

Exploring Long Tail Visual Relationship Recognition with Large Vocabulary: Supplementary Material

Table of Contents

1. Qualitative Examples
2. Implementation Details
3. Hyperparameters
4. Many, Medium, Few splits for GQA-LT and VG8K-LT
5. Object/Subject/Relationship class frequencies for GQA-LT and VG8K-LT
6. Human Subjects Experiment Setup
7. Motivation For Word2Vec and Wordnet Metrics
8. Additional Results on GQA-LT and VG8K-LT
9. Further discussion of RelMix Augmentation
10. Additional Results on VG200 (far more balanced than GQA-LT and VG8K-LT)
11. Further Analysis
12. Code (attached, includes implementation details) : <https://github.com/Vision-CAIR/LTVRR>
13. Video (attached, includes more qualitative examples)
14. Dataset histograms that show the distribution of classes for subject, objects, and relations provided under `./histograms`
15. GQA-LT and VG8K-LT synsets to classes mapping:
 - `./synsets_mapping/gqa_rel_synset_mapping.json`
 - `./synsets_mapping/gqa_sbjobj_synset_mapping.json`
 - `./synsets_mapping/vg_sbjobj_synset_mapping.json`
 - `./synsets_mapping/vg_rel_synset_mapping.json`

1. Qualitative Examples

Fig 1 shows an example of one of the cases where the LSVRU model (left image) predicts a head class (*to the left of*) that doesn't fit well while the LSVRU + ViHub model (right image) instead predicts a tail class (*eating*) which is more accurate and descriptive in this case.

Similarly, Fig 2 shows an example of one of the cases where the LSVRU model (left image) predicts a head class (*to the right of*) that doesn't fit well while the LSVRU + RelMix + ViHub model (right image) instead predicts a tail class (*holding*) which seems more suitable and descriptive for the particular triplet in question.

Fig 3 shows a failure case on the head, when the LSVRU + ViHub predict a tail class while the correct class is a head class. More qualitative examples can be found in the attached video.

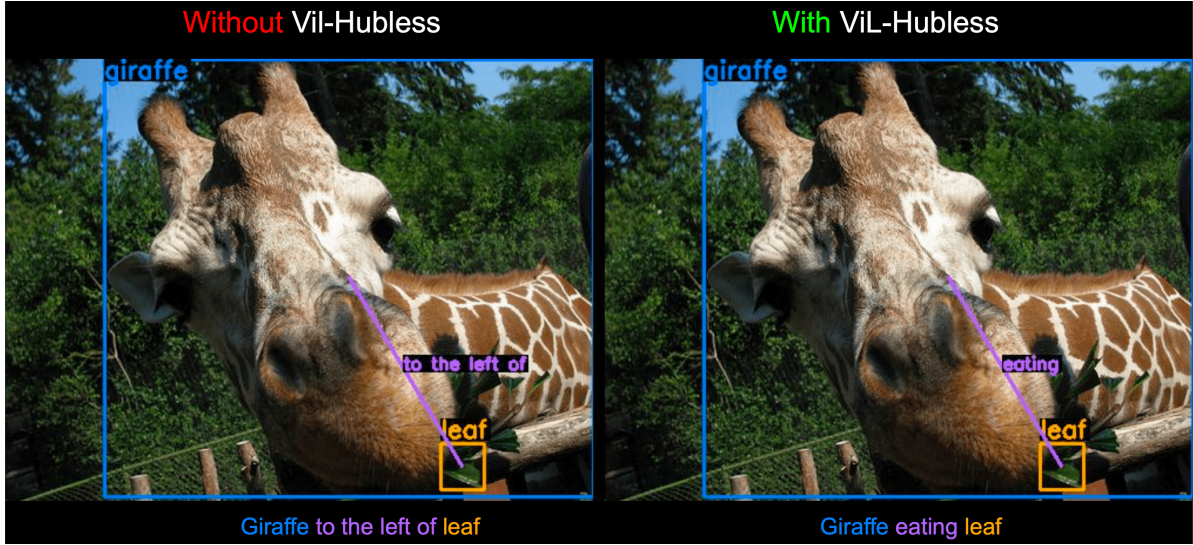


Figure 1: A qualitative example showing how the model with the ViLHub loss performs better on tail relation classes. Blue is subject, purple is relation, and orange is object. The left image is the LSVRU model, and the right image is LSVRU + ViLHub model



