Open-vocabulary Attribute Detection
Supplementary

María A. Bravo Sudhanshu Mittal Simon Ging Thomas Brox
{bravoma,mittal,gings,brox}@cs.uni-freiburg.de
University of Freiburg, Germany
https://ovad-benchmark.github.io

Contents

A. OVAD Benchmark

A.1. Attribute taxonomy

Figure 1 shows the attribute taxonomy. We grouped attributes by type to simplify and optimize the annotation process. This diagram corresponds to the attributes of all objects. 19 attribute types are displayed in the inner circle of the radial tree. Each attribute contains its synonyms separated by '/'. For the human category, *color* type refers to *clothes color* and *hair color*; pattern type refers to *clothes pattern*, and length refers to *hair length*. Additionally, we included *sitting* as an attribute to the position type and *bald* as an attribute to the hair length type. In total, we obtain 117 distinct attributes for the OVAD benchmark.

A.2. Attribute distribution

Figure 2 shows the long-tailed attribute distribution of positive annotations in the OVAD dataset. Following previous works [5, 9], we split attributes in three subsets 'head', 'medium', and 'tail' according to the number of positive instances annotated in the OVAD dataset. To split the classes into 'head', 'medium', and 'tail', we defined two thresholds

$$t_{high} = median(\mathbf{f}) + std(\mathbf{f}), \text{and;}$$
$$t_{low} = median(\mathbf{f}) - std(\mathbf{f})/10.$$

where **f** is the frequency vector of the number of positive annotations. 'head' corresponds to 15 attribute classes whose frequency is above t_{high}, 'tail' are the ones below t_{low} composed of 49 attribute classes, and 'medium' corresponds to 53 attribute classes, the ones whose frequency is between t_{high} and t_{low}.

A.3. Dataset size analysis

To show that the size of the OVAD dataset is sufficient for a reliable evaluation of the OVAD task, we analyze the standard deviation of the performance of the OVAD-Baseline-Box. Figure 3 shows the standard deviation of the mAP for the frequency-defined subsets: 'all', 'head', 'medium', and 'tail'. We randomly selected differently-sized subsets of images from the OVAD dataset for this analysis. The size of the subsets range from 3% to 33% of the total number of images in the OVAD dataset. We evaluated the OVAD-Baseline model, where ground-truth bounding boxes are provided during evaluation. The maximum size of a subset is set to 33% to obtain at least three non-overlapping sets of images, which is required for a reliable calculation of the standard deviation. We conducted

