

Beyond Seen Primitive Concepts and Attribute-Object Compositional Learning

Supplementary Material

7. Dataset Split

7.1. Dataset Creation

Proposing a new benchmark is always challenging and requires meticulous efforts. Since the problem OV-CZSL, requires constructing multiple splits, we create multiple random splits and then pick the best split for each dataset by evaluating seen and unseen accuracy using two standard baselines. Overall, the steps involved in creation of benchmark splits for MIT-states [20] dataset is as follows:

- Step 1: As explained in Paper Section 1.1, the proposed OV-CZSL task is combination of generalized ZSL and CZSL. As a visual feature backbone, we use ResNet18 [18], pre-trained on ImageNet [4]. To ensure that our setup is truly Zero-shot Learning (ZSL) for attributes and objects, we must include all common attributes and objects seen in ImageNet [4] in the seen attributes and object sets. Hence, while creating seen attributes and objects sets for MIT-States, we first find out the objects and attributes in MIT-states [20] which are already present in ImageNet [4]. We exempt those common attributes and objects, shown in Table 7, from being part of unseen attributes and objects A^* and O^* . In other words, these attributes and objects are always part of seen set A and O , for every random split.
- Step 2: We randomly select 20-25% attributes and objects as unseen attributes A^* and unseen objects O^* .
- Step 3: The set of all the compositions which have seen attributes and seen objects is denoted as AO . Among this set, 80% pairs make up the training set Y^s . The rest 20% pairs are part of the test and validation set, to evaluate the model performance on seen attributes, seen objects yet unseen compositions $(AO)^*$.
- Step 4: Iterating through rest of the valid compositional pairs present in MIT-States [20], we split the pairs in 3 other sets: AO^* has pairs which have seen attribute and unseen object, A^*O has pairs with unseen attributes and seen objects, and A^*O^* . Hence, test set Y^u composes of $\{(AO)^*, AO^*, A^*O, A^*O^*\}$.
- Step 5: In order to evaluate on pairs which are already present in training set, we select 25% and 30% pairs from training set AO and add 18% samples of these pairs in the validation and test set. This becomes the set of seen pair samples in the test and validation set, which is used for stopping criteria. Similar split of seen pair exists in CZSL splits as well.
- Step 6: Lastly, we split Y^u into 45%/55% ratio as validation and test sets, such that each set of seen-unseen compositions are part of validation and test sets. The numbers

for each split are shown in the paper, Table 1.

This process creates one split. We randomly create 10 such splits for each benchmark. To ensure the stability of the benchmark split, it requires a balance in performance between the seen pairs AO and unseen pairs $\{(AO)^*, AO^*, A^*O, A^*O^*\}$. Hence, for each split, we run two common CZSL baselines, LabelEmbed (LE) [33] and CompCos [29]. Based on the AUC and difference between best seen and unseen accuracy, we choose one split as the benchmark split for OV-CZSL of the dataset. This process is performed for all datasets (C-GQA [29] and VAW-CZSL [37, 43]), but we only show the 10 random splits and their baselines performance for MIT-States [20] in Table 8.

8. Additional Training Details and Ablation

8.1. Implementation details

Following standard practice in CZSL [28, 29, 33, 39], we use Frozen ResNet18 [18], pre-trained on ImageNet [4] for image features (without finetuning) and BERT [5] text embeddings for labels. A linear layer on top of BERT [5] features is used for pair embeddings, instead of using Object-Conditioned module from OADis [43]. We use image augmentations (random crop, horizontal flip) for all baselines and our method, following OADis [43]. Most details for MIT-States [20] are in the paper, however there is a slight error. Smoothing factor is $\alpha = 0.8$ for MIT-states [20] and the temperature for cosine similarity $\delta = 0.05$. For C-GQA [29] and VAW [37, 43] as well, the smoothing factor for label propagation $T = 0.5$, number of neighbors $k = 5$, weights for losses are $\beta_1 = 0.8, \beta_2 = \beta_3 = 0.95$, and $\gamma_1 = \gamma_2 = 0.05$. OADis [43] also has losses \mathcal{L}_{seen} and \mathcal{L}_{unseen} , which were originally weighted with 0.1. We keep this the same for MIT-States [20], but change the weights for C-GQA [29] to 0.2. Smoothing factor α for C-GQA [29] and VAW-CZSL [37, 43] is 0.5 and 0.8 respectively. Learning rate for C-GQA [29] is $1e-4$, with weight decay $5e-5$ and Adam optimizer, decay milestone [20-130]. For VAW-CZSL [37, 43], it has same configuration as C-GQA [29] with learning decay milestone as [30,40]. We acknowledge that there are multiple hyperparameters in our setup. However, we mainly tune the hyperparameters for smoothing factors (α and T) and number of neighbors. Other hyperparameters are almost same as OADis [43].