Figure 2: A qualitative example showing how the model with the RelMix augmentation and ViLHub loss performs better on tail relation classes. The left image is the LSVRU model, and the right image is LSVRU + RelMix + ViLHub model

2. Implementation Details

Visual Embedding sub-network. Similar to [6, 8], we learn embeddings for subject and object in a separate semantic space from the relation space. More concretely, we first get a global feature map of an input image processing in through a CNN (*conv1_1* to *conv5_3* of VGG16), then we perform ROI-pooling of subject, relation and object features to get z^s , z^p , z^o with the corresponding regions \mathcal{R}_S , \mathcal{R}_P , \mathcal{R}_O . Each branch followed by two fully connected layers which output three intermediate hidden features h_2^s , h_2^p , h_2^o . For the subject/object branch, a fully connected layer w_3^s is added to get the visual embedding x^s , and similarly for the object branch to get x^o . Since we expect the network to recognition the object whether it appeared as as a subject or an object in a relationship, all the parameters of both branches are shared. Since involving relation features for subject/object embeddings may undesirably entangling the two spaces, For the relation branch following [8], we apply an effective two-level feature fusion to finally get the relation embedding x^p .

Language Embedding sub-network. On the language side, we feed word vectors of subject, relation and object labels into a two-layer neural network of one or two *fc* layers which outputs the final embeddings. Similar to the visual module, we share subject and object branches share weights while the relation branch is unshared. The purpose of this module is to map

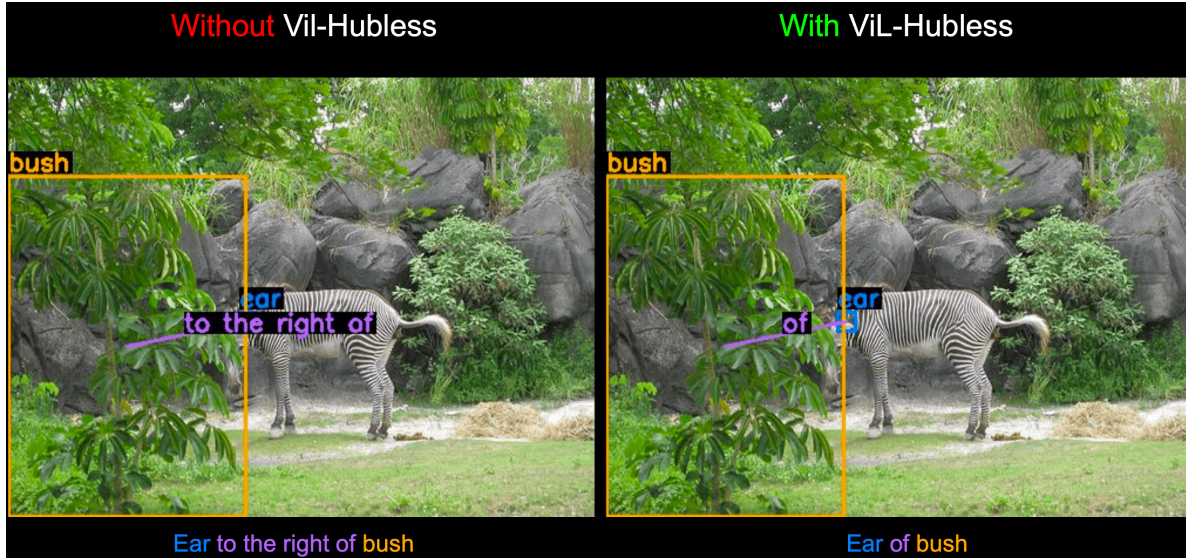


Figure 3: A qualitative example showing how the model with the ViLHub sometimes fails by predicting a tail class instead of a head class. The left image is the LSVRU model, and the right image is LSVRU + ViLHub model

word vectors into an embedding space that is more discriminative than the raw word vector space while preserving semantic similarity. During training, we feed the ground-truth labels of each relationship triplet as well as labels of negative classes into the semantic module, as the following subsection describes; during testing, we feed the whole sets of object and relation labels into it for nearest neighbors searching among all the labels to get the top k as our prediction.