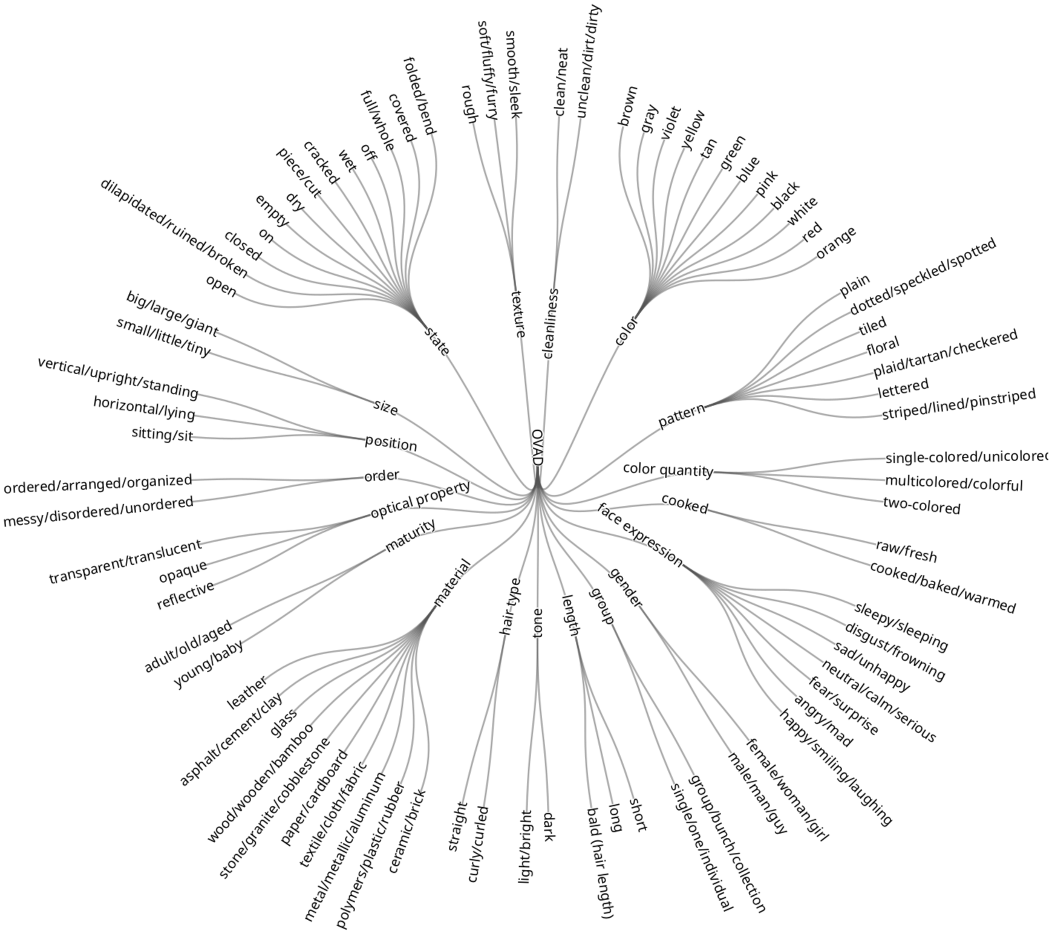

Figure 1. The figure shows the taxonomy of attribute categories as a radial tree. The 117 attribute categories are divided into 19 attribute types, shown in the first circle. Certain attribute types are repeated for the human category, where the color includes hair color and clothes color. Similarly, pattern refers to clothes pattern, length refers to hair length, and tone refers to hair tone.

this experiment six times using different data shuffles to select the splits. In every run we selected a maximum of six non-overlapping splits (for every data size percentage) and computed the standard deviation of the mAP per size of the subset. Then, we average the standard deviation across the six experiments and report the results in Figure 3. We observed that the standard deviation decreases as the size of the subset increases. At 23%, the standard deviation is lower than 1% for all attribute partitions, and the standard deviation of the 'tail' attribute classes is similar to 'head' and 'medium' attribute partitions. When the 'tail' attribute curve is extrapolated to 100% of the dataset size (2000 images), the standard deviation is estimated to be less than 0.3%.

B. Dataset Creation

B.1. Annotation process

As mentioned in the main paper, the OVAD dataset is fully annotated by humans following strict guidelines to achieve consistent and dense annotations (guidelines are attached at the end of the supplementary). The annotation process started from scratch for attributes to avoid any preexisting errors from previous datasets. We utilized the identified 19 attribute types and the taxonomy to facilitate the annotation process. We randomly selected 2000 images from the 2017 validation set of MS COCO dataset and used the object annotations as a starting point.

The annotation system offers a drop-down list of attributes for feasible attribute types for each object instance. For every object, the annotators marked one of the attributes within every attribute type as positive or unknown. For every attribute type, our system allows only one possible at-

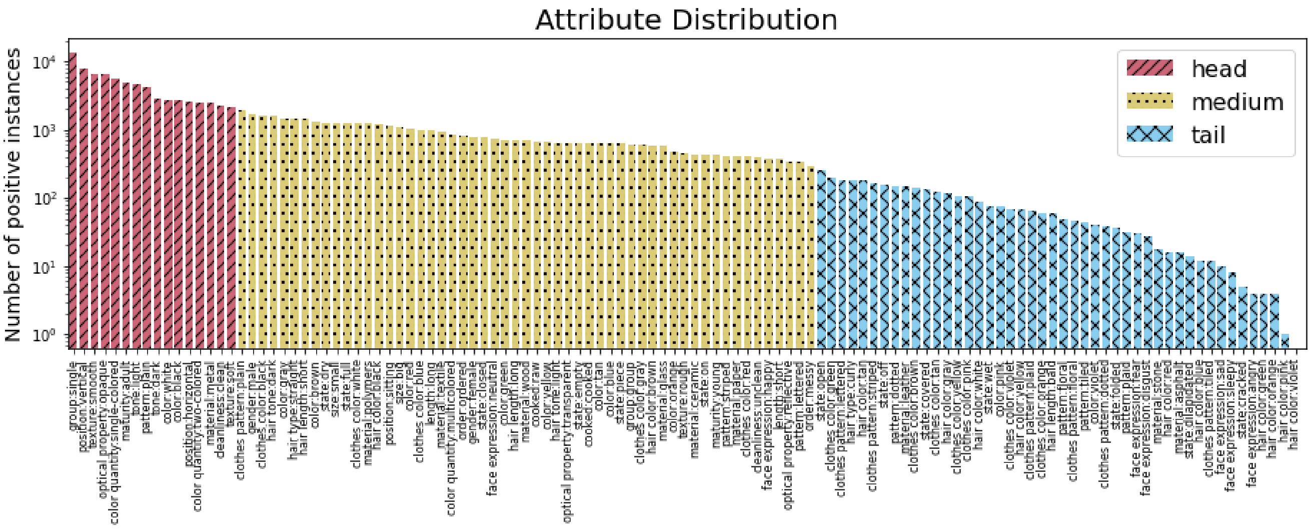

Figure 2. The figure shows the attribute frequency distribution in the OVAD benchmark. Bar colors correspond to the frequency-defined subsets *head*, *medium* and *tail*.

