aware prompt is better than just object-agnostic or just object-aware.

OpenTAP achieves better results than CLIP on both seen/unseen set in both experiments by a clear margin. Despite CLIP having been trained on enormous image-text data and shown to be successful for object recognition [14, 13], the results here suggest that CLIP is lacking in terms of attribute understanding. Note that OpenTAP and CLIP still use the same text embeddings as classifiers, and the higher accuracy of OpenTAP is attributed to its architecture and training algorithm to allow OpenTAP to detect better visual cues for attributes.

The results in **LSA common→rare** experiment show that OpenTAP can recognize both seen (common) and unseen (rare) attributes better than CLIP. However, the gap between seen and unseen in this case is not small (10% gap), which suggests there's still room for improvement. When evaluating TAP in **LSA common**, the gap between seen and unseen is less as expected. This result also shows that TAP can extrapolate and recognize unseen but common ones.

Qualitative results: We show in Fig. 3 example of attribute prediction results. We can see that OpenTAP can predict even attributes that are rare (*loitering, buff, sprawling*). In addition, we present in Fig. 4 top image retrieval results for some unseen attributes. The unseen attributes *excited* and *fishing* from our

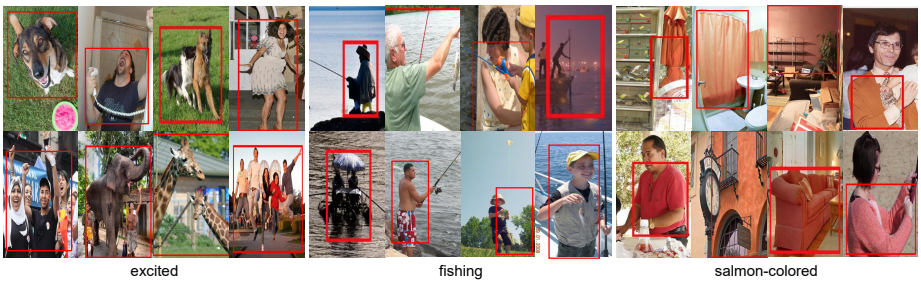

Fig. 4. Qualitative results. Top image retrieval results for several unseen classes.

Table 4. Ablation on class embeddings

Methods	AP_s	AP_u	AP
OpenTAP-ViCo	11.40	3.75	7.98
OpenTAP-BERT	14.00	4.81	9.90
OpenTAP-Phrase BERT	14.66	4.80	10.26
OpenTAP-CLIP	15.39	5.37	10.91

Table 5. Ablation on training portion

Methods	AP_s	AP_u	AP
VG	9.59	3.88	7.04
VG+Flickr	11.00	4.64	8.16
VG+Flickr+COCO	13.27	4.98	9.56
VG+Flickr+COCO+LNar	15.39	5.37	10.91

LSA common experiment provide good results, presumably by extrapolating from near seen classes (*e.g.*, *yelling* and *laughing* for *excited*; *holding rod* and *near water* for *fishing*). Similarly, for **LSA common→rare**, *salmon-colored* is rare and unseen but the model is also able to extrapolate based on all common color classes that it has seen during training, such as *orange, pink*.

5.3 Ablation studies

We conduct ablation study on the **LSA common→rare** split to investigate how our choice of attribute embeddings and our constructed dataset LSA is helpful.

Class embeddings: We investigate other text embeddings to be used with OpenTAP: (1) ViCo [17], word embedding learned from object-attributes co-occurrences in Visual Genome, (2) BERT embeddings [10] that has been used for open-vocabulary object detection in [65, 22], and (3) PhraseBERT [57]. The results are presented in Table 4, which show that OpenTAP is not dependent solely on CLIP since even BERT embeddings help OpenTAP outperform CLIP baselines on unseen classes. ViCo, even though is trained on object-attributes in VG, results in low mAP. CLIP text embeddings result in the highest mAP.

Training data portion: Because LSA is an aggregation of multiple datasets with different levels of grounding, we ablate each one to see their contribution to the final performance. We present the results in Table 5, where we can see that all datasets contribute positively, even ungrounded (COCO) and weakly grounded (LNar) one. This shows that with additional ungrounded image-caption (*e.g.*, SBU [43], Conceptual Captions [54]), OpenTAP could achieve even better.