Language and Visual Context Word Embeddings. The language sub-network takes as an input skip-gram word Embeddings, which tries to maximize classification of a word based on another word in the same context. We performed experiments with two skip-gram models trained on language and visual contexts. The language word embedding model is provided by word2vec [5], pre-trained on Google News corpus as context. The second *visual word embedding model* is trained with the same loss of a skip-gram word2vec model where the context is defined as the training relationship instances. The optimization maximizes the likelihoods of each relationship element given the other two (e.g., each of S, R, and O given SO, RO, SR, respectively).

3. Hyperparameters

All models are trained with a base learning rate $LR = 0.01$ on 8 V100 gpus with a batch size of 1 image per batch and 512 boxes within a single image per batch. All models trained on GQA-LT were trained for 12 epochs, and all models trained on VG8K-LT were trained for 8 epochs. Models trained on GQA-LT were started with a random seed of 0 (for Numpy and Pytorch), and models trained on VG8K-LT were started with a seed of 3 (for Numpy and Pytorch). The train/val/test splits for both datasets are provided with the attached code.

4. Many, Medium, Few splits for GQA-LT and VG8K-LT

As discussed in the main paper, we split VG8K-LT data into *many*, *medium*, and *few* shots based frequency percentiles, *many*: top 5% most frequent classes, *medium*: middle 15%, and *few*: bottom 80%. Here we will give details on the range of classes withing each split for both GQA-LT and VG8K-LT.

Tables 1 and 2 shows the number of classes in various categories for GQA-LT and VG8K-LT respectively. The split is shown for both the training and testing data, and also the number of synsets in all the categories is also shown. For full splits and detailed information about this, please refer to the csv files under `./histograms`, provided in the supplementary material.

5. Object/Subject/Relationship class frequencies for our GQA-LT and VG8K-LT Benchmarks

The Fig 4 shows the Subject, Object, Relationship class frequencies for GQA-LT and VG8K dataset. While the values here shown in graphs are in log scale, the frequencies for sbj/obj/rel for both of the dataset can be found in the code folders

Table 1: GQA-LT split

	Number of classes						Number of unique synsets					
	subjects/objects			relations			subjects/objects			relations		
	many	med	few	many	med	few	many	med	few	many	med	few
Train	75	223	1394	15	43	252	71	207	1078	11	33	143
Test	75	223	1188	15	43	232	71	207	955	11	33	132

Table 2: VG8K-LT split

	Number of classes						Number of unique synsets					
	subjects/objects			relations			subjects/objects			relations		
	many	med	few	many	med	few	many	med	few	many	med	few
Train	154	464	4639	72	212	1715	152	447	2628	38	93	358
Test	154	460	2453	72	214	1143	152	444	1678	38	93	290