8.2. Additional Ablation

Following the paper, all ablation studies are conducted on MIT-States [20] dataset. The final Test AUC and HM are the considered measures for searching best hyper-

Table 7. **Dataset Splits.** We propose new benchmark splits for OV-CZSL on datasets MIT-states [20], C-GQA [29] and VAW-CZSL [37, 43].

Datasets	Common Objects	Common Attributes
MIT-states [20]	balloon, banana, brass, bubble, bucket, candle, castle, church, cliff, desk, drum, envelope, fig, laptop, lemon, library, necklace, orange, pizza, plate, pot, screw, tiger, vacuum, valley, velvet, wool	upright

Table 8. **Using NEL with other baselines.** We show effect of NEL for different baselines. All methods using NEL perform better for OV-CZSL splits, which include one unseen component.

Models	Split 1	Split 2	Split 3	Split 4	Split 5	Split 6	Split 7	Split 8	Split 9	Split 10
LE [33]	0.81 / 6.57	1.01 / 7.64	0.75 / 6.63	0.78 / 6.69	0.67 / 6.35	0.82 / 7.10	0.87 / 6.88	0.92 / 7.36	0.71 / 6.34	1.0 / 7.60
CompCos [29]	1.66 / 10.70	1.97 / 10.22	1.53 / 9.65	1.79 / 9.88	1.47 / 9.86	1.97 / 10.53	1.85 / 10.65	1.63 / 10.02	2.05 / 10.79	1.39 / 10.58
	1.23 / 8.63	1.49 / 8.93	1.27 / 8.25	1.28 / 8.28	1.07 / 8.10	1.39 / 8.81	1.36 / 8.76	1.27 / 8.69	1.38 / 8.56	1.19 / 9.09

parameters.

Textual Features. Textual features are the source of regularization and knowledge transfer, which makes these a crucial design decision. We observe that for zero-shot learning task, GloVe [36] performs better than BERT [5] embeddings. Since BERT is contextual embeddings, attribute-object embeddings from BERT are more helpful than GloVe. We also compare with other common word embeddings like Fasttext [2] and word2vec [30] in Table 11. Most word embeddings perform better for unseen compositions of seen attributes-objects $(AO)^*$ and A^*O . We use BERT for all baselines of OV-CZSL since it outperforms for the most challenging set A^*O^* . **Neighbors.** We use GloVe [36] embeddings for neighbor selection for two reasons: (1) since these are word embeddings, they capture attribute features and it’s neighbors more efficiently. (2) GloVe embeddings have strong semantic structure and arithmetic qualities (e.g., $\text{emb}(\text{king}) + \text{emb}(\text{women}) - \text{emb}(\text{man}) = \text{emb}(\text{queen})$). Since, BERT [5] are contextual embeddings, it does not necessarily follow this structure. GloVe [36] embeddings for single word are more robust and have algebraic expression in embedding space, which is why GloVe seemed optimal for neighbor search. For example, when we compute neighbors for object ‘elephant’, GloVe [36] features gives neighbors like: [rhinoceros, rhino, animal, whale] as top neighbors, which are all large animals and make sense. However, using BERT [5] embeddings produces top neighbors: [monkey, lion, camel, lizard, mermaid], which has low similarity with the ‘elephant’ in general, apart from the fact all are animals. Moreover, we quantitatively show that using GloVe [36] embeddings for searching for neighbors gives better performance in comparison to using BERT [5] in Table 9.