5.4 Closed-set human-object interaction classification

We further show the generalizability of OpenTAP on the human-object interaction dataset HICO [6] which contains image-level interaction labels (*e.g. boarding plane, riding boat*) that are similar to what OpenTAP has learned from LSA.
Dataset: HICO [6] contains 600 human-object interaction (HOI) labels of 117 verbs and 80 object categories. Every image in the dataset contains one or more HOI classes that need to be predicted, making this a multi-label prediction problem. HICO training set contains 38,116 images, while the test set comprises 9,658 images. Following prior work, we use 10% of the training images for validation.
Baselines and Setup: We compare with PastaNet [35] and HAKE [34] which are SOTA models on HICO that additionally use object detection and human keypoints. We also compare with DEFR [24], a SOTA model that uses ResNet as image backbone and CLIP text embedding as initialization for the classifiers which are later finetuned. For our OpenTAP model, we also finetune the CLIP-initialized classifiers. One difference between DEFR and OpenTAP is the image backbone. While DEFR uses backbone pretrained on CLIP 400M image-text pairs, OpenTAP uses ImageNet- and LSA-pretrained backbone. For fair comparison, we compare with DEFR-RN50 that uses CLIP ResNet-50 as backbone. More implementation details in this experiment are presented in the supplementary.
Results: We report results in Table 6, showing that OpenTAP outperforms PastaNet and HAKE without having to use object detector and human keypoints. OpenTAP also surpasses DEFR-RN50 by a clear margin. These are evidence that our proposed architecture and training algorithm for OpenTAP are effective for learning attributes of objects that can even generalize to HOI classes.

Table 6. Results on HICO image classification

Methods	Bbox	Pose	CLIP text	mAP
PastaNet	✓	✓		46.3
HAKE	✓	✓		47.1
DEFR-RN50			✓	49.7
OpenTAP			✓	**51.7**

6 Conclusions

In this paper, we propose a Transformer-based model for attribute prediction that can predict a large number of unique attributes, and can be extended to learn open-vocabulary attribute by leveraging image-text datasets and pretrained text embeddings. We expand the definition of attributes to include things that a given object interacts with, which we argue to be a part of the object property as well. Our proposed pretrained TAP model not only achieves a new SOTA on a strongly supervised setting after finetuning, but also shows good performance without any finetuning. Our TAP model can be extended to OpenTAP, which is capable of predicting novel attributes unseen during training, with greater accuracy than CLIP.
Acknowledgements: This work was partially supported by DARPA SAIL-ON (W911NF2020009) and SemaFor (HR001119S0085) programs, and partially supported by gifts from Adobe.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6077–6086 (2018) 4
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual question answering. In: The IEEE International Conference on Computer Vision (ICCV) (2015) 1
3. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: European Conference on Computer Vision. pp. 663–676. Springer (2010) 3
4. Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., Plank, B.: Automatic description generation from images: A survey of models, datasets, and evaluation measures. Journal of Artificial Intelligence Research **55**, 409–442 (2016) 1
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020) 3
6. Chao, Y.W., Wang, Z., He, Y., Wang, J., Deng, J.: Hico: A benchmark for recognizing human-object interactions in images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1017–1025 (2015) 3, 8, 14
7. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015) 4
8. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Learning universal image-text representations (2019) 3
9. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5177–5186 (2019) 9, 12
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 7, 8, 13
11. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1785. IEEE (2009) 3
12. Ferrari, V., Zisserman, A.: Learning visual attributes. In: Advances in neural information processing systems. pp. 433–440 (2008) 3
13. Gao, M., Xing, C., Niebles, J.C., Li, J., Xu, R., Liu, W., Xiong, C.: Towards open vocabulary object detection without human-provided bounding boxes. arXiv preprint arXiv:2111.09452 (2021) 3, 4, 11, 12
14. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021) 3, 4, 11, 12
15. Guadarrama, S., Rodner, E., Saenko, K., Zhang, N., Farrell, R., Donahue, J., Darrell, T.: Open-vocabulary object retrieval. In: Robotics: science and systems. vol. 2, p. 6 (2014) 3
16. Guo, S., Huang, W., Zhang, X., Srikhanta, P., Cui, Y., Li, Y., Adam, H., Scott, M.R., Belongie, S.: The imaterialist fashion attribute dataset. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019) 1, 3

17. Gupta, T., Schwing, A., Hoiem, D.: Vico: Word embeddings from visual co-occurrences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7425–7434 (2019) 13
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 8
19. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). https://doi.org/10.5281/zenodo.1212303 4
20. Huang, H., Liang, Y., Duan, N., Gong, M., Shou, L., Jiang, D., Zhou, M.: Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. arXiv preprint arXiv:1909.00964 (2019) 6
21. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019) 4
22. Huynh, D., Kuen, J., Lin, Z., Gu, J., Elhamifar, E.: Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. arXiv preprint arXiv:2111.12698 (2021) 3, 4, 13
23. Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., Chen, X.: In defense of grid features for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10267–10276 (2020) 4
24. Jin, Y., Chen, Y., Wang, L., Wang, J., Yu, P., Liang, L., Hwang, J.N., Liu, Z.: Decoupling object detection from human-object interaction recognition. arXiv preprint arXiv:2112.06392 (2021) 14
25. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3668–3678 (2015) 1
26. Kafle, K., Kanan, C.: Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding (2017) 1
27. Kafle, K., Shrestha, R., Kanan, C.: Challenges and prospects in vision and language research. Frontiers in Artificial Intelligence **2**, 28 (2019) 1
28. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1780–1790 (2021) 3, 5, 8
29. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014) 1
30. Kosti, R., Alvarez, J.M., Recasens, A., Lapedriza, A.: Emotion recognition in context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1667–1675 (2017) 1
31. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**(1), 32–73 (2017) 1, 3, 4
32. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: European Conference on Computer Vision. pp. 121–137. Springer (2020) 3