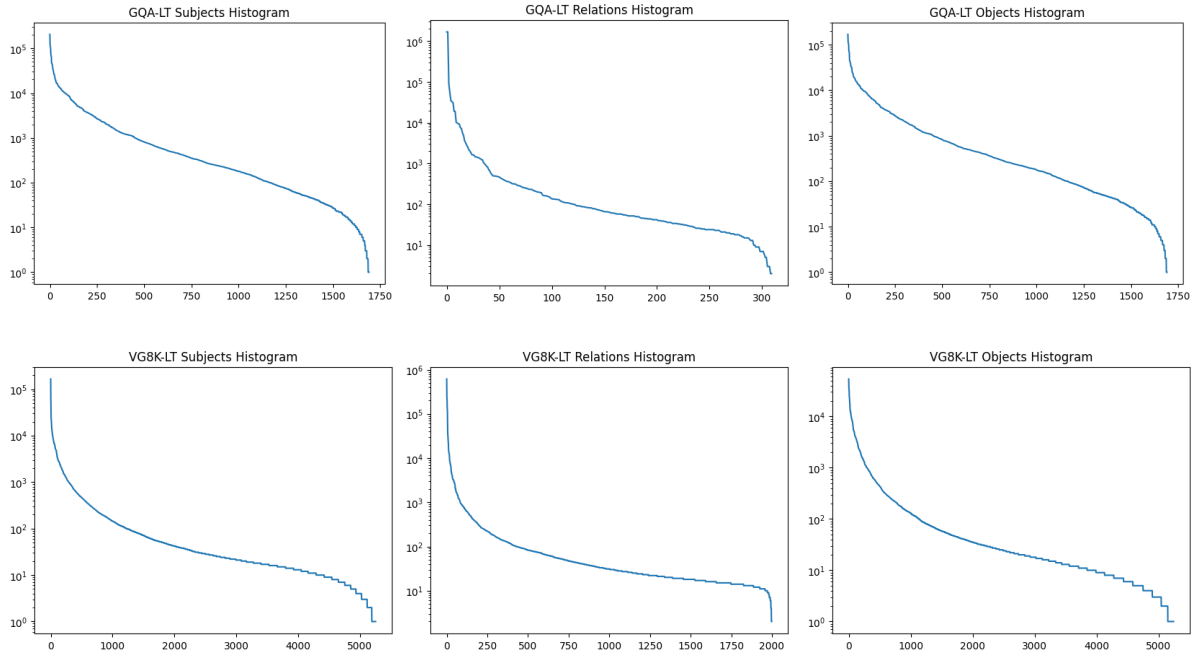


Figure 4: The histograms showing the sbj, rel, obj frequencies for the GQA-LT and VG8K-LT dataset. The figures shows the frequency values in log scale. The actual frequencies for each class are attach in csv format.

provided alongside the supplementary submission.

6. Human Subjects Experiment Setup.

We randomly selected 100 examples from Visual Genome dataset [2] and evaluated 5 hypotheses for each. Out of these 5 hypotheses, 1 was the Ground Truth (GT) and the other 4 were top predictions from [8] excluding GT. In this experiment we had 3 human subjects, who were asked to evaluate each hypothesis from a scale of 1 to 5, 1 being the worst and 5 the best. The human subjects were blind to which hypothesis was the ground truth and which were a prediction by the model [8].

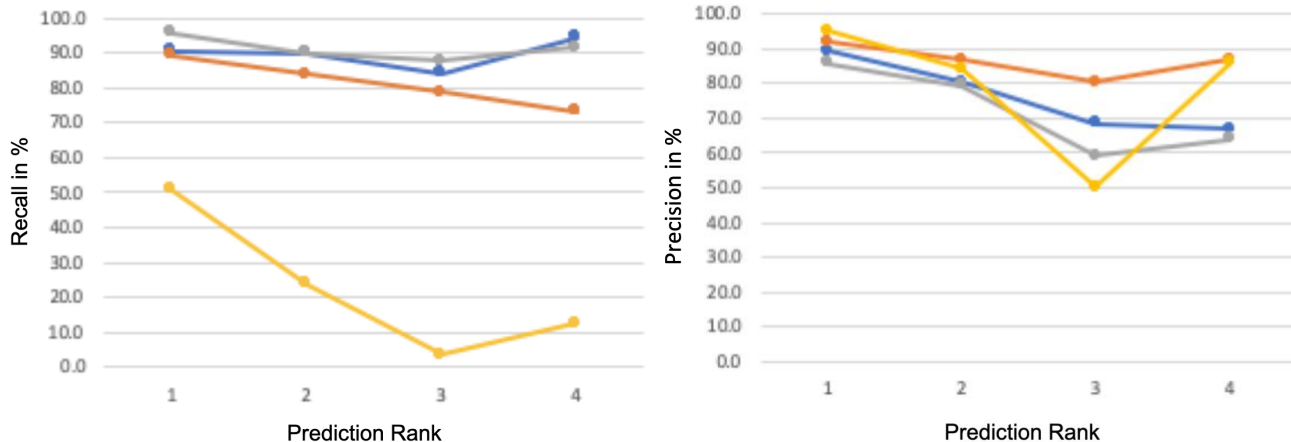


Figure 5: Precision and Recall scores of each of the human subjects (gray, blue, red) and the ground truth (gold) against the Human-GT