Another idea is using neighbors from the dataset itself.

Table 9. Ablation for different smoothing factors: We set different smoothing factors α for \mathcal{L}_{AO}^{NE} with dataset MIT-states [20], to empirically find the best value for α .

Emb	Neighs	Test@1	HM	$(AO)^*$	AO^*	A^*O	A^*O^*
BERT	External.	1.71	9.56	12.17	3.24	4.87	2.07
GloVe	MIT-states	2.13	10.16	17.94	4.25	7.67	2.40
GloVe	External.	2.41	10.94	18.87	5.49	8.24	3.54

For this, instead of using external sources of attributes and objects, we only use the unseen attributes and objects to find neighbors from. For smaller dataset like MIT-states [20], these neighbors do not make much sense but on a higher level correlate seen attributes with unseen attributes explicitly. However, the generalizability of model gets affected, as the unseen attribute-unseen object pairs accuracy is better if external sources of attributes and objects are used (as shown in Table 9). The column 1 shows the embeddings used for neighborhood search and column 2 refers to is the neighbors are extracted from external sources or within the dataset.

Impact of Smoothing Factors In our work, we use two smoothing factors: α and T . α is the smoothing term for label smoothing, whereas T is the smoothing term for neighbors in label propagation. We explain the significance of both separately. To have a balance between generalization and learnability, we empirically find the smoothing factor for MIT-States [20] dataset, and use the same for other datasets.

First, for label smoothing α . Let us assume there are total K labels. For each sample, one of these K labels is correct. Here in Table 10 we show the smoothing factor changes for MIT-States [20] dataset. Note that for $\alpha = 1$, \mathcal{L}_{AO}^{NE} is same as \mathcal{L}_{AO} . Hence, the range to vary α is between [0,0.9]. With $\alpha = 0.8$ in Table 10, the correct label logit is weighted with

Table 10. Ablation for different smoothing factors: We set different smoothing factors α and T for \mathcal{L}_{AO}^{NE} with dataset MIT-states [20], to empirically find the best value for α and T .

α	Test@1	HM	$(AO)^*$	AO^*	A^*O	A^*O^*
0.3	2.22	10.38	13.25	3.85	11.11	5.18
0.5	2.34	10.56	16.75	5.67	7.56	4.16
0.8	2.41	10.94	18.87	5.49	8.24	3.54
0.9	2.30	10.87	17.49	4.85	8.12	2.79
<hr/>						
T						
0.2	2.38	10.64	18.74	5.07	8.46	2.90
0.5	2.41	10.94	18.87	5.49	8.24	3.54
0.7	2.37	11.06	18.72	5.10	8.37	3.15
0.9	2.37	10.97	18.63	5.01	8.65	3.09

factor $1 - \alpha = 0.2$, whereas other logits are weighted with $0.8/(K - 1)$.

For label propagation among n neighbors, we use $T = 0.5$, which means that the correct label is weighted with 0.5, as the rest neighbors are weighted with $0.5/n$. Hence, α and T are both smoothing factors but for different purposes. Moreover, the lower the smoothing factor, the more weightage is given to neighbors/other labels, and leads to higher accuracy for unseen attribute-unseen object pair. NEL uses both label smoothing and propagation smoothing factors. Using higher value of T leads to better generalizability but less learnability. With higher T , generalizability reduces, since more weightage is given to the single correct label. In Table 10, we fix each smoothing factor and vary the variable for finding the best hyperparameters.