33. Li, Y., Huang, C., Loy, C.C., Tang, X.: Human attribute recognition by deep hierarchical contexts. In: European Conference on Computer Vision. pp. 684–700. Springer (2016) 1
34. Li, Y.L., Xu, L., Liu, X., Huang, X., Xu, Y., Chen, M., Ma, Z., Wang, S., Fang, H.S., Lu, C.: Hake: Human activity knowledge engine. arXiv preprint arXiv:1904.06539 (2019) 14
35. Li, Y.L., Xu, L., Liu, X., Huang, X., Xu, Y., Wang, S., Fang, H.S., Ma, Z., Chen, M., Lu, C.: Pastanet: Toward human activity knowledge engine. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 382–391 (2020) 14
36. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1096–1104 (2016) 1
37. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (2015) 1, 3
38. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: European conference on computer vision. pp. 852–869. Springer (2016) 2
39. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265 (2019) 3, 6
40. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: Composition with context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1792–1801 (2017) 3
41. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 953–962 (2021) 3
42. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 169–185 (2018) 3
43. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2text: Describing images using 1 million captioned photographs. In: Neural Information Processing Systems (NIPS) (2011) 13
44. Parikh, D., Grauman, K.: Relative attributes. In: 2011 International Conference on Computer Vision. pp. 503–510. IEEE (2011) 3
45. Patterson, G., Hays, J.: Coco attributes: Attributes for people, animals, and objects. In: European Conference on Computer Vision. pp. 85–100. Springer (2016) 3, 4
46. Pham, K., Kafle, K., Lin, Z., Ding, Z., Cohen, S., Tran, Q., Shrivastava, A.: Learning to predict visual attributes in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13018–13028 (2021) 1, 2, 3, 7, 8, 9, 12
47. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015) 4
48. Pont-Tuset, J., Uijlings, J., Changpinyo, S., Soricut, R., Ferrari, V.: Connecting vision and language with localized narratives. In: European Conference on Computer Vision. pp. 647–664. Springer (2020) 4

49. Purushwalkam, S., Nickel, M., Gupta, A., Ranzato, M.: Task-driven modular networks for zero-shot compositional learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3593–3602 (2019) 3
50. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021) 2, 3, 8, 12
51. Saini, N., Pham, K., Shrivastava, A.: Disentangling visual embeddings for attributes and objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13658–13667 (2022) 3
52. Sarafianos, N., Xu, X., Kakadiaris, I.A.: Deep imbalanced attribute classification using visual attention aggregation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 680–697 (2018) 9, 12
53. Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., Manning, C.D.: Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In: Workshop on Vision and Language (VL15). Association for Computational Linguistics, Lisbon, Portugal (September 2015) 4
54. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1238, https://aclanthology.org/P18-1238 13
55. Siddiquie, B., Feris, R.S., Davis, L.S.: Image ranking and retrieval based on multi-attribute queries. In: CVPR 2011. pp. 801–808. IEEE (2011) 3
56. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019) 3
57. Wang, S., Thompson, L., Iyyer, M.: Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. arXiv preprint arXiv:2109.06304 (2021) 13
58. Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45 (2020) 8
59. Wu, C., Lin, Z., Cohen, S., Bui, T., Maji, S.: Phrasecut: Language-based image segmentation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10216–10225 (2020) 1
60. Wu, H., Mao, J., Zhang, Y., Jiang, Y., Li, L., Sun, W., Ma, W.Y.: Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6609–6618 (2019) 4
61. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016) 6
62. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2251–2265 (2018) 1
63. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5410–5419 (2017) 2

64. Zareian, A., Karaman, S., Chang, S.F.: Bridging knowledge graphs to generate scene graphs. In: European Conference on Computer Vision. pp. 606–623. Springer (2020) 4
65. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14393–14402 (2021) 3, 13
66. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5532–5540 (2017) 2
67. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., Gao, J.: Vinvl: Making visual representations matter in vision-language models. arXiv preprint arXiv:2101.00529 (2021) 3
68. Zhao, H., Puig, X., Zhou, B., Fidler, S., Torralba, A.: Open vocabulary scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2002–2010 (2017) 3
69. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017) 1