Afterward we created a new ground truth from the majority voting of the 3 subjects and the ground truth on each example, we call this new ground truth Human-GT.

We then computed the precision and recall of each of the human subjects and the ground truth against the Human-GT and we show the recall and precision in Fig 5. If we look at Fig 5 we can see that the ground truth has a very low recall compared with the human subjects. This implies that a very large percentage of the hypotheses labeled as incorrect by the ground truth is, in fact, correct (high number of false negatives). This confirms our suspicion that the GT on its own is not sufficient to gain a deep understanding of how the models are performing on this problem. Note, because the gap in recall between the GT and human subjects was large enough, 100 random examples are enough to reach confidence level of 99.73%. We did not observe a difference in precision between the human subjects and the GT. This implies is that the labels that GT labels as correct are also considered correct by the Human-GT (low number of false positives).

7. Motivation For Word2Vec and Wordnet Metrics

The ground truth by construction assumes that there is one and only one right answer. Fig 6 shows an example of such case. In this case, the top 5 predictions from the model are all correct and very plausible. It's very hard to say any of these are wrong. There are many examples as the one in Fig 6 throughout the 2 datasets (GQA-LT and VG8K-LT). This illustrates that the ground truth with only one correct answer is fundamentally flawed for this task, and this is the motivation behind our proposed metrics.

8. Additional Results on GQA-LT and VG8K-LT

Table 6 shows some of the main models from the paper on GQA-LT dataset. The scores shown in Table 6 are calculated over several runs for each model (between 2 and 3 runs) and the mean and confidence intervals (calculated at confidence=95%) are reported. The table shows that most improvement are outside the margin of error. This further strengthens our confidence in the results reported in the main paper. In Table 5 shows the average per-class word similarity measured through wordnet and word2vec metrics for the subject and object categories. We can see the pattern more consistently here, where the models with the ViHub loss added have higher average per-class word similarity to the ground truth.

Table 3 and 4 show the performance on subject, relation, object (SRO) triplets scores on GQA-LT and VG8K-LT, respectively. An SRO triplet prediction is considered correct if the prediction for the subject, object, and relation are all correct. We separate the SRO triplets into few, medium, and many shots based on how many times the SRO triplet occurs in the training data. As we did with the subject/object and relation tables in the main paper, we determine many, medium, few shots based on frequency percentile. (many: top 5% most frequent, medium: middle 15%, and few: bottom 80%). Table 3 and 4 show that adding the ViHub loss and ReMix augmentation increases the performance on almost all the cases.

Tables 7, 8 and 9 show the compositional results (the results when grouped by SO, SR and OR) on many, med and few categories respectively. In all these, we see a clear gain when adding ViHub & ReMix on top of various base models.



Figure 6: **This example is meant to show how some boxes can have multiple good answers.** The top 5 predictions for the above box are: [Baseball Cap, Cap, Green Hat, Hat, Head]. This shows how it is unreasonable to evaluate this task assuming there is only one correct answer.