Different ratio of NEL and main loss In equation 4, γ_1 and γ_2 control the weights for attributes and object losses. We follow same values of γ_1 and γ_2 , as mentioned in OADis [43]. Moreover, the total loss for each embedding space is weighted sum of Cosine Cross-Entropy and Neighborhood Expansion Loss as shown in equation 4. We show experiments for β_1 in Table 12. Further values of β_2 and β_3 also follow the similar results. Based on eq 4. the weight for \mathcal{L}_{AO}^{NE} is $1 - \beta_1$. The results show that giving more weight to Neighborhood Expansion Loss \mathcal{L}_{AO}^{NE} , harms the seen pair accuracy while keeping unseen pair accuracy. We use 0.8 as final value as it is a trade-off between seen and unseen accuracy. Since there are various unseen composition splits to evaluate on for OV-CZSL, *i.e.* $\{(AO)^*, AO^*, A^*O, A^*O^*\}$, we select the α based on only Test AUC and HM. AUC automatically provides the best trade-off between the seen and unseen accuracy overall and provides the best balance between \mathcal{L}_{AO} and \mathcal{L}_{AO}^{NE} .

Table 11. **Ablation on different Text Embedding.** We show effect of using different text embeddings for OV-CZSL. We observe that BERT [5] improves the unseen compositions A^*O^* the most, hence we use BERT [5] for our approach.

Emb	Test@1	HM	AO	$(AO)^*$	A^*O	AO^*	A^*O^*
GloVe	2.37	10.46	13.13	19.11	10.70	3.83	0.69
word2vec	2.37	10.68	13.82	17.46	10.61	4.19	1.13
fasttext	2.33	10.14	13.02	17.04	10.93	4.88	1.64
BERT	2.41	10.94	14.11	18.87	8.24	5.49	3.54

Table 12. Ablation for weights for Neighborhood Expansion Loss: We set different β_1 , weights for neighborhood expansion loss for pair embeddings, for MIT-states [20] split 1. We show that using higher value of β_1

β_1	Test@1	HM	$(AO)^*$	AO^*	A^*O	A^*O^*
0.7	2.23	10.50	16.80	4.83	8.67	4.10
0.8	2.41	10.94	18.87	5.49	8.24	3.54
0.9	2.30	10.78	19.87	5.25	7.05	2.53

8.3. Open-World evaluation

Current evaluation setup in the paper, is closed world. That means the model is only evaluated on valid compositions of seen and unseen attribute-object pairs. However, in real world, more realistic setting would be open world, where during evaluation, we do not know valid compositions. CompCos [29] and KG-SP [21] are two state-of-the-art methods for Open World evaluation for CZSL task. However, when we train our splits for open world evaluation, these SOTA methods do not perform well on the unseen compositions: $\{AO^*, A^*O, A^*O^*\}$. To make sure the model can identify the invalid compositions such as “ripe dog”, CompCos [29] and KG-SP [21] use feasibility scores, which removes the invalid compositions from feature space during inference. Moreover, this feasibility score is used during training as well, to avoid transfer of knowledge from seen compositions to unseen and invalid compositions. KG-SP [21] uses ConceptNet to compute the feasibility scores. We use the open-world evaluation for MIT-States [20] OV-CZSL benchmark, for CompCos [29] and KG-SP [21]. However, while evaluating in Open-world scenario, both CompCos [29] and KG-SP [21] fail to generalize to unseen composition $\{AO^*, A^*O, A^*O^*\}$. Our hypothesis is that for CZSL, all attributes and objects are seen. Hence, feasibility of pairs of seen attributes and objects, but unseen compositions can be computed through various means, such as using feasibility scores of ConceptNet features, which is used during training to transfer knowledge to valid compositions. However, for OV-CZSL, when unseen attributes and objects are not known during training,









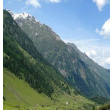
	$(AO)^*$	A^*O	AO^*	A^*O^*
MIT-States	 ancient clock engraved metal engraved coin	 blunt sword blunt blade curved blade	 broken laptop old laptop broken computer	 cooked fish cooked salmon sliced fish
C-GQA	 brown trunk large ear black tail	 large mirror gold mirror framed mirror	 young girl blond girl white shirt	 white horse white carriage standing horse
VAW-CZSL	 clay court running player wide court	 multicolored kite flying kite colorful kite	 white t-shirt dressed man baggy shorts	 grassy hill brown mountain tall mountain