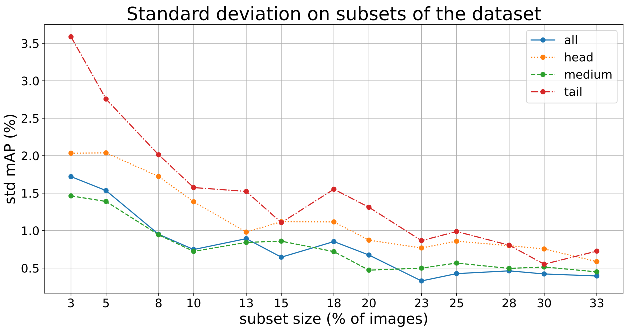

Figure 3. Standard deviation of the mAP performance for the oracle OVAD baseline. We show the scores on differently-sized non-overlapping subsets of images, from 3% to 33% of the OVA dataset. All splits ('all', 'head', 'medium', and 'tail') show a decreasing behavior as the number of images increases. At 33%, the standard deviation is lower than 1% for all attribute splits.

tribute selection. We consider all attributes under the same attribute type mutually exclusive except for two types - *color* and *state*. We use this exclusiveness property to automatically annotate negative attributes by considering all the non-selected attributes (from that attribute type) as negative or unknown. For the attribute type *color*, which is not exclusive, the annotation system offered the possibility to select more than one option as positive. The non-selected colors were either considered negative or ignored depending on the *number of colors* attribute. The type *state* considers a wide range of attributes that are not all mutually exclusive, but some are antonym pairs (*e.g.* wet/dry, open/close). In this case, only the antonyms of the positive-selected attribute were marked as negative, and the rest as unknown.

It is worth noting that the taxonomy and exclusiveness property was exploited for the benchmark annotation only. It is neither available for training the models nor for predicting the attribute scores, which is done in an open-vocabulary fashion. Providing the attribute classes or the taxonomy to the model during training is against the purpose of the proposed benchmark.

B.2. Annotation quality control

We followed a progressive annotation approach. Each annotator received an initial set of images along with the annotation guidelines. Then, a second annotator revised the same set and, based on the annotation guidelines, corrected and completed the missing annotations. Once the annotations were revised, the first annotator received feedback. We repeated this process until a reasonable quality of annotations was achieved (approx. five sets of 50 images each). The progressive process resulted in high annotation quality, with a revision of approximately 80% of the annotations. The remaining 20% of the annotation had only one annotation round, corresponding to the last sets annotated by the trained annotators.

To test the annotation quality of the above-mentioned revised and remaining set of images, we selected 10% of the data from each of the two sets to perform a second independent annotation from scratch by the experienced annotators and measured the consistency of annotations. As a result, we obtained an overall consistency of 89.44% for the revised images and 86.35% for the remaining non-revised set. Additionally, we considered a golden set of 50 images which all the annotators had to do at the end of the annotation process. For this set we obtained an overall consistency of 91.26%±2.79. This consistency metric includes positive, negative, and unknown annotations.

B.3. Human bias in attribute annotation

Attribute-level annotations are prone to human biases, which can cause ambiguous type errors, especially in the case of unclear images. We make extensive quality checks to minimize such errors. At least 80% of the images were revised by a second annotator to establish consistency and correctness of the annotations. On average, annotators spent 12 minutes per image annotating all attributes that apply. While revising, annotators spent approximately 3 minutes per image. The average hourly wage was 12 Euros per hour. Our dataset was annotated by fourteen annotators from 6 nationalities, different age groups, sex, and skill levels. This ensures our annotations are balanced for cultural, age, and sex biases.

B.4. Exceptions in attribute annotation

Our annotation process offers restricted options to annotate based on the object class category. It does not allow infeasible annotations for each attribute category. *E.g.*, there is no attribute annotation for the 'material' of a person or 'cooked' state for a skateboard since it is irrelevant in most of the cases. However, there can be exceptions such as a photo of a person on a banner or a cake in the form of a skateboard. Some of such exceptions are also missed due to stereotyping. *E.g.*, if there is a car, the annotator might label its material as metal even when it is visually indiscernible. One of the limitations of this work is that our annotation process does not consider such exceptions, thus adding some noise to our annotations.

Figure 4 shows some examples of exceptions and corner cases in which some attributes are missed due to our annotation system. For every image, the highlighted attributes correspond to the exception cases. For the first and second row, there is a limitation of not considering specific attributes for different objects such as 'material' for 'apple', 'person', or 'cake', and 'clothes color' for 'teddy bear'. For other cases, our annotation system includes the corner cases and selects the correct attribute. The third and fourth row of Figure 4 shows the possibility of selecting 'material' for animals or a different material for some vehicles.

B.5. Dataset annotations and visualization

Our dataset annotations can be found in our web page: https://ovad-benchmark.github.io. They are in a json file. The format of the annotations is compatible with the MS COCO annotations. Attribute annotations for every instance correspond to a list under the key "att_vec" with values of 1, 0, and -1 corresponding to positive, negative and unknown labels respectively. The attribute list is also included in the json file. Additionally, we offer a visualizer in our project web page. It includes a search system by object and attribute. We distinguish between base and novel object classes and positive, negative and unknown attribute classes using color codes.

C. Supervised Ablation

C.1. OVAD supervised training ablation

To check the feasibility of the OVAD task using our dataset, we perform a supervised 4-fold cross-validation experiment to get an upper bound performance. For each run, we consider the 500 image set as the test set and fine-tune a ResNet50 [6] architecture pre-trained on Imagenet [4] using the remaining 1500 images. We train the multi-label attribute classification model in the box-oracle setup. The model achieves an average performance of 48.16±0.52 mAP compared to the chance performance of 8.29±0.06 mAP. For reference, our OVAD-Baseline-Box achieves 23.30±0.76 mAP on the same splits.