Table 3: Average per-relationship triplet accuracy on GQA-LT using Synsets, showing benefit to adding ViHub and RelMix in most models

Model	many	median	few	all
LSVRU [8]	42.8	25.8	9.0	13.2
LSVRU + ViHub	44.3	30.0	11.7	16.0
LSVRU + ViHub + RelMix	45.5	30.9	12.2	16.6
Focal Loss [3]	43.2	27.5	9.7	14.1
Focal Loss + ViHub	44.4	29.8	11.5	15.9
OLTR [4]	42.6	26.0	9.1	13.3
OLTR + ViHub	41.8	26.4	9.7	13.7
WCE	19.4	14.8	7.6	9.2
WCE + ViHub	18.0	14.8	7.9	9.4
WCE + ViHub + RelMix	27.3	21.1	10.2	13.2
DCPL [1]	33.7	20.2	7.3	10.6
DCPL + ViHub	29.3	19.6	7.8	10.6
DCPL + ViHub + RelMix	31.3	20.8	8.3	10.9
EQL [7]	44.4	30.0	11.7	16.0
EQL + ViHub	42.3	31.1	12.8	16.8
EQL + ViHub + RelMix	44.6	31.8	12.9	17.0

Table 4: Average per-relationship triplet accuracy on VG8K-LT using Synsets, showing benefit to adding ViLHub and RelMix for most models

Model	many	median	few	all
LSVRU [8]	24.0	10.1	3.1	5.2
LSVRU + ViLHub	28.3	13.3	4.1	6.7
LSVRU + ViLHub + RelMix	29.1	13.7	4.3	6.9
Focal Loss [3]	21.2	9.7	2.9	4.9
Focal Loss + ViLHub	25.1	11.4	3.5	5.7
WCE	12.1	5.8	2.4	3.4
WCE + ViLHub	11.5	5.6	2.3	3.3
WCE + ViLHub + RelMix	14.6	6.9	3.1	4.1
DCPL [1]	15.8	6.6	2.5	3.8
DCPL + ViLHub	17.1	7.2	2.5	4.0
DCPL + ViLHub + RelMix	17.6	7.3	2.6	4.1

Table 5: Per-class word similarity on subjects/objects in GQA-LT, models with ViLHub show higher average word similarity

Models	lch	wup	lin	path	w2v
LSVRU [8]	51.2	59.4	36.8	27.5	45.2
LSVRU + ViLHub	53.4	61.3	39.5	30.6	47.1
Focal Loss [3]	51.8	60.0	37.5	28.3	45.7
Focal Loss + ViLHub	53.2	61.1	39.1	30.3	47.0
WCE	53.5	61.1	39.4	31.8	47.8
WCE + ViLHub	54.8	62.1	41.0	33.5	49.2
DCPL [1]	49.1	57.4	34.1	25.8	43.0
DCPL + ViLHub	50.4	58.4	35.5	27.3	44.7

Table 6: Average per-class accuracy on GQA-LT based on Synsets, performed over several runs for each model and shows the mean and confidence intervals calculated at confidence=95%

Model	sbj/obj				rel			
	many	medium	few	all	many	medium	few	all
LSVRU [8]	68.6±0.6	38.1±2.2	7.1±0.4	14.8±0.7	62.4±0.28	16.0±0.83	7.4±1.2	11.6±1.1
LSVRU + ViLHub	69.3±0.6	40.4±0.9	8.0±0.2	15.9±0.2	63.5±0.1	17.5±0.2	7.5±0.3	11.8±0.3
Focal Loss [3]	68.7±0.6	39.9±0.8	7.6±0.1	15.5±0.2	60.5±0.2	15.9±0.6	7.6±0.5	11.6±0.5
Focal Loss + ViLHub	69.3±0.3	44.0±0.8	9.7±0.2	17.8±0.3	62.8±0.5	14.2±0.3	7.2±0.5	11.1±0.5
WCE	52.9±1.5	40.2±2.0	13.7±0.5	19.7±0.5	52.2±2.2	37.9±3.7	14.5±2.5	20.0±2.2
WCE + ViLHub	50.8±1.2	43.4±1.2	16.9±0.9	22.6±0.5	53.4±0.8	37.2±2.2	14.4±1.7	19.8±1.6

Table 7: The table shows the average per-group relationship triplet performance for the case of 'many' occurring classes on GQA-LT based on synsets. The results are shown when grouped by Subject & Object (SO), Subject & Relation (SR), Object & Relation (OR). The table shows a performance increase with the addition of RelMix + ViLHub