Figure 5. Qualitative Results: We show results on MIT-States [20] for different sets, examples with top 3 predictions for an image. Most incorrect labels are also from unseen compositions Y^u , and although capture the attributes in the images, but are not considered correct. Labels in green are the correct annotation. We also observe redundancy in labels, such as row 1, column 2 `blunt blade` in A^*O . Top prediction also includes `blunt sword`. All swords have blades. However, having separate labels for these makes the task harder.

the feasibility scores describing valid compositions are only used during evaluation. Hence, the model fails to learn the validity of scores, during inference and fails to perform for the unseen compositions. Therefore, we conclude that OV-CZSL is more challenging setup than CZSL, and currently can be evaluated in Closed-world setup only.

9. Qualitative Results

We show qualitative results on the same datasets. In Figure 5, we first show for different sets (seen and unseen pairs), examples of top 3 predictions for an image. All the predicted labels are part of unseen compositions Y^u . Although predictions are not always right, they still represent the concepts present in the images. The current evaluation protocol do not evaluate based on if either of attribute or object is correctly identified, the image should be considered partially correctly classified in compositional task. Another aspect we observe is that some attributes are also present in the current vision+language models are tied to correct labels, even if other labels also are correct. For instance, “blunt blade” in row 1 Figure 5, A^*O is also “blunt sword”. All swords have blades, but with separate labels the task is harder. We urge the community to direct the dataset creation such that visual concepts are classified as one categories, as opposed to language driving the categorization of visual concepts. Further, we show some qualitative results to highlight that our model can still learn and generalize to out-of-domain data. In instance, in Figure 6 (a), We show the 5 nearest neighbor images from VAW-CZSL [37, 43] dataset, using the composition feature labels from model trained on MIT-states [20]. Similarly, (b) shows that if model is trained on VAW-CZSL [37, 43], it’s nearest neighbors in feature space for various unseen compositions make sense, when retrieved from MIT-sates [20]. Moreover, we observe that

for all compositional sets, $\{(AO)^*, AO^*, A^*O, A^*O^*\}$, the model learn the attribute and object features to generalize to totally unseen images from different dataset.

10. Limitations.

We emphasize that OV-CZSL is challenging setup since the problem to discriminating visually similar concepts is still restricted. For instance, a model can not discriminate `peel` from `slice` and `chop`. However, we propose this direction to at least learn that `peel`, `slice` and `chop` are all applied to fruits. Hence, if training set has `sliced apple`, in test set our model should be somewhat be confident in predicting `chopped pear`. With this current setting of OV-CZSL, we do not claim to differentiate `peeling` from `chopping`, but we propose that `sliced potato` and `chopped pear` are visually more closer than `sliced potato` and `raw apple`.

Moreover, our evaluation protocol only evaluates for valid compositions, also referred to as Closed-world setting which is a standard evaluation metric for CZSL benchmarks [39, 43]. The other evaluation setting uses all possible compositions for attributes and objects and discard the invalid compositions during testing. OV-CZSL is an extension of CZSL with using wider use of ‘open-vocabulary’ for training. However, the during testing, we only evaluate on valid test compositions, rather than for every possible composition like open-world setting. We quantitatively show that using open-world evaluation setting is even more challenging for OV-CZSL but can be potential direction for future research.

Large Language+Vision Models. We emphasize that we acknowledge the presence of larger models, however choose to not include those in this study since the focus of this work is to lean beyond seen samples, while all these