C.2. Cross-dataset transfer ablation

Our primary interest lies in the OVAD task, which considers all attributes as novel categories. However, in order to investigate the potential transfer of knowledge from previous benchmarks to our OVAD benchmark, we conducted an ablation experiment. We trained two ResNet50 networks using the cropped objects from the COCO Attributes [8] and VAW [9] datasets, respectively. We trained a multi-label attribute classification model using the box-oracle setup, incorporating a projection layer at the end of the network to obtain vector representations of the same dimension as the CLIP text encoder. We computed the similarity between every attribute encoded by the CLIP text encoder and the object visual vector. We train the models using binary cross entropy loss with the positive and negative attribute labels from the datasets. Our models achieved a performance of 15.96 mAP and 18.20 mAP on OVAD (box-oracle setting) after training on the COCO Attributes and VAW datasets respectively.

D. OVAD-Baseline

D.1. Implementation details

Our OVAD-Baseline method uses ResNet50 [6] as backbone, pre-trained on ImageNet [4], for the detector model F and the CLIP text encoder [10] as the language model G. Similar to CLIP, we use the cosine similarity (Equation (1) in the main paper) between the visual representation f_b of the object's bounding box and the text class embedding g_c for applying the classification losses during training and for calculating the prediction scores during inference.

We set τ to 50 during training and testing for both object and attribute prediction. The temperature parameter is

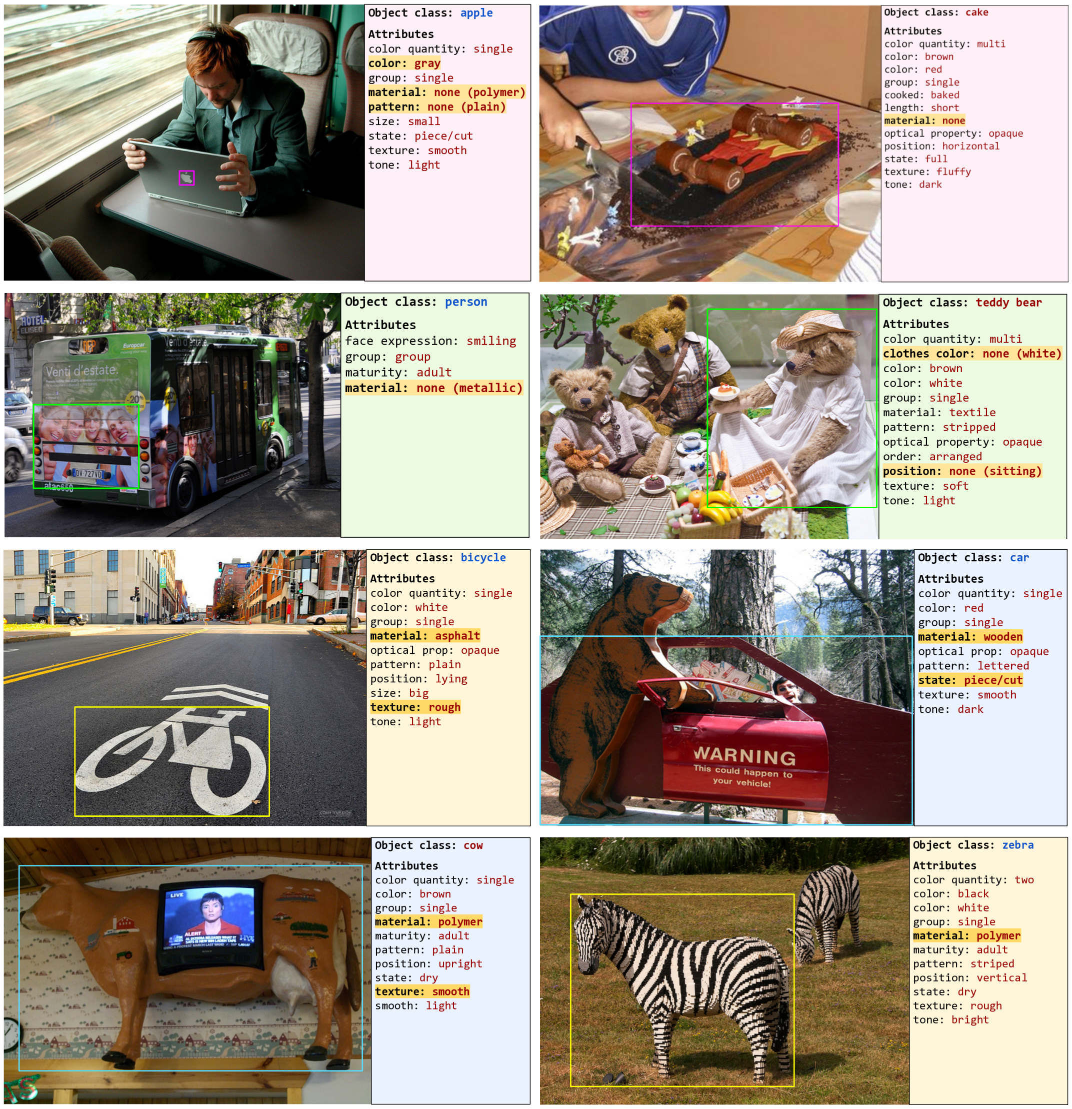

Figure 4. Exceptions in attribute annotation. Each example shows some of the corner cases present in the OVAD benchmark. The exception attributes for each instance are highlighted in yellow. The correct version is included in the parenthesis if the annotation is marked as unknown in our benchmark.