Model	SO	SR	OR
LSVRU [8]	38.6	30.3	31.5
LSVRU + ViLHub	40.5	32.8	33.7
LSVRU + ViLHub + RelMix	41.7	33.8	33.8
FL [3]	39.2	31.1	32.3
FL+ViLHub	40.5	32.8	33.7
WCE	18.3	17.3	17.2
WCE + ViLHub	17.0	17.0	16.8
WCE + ViLHub + RelMix	25.1	23.1	22.5
DCPL [1]	30.2	24.7	25.1
DCPL + ViLHub	26.8	23.4	23.2
DCPL + ViLHub + RelMix	28.7	24.4	24.9

Table 8: The table shows the average per-group relationship triplet performance for the case of 'medium' occurring classes on GQA-LT based on synsets. The results are shown when grouped by Subject & Object (SO), Subject & Relation (SR), Object & Relation (OR). The table shows a performance increase with the addition of RelMix + ViLHub

Model	SO	SR	OR
LSVRU [8]	21.8	11.3	10.8
LSVRU + ViLHub	25.7	14.2	13.9
LSVRU + ViLHub + RelMix	26.6	14.7	13.8
FL [3]	23.2	11.9	11.5
FL + ViLHub	25.5	14.1	13.7
WCE	13.7	9.4	9.4
WCE+ViLHub	13.7	9.6	9.5
WCE + ViLHub + RelMix	18.1	12.8	13.0
DCPL [1]	17.2	9.2	9.0
DCPL + ViLHub	17.2	9.5	9.2
DCPL + ViLHub + RelMix	18.1	9.9	9.8

Table 9: The table shows the average per-group relationship triplet performance for the case of 'few' occurring classes on GQA-LT based on synsets. The results are shown when grouped by Subject & Object (SO), Subject & Relation (SR), Object & Relation (OR). The table shows a performance increase with the addition of RelMix + ViLHub

Model	SO	SR	OR
LSVRU [8]	7.5	4.3	4.2
LSVRU + ViLHub	10.2	5.3	5.2
LSVRU + ViLHub + RelMix	10.5	5.5	5.6
FL [3]	8.2	4.3	4.2
FL + ViLHub	10.0	5.1	4.9
WCE	7.1	4.2	3.6
WCE + ViLHub	7.4	4.7	3.9
WCE + ViLHub + RelMix	9.5	5.7	5.1
DCPL [1]	6.3	3.3	3.2
DCPL + ViLHub	6.8	3.7	3.4
DCPL + ViLHub + RelMix	7.5	3.9	4.0

9. Further discussion of RelMix Augmentation

Table 10: Some ablations for different proportions of augmented data being added to the original data

Model	sbj/obj				rel			
	many	medium	few	all	many	medium	few	all
LSVRU [8]	68.3	37.0	6.9	14.5	62.6	15.5	6.8	11.0
LSVRU + RelMix ($\eta = 30\%$)	68.2	37.5	8.5	15.7	62.5	15.6	6.8	11.0
LSVRU + RelMix ($\eta = 50\%$)	68.2	37.7	9.3	16.5	62.6	16.0	6.9	11.1
LSVRU + RelMix ($\eta = 70\%$)	68.5	37.7	9.1	16.3	62.8	16.2	6.9	11.2

In Table 10, we evaluate RelMix performance when different proportions ($\eta = 30\%, 50\%, 70\%$) of augmented data (w.r.t the original training size) is added to the original data mix. As can be seen from Table 10, we see slightly better results for $\eta = 50\%$ as compared to other proportions. Apart from this as we increase the augmentation proportion we see a slight gain in training time because of the increment in training data (original + augmented). Taking all this in account, we chose $\eta = 50\%$ for all the experimental results mentioned in Table 1&2 in the main paper.

Choice of λ in Eq.6 in main paper: We validate various values for our mixing parameter λ ranging from 0.5 – 0.9 and also try random value assignment within the range. We obtain the optimal results on tail classes being when λ is in the range 0.7 – 0.8, with the difference being of approximately 0.5% in the *few* category and 0.3% overall. This is since a bigger λ value leads to higher percentage of features from \mathbf{x}_i augmented from tail categories.