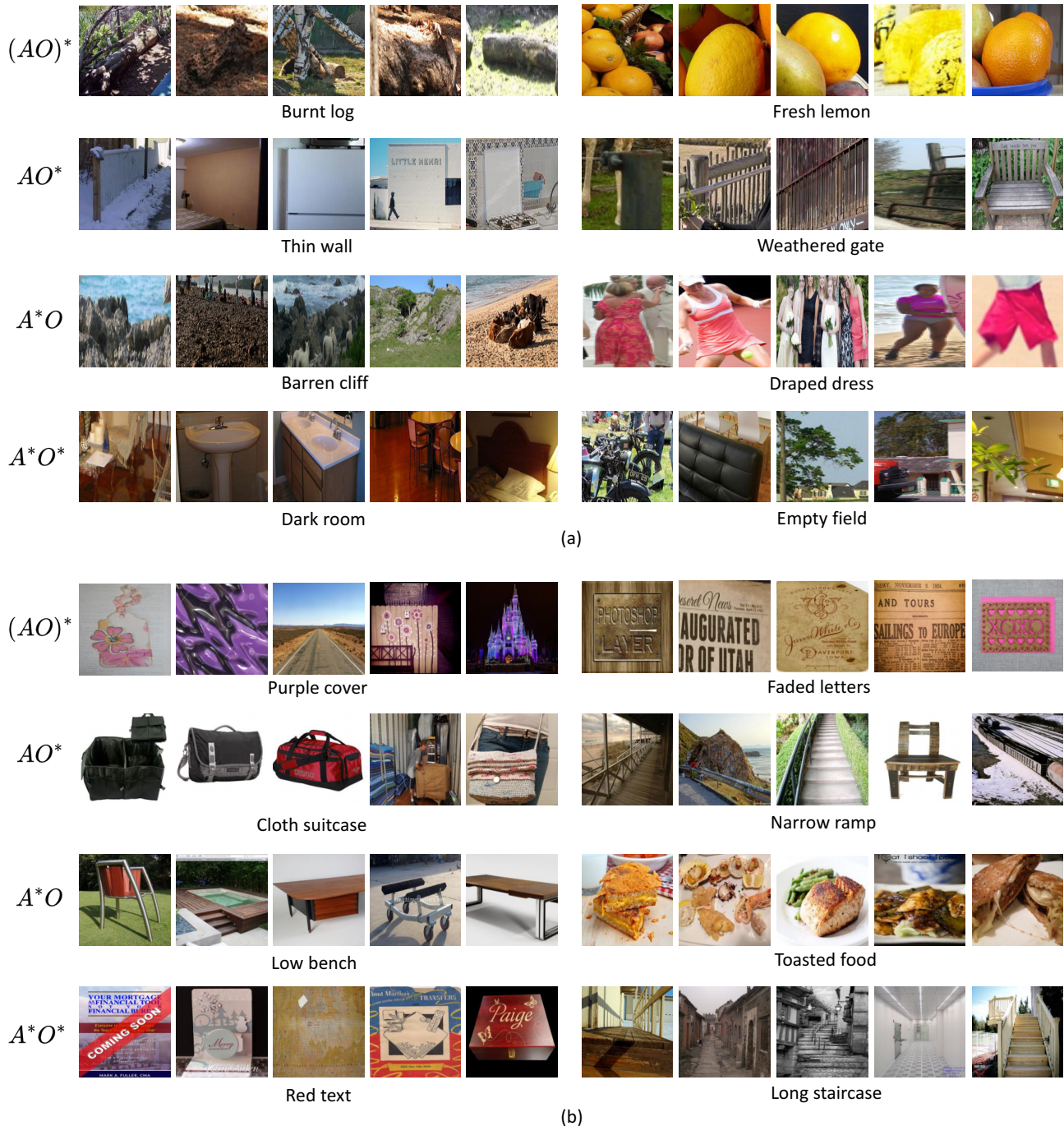


Figure 6. **Qualitative results for Cross-Dataset Nearest neighbors.** (a) We show the 5 nearest neighbor images from VAW-CZSL [37, 43] dataset, using the composition features from model trained on MIT-states [20]. Similarly, (b) shows that if model is trained on VAW-CZSL [37, 43], it's nearest neighbors in feature space for various unseen compositions make sense, when retrieved from MIT-sates [20].

models already are trained on exponential amount of data such that no category from academic datasets is unseen for these models. Moreover, including CLIP [40] in main paper Table 5, we show that our method can help these models improve on subtle attributes fine-tuned learning, but is still not useful for Zero-shot learning setup. Hence, we want to

clarify the scope and limitation of this work to urge the reviewers evaluate this work under the same constraints and limitations.