selected empirically for object detection. For reference, Detic [15] uses a value of 50, and CLIP [10] uses a value of 14.29 for the temperature hyperparameter. As mentioned in Section 4 in the main paper, we use a binary cross entropy objective for all the losses that use the classification head, which are image-caption matching and parts-of-caption matching (noun/noun phrase/noun complement) with max-area box. For efficiency, we compute the text representations offline and load the features of positive and negative image-text pairs during training. We select one positive and 63 negative captions for image-caption matching and compute the similarity with the bounding box covering the whole image. For parts-of-caption matching, we select all positive samples p and $50 - p$ negative parts-of-caption to compute binary cross-entropy with the maximum area bounding box proposal. We use COCO Captions [3] 2017 training set as the image-caption dataset for training.

During training, we use a base learning rate of 0.02 with a step reduction of 10x at 60 k and 80 k iterations. We train the model for a total of 90 k iterations with 1 k warmup steps using the SGD optimizer, similar to previous open-vocabulary detection methods [13, 15]. The training is done per batch of one type of data at a time; a batch contains either box+class labels of base classes or caption+parts-of-caption labels. We use the same sampling ratio of training as Detic [15] of 1:4 for the batch type of images, using four times more batches with the captions. We use a batch size of 64 for image-caption data and 16 for box+class data.

During inference, we considered several synonyms for every attribute class representing the category. We refer to this set as $S_a = \{w_i : w_i \text{ is a synonymous term for attribute } a\}$. These sets are shown on Figure 1 and are listed under every attribute type using '/'. When calculating the text-attribute embedding g_a for every category, we average the representations of the individual synonyms using

$$g_a = \frac{1}{|S_a|} \sum_{w_i \in S_a} g_{w_i}. \tag{1}$$

We use the similarity score (Equation (1) in the main paper) as the prediction score for every attribute.

E. Experimental Extension

E.1. Open-vocabulary attribute detection results

Table 1 shows the results of evaluating the different state-of-the-art methods on both open-vocabulary object detection benchmarks based on the MS-COCO [7] validation dataset. We use two different sets of image annotations, our extended OVD-80 benchmark with updated object annotations and the OVD benchmark proposed by Bansal *et al.* [1]. Results are shown in the *Generalized* scenario where detection is performed across both base and novel classes together.

Our method achieves a high AP_{50} performance for novel categories with a trade-off with base class performance. Methods marked with * use novel classes to obtain pseudo image-labels, therefore are not open-vocabulary by definition. The order of the methods is consistent across both image sets, OVD and OVD-80. OVD-80 has 32 novel objects class making the object detection task more challenging compared to having 17 novel object classes in OVD.

E.2. Performance per attributes type

Figure 5 shows the performance of the five methods for every type of attribute category. Categories such as material, optical property, order, size, and texture show a bigger improvement over chance performance than other attribute types.

Figure 6 shows the mAP scores for all six foundation models per attribute type in the box-oracle setup. X-VLM outperforms all other methods by a large margin for the majority of attribute types. Some attributes such as cooked, gender and maturity have a higher relative improvement over chance level.

F. Qualitative Results

Figure 7 shows qualitative examples of the baseline method for OVAD. For every example, the first image corresponds to the prediction and the second to the ground truth annotations. All base and novel entities are shown in blue and red respectively. The prediction which has the maximum overlap with the ground truth bounding box is considered as the final prediction. We rank the attribute prediction scores for each attribute category and select the top 200 scores for visualization. Figure 7 shows some images with high mAP performance.

G. Licences of Assets

We provide some additional details about the datasets, codes and other used assets. These details include the source and their licenses.

(CVAT) Computer Vision Annotation Tool The application and the code for the CVAT tool [12] are available at the GitHub repository: https://github.com/openvinotoolkit/cvat, web page: https://cvat.org. The repository is licensed under the MIT license.

MS COCO Both MS COCO detection and caption datasets [3, 7] are available at their web page https://cocodataset.org and github repository https://github.com/cocodataset/cocoapi. These dataset follow the following licences: Attribution-NonCommercial-ShareAlike License, Attribution-NonCommercial License, Attribution-NonCommercial-NoDerivs License, Attribution License, Attribution-ShareAlike License, Attribution-NoDerivs License, No known copyright restrictions, United States Government Work.

OVAD dataset attribute annotations license The OVA benchmark, annotations along with the website are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

H. Annotation Guidelines

These guidelines were provided to the annotators to maintain consistency and agreement in the annotations.

1. Given an image, check for every object marked and verify that it has the correct class.

2. Add bounding boxes for the missing objects, revise inaccurate bounding boxes.

3. Annotate attributes as positive only based on their visual appearance.

4. Assign 'unknown' for cases where: (a) the attribute is not visible, like in the presence of occlusion or (b) a discrete label can not be assigned because of ambiguity or an in-between case.

Method	Generalized (OVD-80) - 2,000 images			Generalized (OVD) - 4,836 images		
	Novel (32)	Base (48)	All (80)	Novel (17)	Base (48)	All (65)
OV-Faster-RCNN	0.4	53.1	32.0	0.3	53.0	39.2
OVR [13]	17.9	51.8	38.2	22.8	46.0	39.9
VL-PLM [14]*	19.7	**58.9**	43.2	34.4	**60.2**	**53.5**
Detic [15]*	20.0	49.2	37.5	27.8	47.1	45.0
LocOv [2]	22.5	52.5	40.5	28.6	51.3	45.7
OVAD-Baseline	24.7±0.6	49.1±0.2	39.3±0.4	30.0±0.5	48.3±0.4	43.5±0.3
Rasheed *et al.* [11]*	**32.5**	56.6	**46.9**	**36.6**	54.0	49.4

Table 1. AP_{50} on Open-Vocabulary Object Detection. *: novel class labels were used during training to filter captions and obtain the image tags (Detic), or to obtain pseudo-labels (VL-PLM).

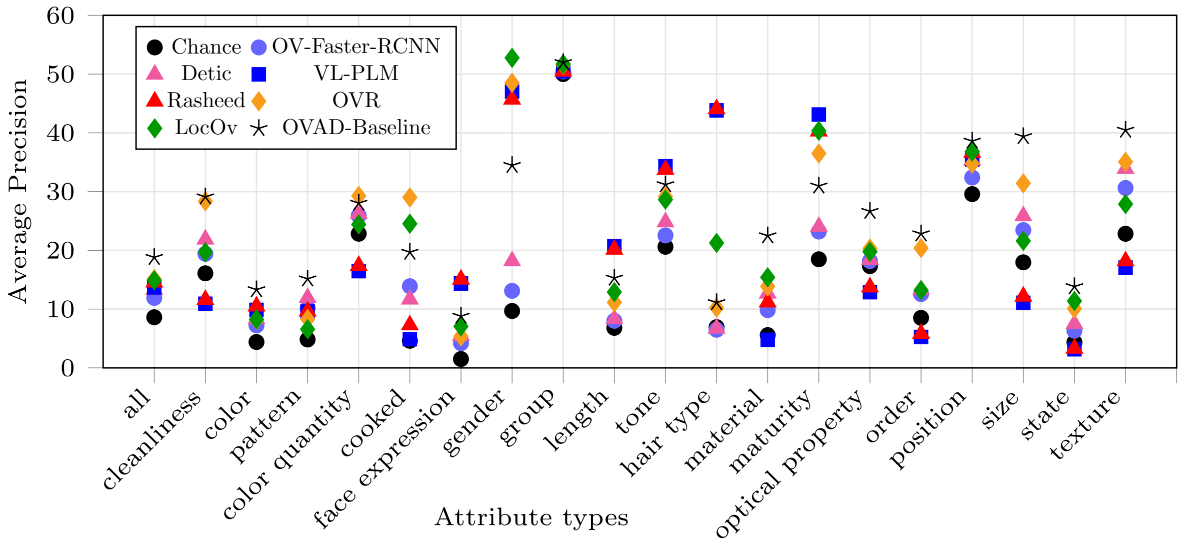

Figure 5. Comparison between different baseline methods on open-vocabulary attribute detection on the OVAD benchmark.

5. Check through all the feasible attribute types and select the most appropriate attribute category as positive according to the attribute descriptions (included below).

We considered four types of object categories. The valid set of attribute types is allocated based on this object category.

- human: person
- animal: bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
- food: banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
- object: bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass, cup, fork, knife, spoon, bowl, chair, couch, potted plant, bed, dining table, toilet, tv, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush

Based on these categories, we defined the possible attributes to assign from the 19 attribute types according to attribute descriptions.

Attribute categories

1. **cleanliness**

    - **clean/neat -** This attribute is marked when an object is clearly clean. This attribute usually ap-

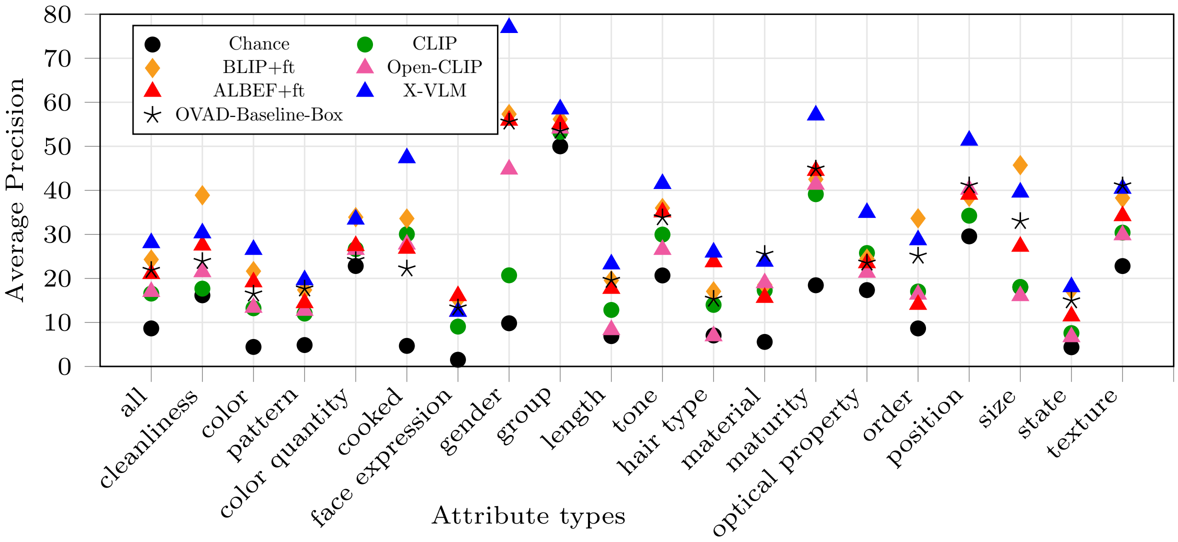

Figure 6. Comparison between different the different foundation models on the box-oracle OVAD benchmark.

plies to objects that appear to be new or especially clean for the picture and for animals that are fully visible and no dirt can be seen.

- **unclean/dirt/dirty/muddy -** This attribute is marked when an object is clearly dirty. This attribute usually applies to objects with something over them, some visible spillage or dust. It usually applies to graffiti on walls not designed for that, animals with mud, dirty dishes, or objects on the street that are poorly maintained.

2. **color, clothes color, hair color: (black, white, gray, tan, brown, green, red, yellow, blue, orange, violet, pink) -** This attribute refers to the visible color of the object. Color/clothes color/hair color applies to different object types differently. For example, hair color and clothes color apply only to humans; however, humans have no color attribute.

3. **color quantity**

   - **single-color, two-colored -** This attribute is marked when an object comprises exactly one or two colors.

   - **multi-color -** This attribute is marked when more than two colors are present in the object. Even when some text or lines are in a third color, it is marked as multi-colored.

4. **cooked -** This attribute type is marked only for food object categories. It denotes whether the food is cooked/baked or raw.

5. **face expression -** This attribute type refers to the person's facial expression. This attribute is only marked when the face of the person is clearly visible.

6. **gender -** The attribute type refers to the gender of the person. This attribute is marked based on the combination of body features, face features, clothing, context, etc. We understand that sometimes it can be challenging to mark the gender of a person just based on appearance. Therefore, we take extreme measures, particularly for this attribute, and only mark it when it is very evident.

7. **group -** This attribute type refers to the number of instances in the bounding box. There are two possible categories for this type of attribute - single/individual or group/collection.

8. **hair type -** This attribute type refers to the hair type of the person and is classified either as curly/curled or straight.

9. **length: long, short; hair length: long, short, bald** This attribute type is marked when an object is evidently extra-long/short relative to its standard/average size. For example, an international airplane like Airbus-380 is marked as long, whereas a private jet is marked as short.

10. **material -** This attribute type refers to the most visible material in appearance. If two dominant materials exist, then the structure's material is marked, and if the object is covered with another material, then the surface's material is marked.

11. **maturity -** This attribute type refers to the physical maturity of humans or animals. This attribute is either marked as adult/old or young/baby.

12. **optical property -** This attribute type expresses the optical property of the object's material. Most objects are marked as opaque. If the surface of the opaque object is reflective, then the optical property is marked as reflective. Other remaining attribute includes transparent/translucent objects.

13. **order**

    - **unordered -** This attribute is marked when an object is cluttered and fails to follow any particular order. This attribute usually applies to objects which carry or comprise multiple elements or parts. For example, a working desk with cluttered items or a couch with objects lying on it in an unorganized way.
    - **ordered -** This attribute is marked when the object is organized and holds an order. This attribute usually applies to objects which carry or comprise multiple elements or parts.

14. **pattern -** This attribute type refers to the pattern of the surface of the object. It includes clothes patterns, object surface patterns, etc. If the surface is a mixture of two patterns, then the pattern is marked as unknown.

15. **position -** This attribute type refers to the orientation of the object. This attribute also includes the sitting attribute. There is no consensus on the object's orientation for certain classes like table, bowl, and microwave. Therefore, they are marked as unknown.

16. **size: big, small -** This attribute type is marked when an object is evidently extra-big/small relative to its standard/average size. For example, an elephant could be considered big by default, however it is only marked as big only if it is extra large relative to a normal-sized elephant.

17. **state -** This attribute type is a non-exclusive attribute that contains multiple attribute sub-types like dry/wet, closed/open, turned on/off, etc. Different sub-types apply to different object categories. Electronic devices can be marked as either turned on or off. Animals can be marked as dry or wet. Container-type objects can be marked as either open or closed.

18. **texture** This attribute type refers to the visual appearance of the consistency of the surface of the objects.

    - **smooth/sleek -** This attribute is assigned for objects having a flat, regular surface or appearance.
    - **soft/fluffy/furry -** This attribute corresponds to objects with surfaces covered with fur or hair, as well as objects that could be easily pressed and deformed.
    - **rough -** This attribute describes objects with irregular or uneven textures whose appearance shows irregularities on the surface.

19. **tone -** This attribute type refers to the tone of the surface of the object. The tone is either marked as light/bright or dark. It could refer to the color tone of the object or hair tone, depending on the object class. For the person object class, only hair tone is marked.

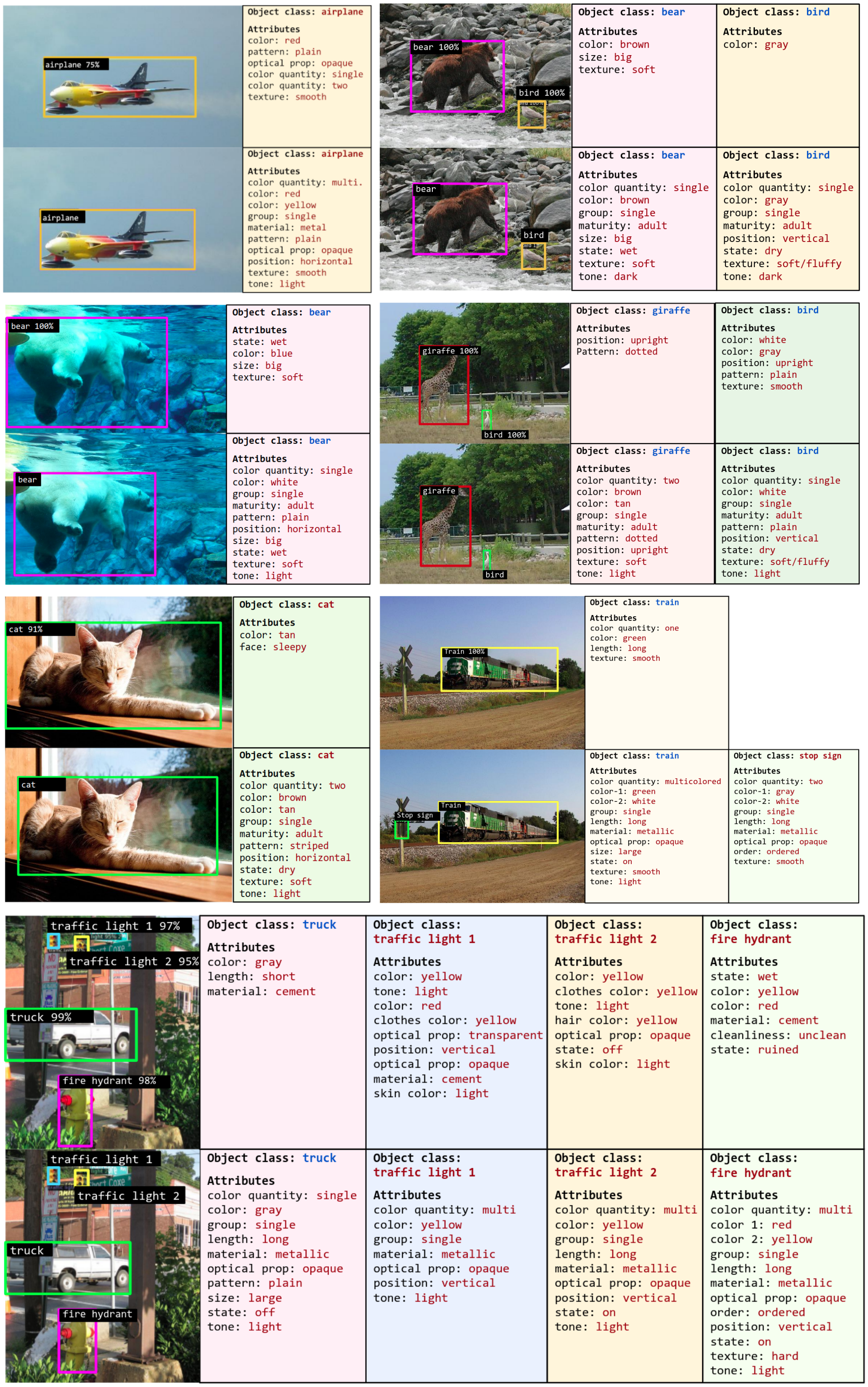

Figure 7. Qualitative examples. For each example, top row shows the predictions of the proposed OVAD base model and bottom row shows the ground-truth.

References

[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 6

[2] Maria A. Bravo, Sudhanshu Mittal, and Thomas Brox. Localized vision-language matching for open-vocabulary object detection. In *GCPR*, 2022. 7

[3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5, 6

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[8] Genevieve Patterson and James Hays. Coco attributes: Attributes for people, animals, and objects. In *ECCV*, 2016. 4

[9] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, 2021. 1, 4

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4, 5

[11] Hanoona Abdul Rasheed, Muhammad Maaz, Muhammd Uzair Khattak, Salman Khan, and Fahad Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NeurIPS*, 2022. 7

[12] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya4607, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming, and Tritin Truong. opencv/cvat: v1.1.0, 2020. 6

[13] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 5, 7

[14] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, 2022. 7

[15] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 5, 7