10. Additional Results on VG200 (far more balanced than GQA-LT and VG8K-LT)

Table 11: The main results on VG200 dataset.

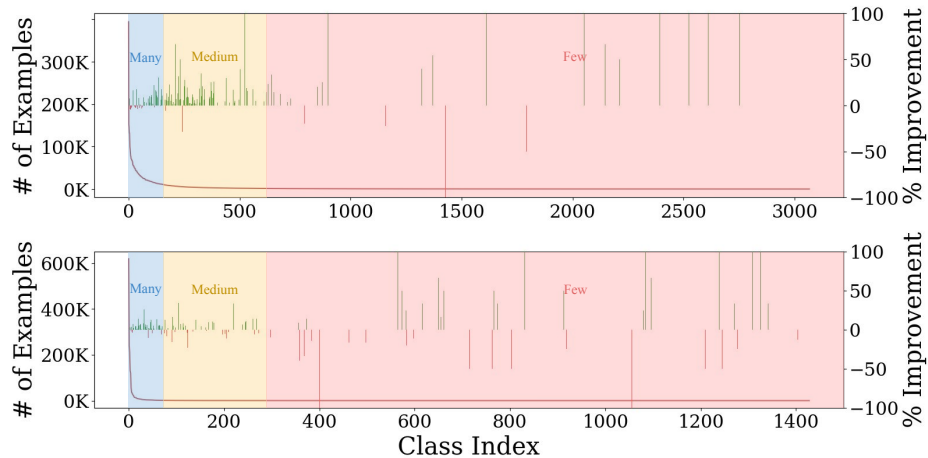
Models	Graph Constraint						No Graph Constraint			
	SGCLS			PRDCLS			SGCLS		PRDCLS	
Recall at	20	50	100	20	50	100	50	100	50	100
LSVRU [8]	36.0	36.7	36.7	66.8	68.4	68.4	-	-	-	-
LSVRU + ViHub	35.9	36.7	36.7	66.6	68.4	68.4	48.5	49.8	93.4	96.7
LSVRU + ViHub + RelMix	36.2	36.9	36.9	66.9	68.5	68.5	48.8	50.1	93.5	96.8

We also evaluate the performance of our proposed ViHub loss with RelMix augmentation on VG200 dataset. It contains most frequent 150 objects and 50 relations, and each category frequency in this dataset is considerably more balanced than in GQA-LT and VG8K-LT. We follow the same data split as in [8]. Table 11 shows the performance of our model on top of LSVRU when evaluated on VG200. As can be clearly seen, the proposed ViHub+RelMix does not deteriorate the base model’s performance on both the metrics (SGCLS and PRDCLS) and even manages to slightly improve upon it. However, a thing to keep in mind here is that our model manages to make the final prediction much more balanced (as can be seen from Table 1 and 2 in the paper) while not deteriorating the performance on these standard metrics (which are inherently much more biased towards *head* class classification).

11. Further Analysis

Figure 8 shows the same comparison done in the main paper in Figure 4, but for several other models, comparing the results with and without using the ViHub loss. We can observe that the same pattern of improving the performance on *medium* and *few* shots seen in the main paper still holds true for other models. The only exception is the performance on relationships for DCPL vs DCPL + ViHub, where we see classes worsening on the *few* category. However, ViHub loss still shows performance improvement on the *medium* category.

Figure 7 shows the comparison between LSVRU vs LSVRU + ViHub for VG8K-LT dataset. We can see that adding the ViHub improves performance on the *medium* and *few* categories, as it did on GQA-LT.



(a) LSVRU vs. LSVRU + ViLHub for S/O (upper) and R (lower) on VG8K-LT

Figure 7: Comparisons of subject/object (upper) and relations (lower) performances between LSVRU model with and without ViLHub on VG8K-LT dataset. Note that the number of classes is slightly less than the listed number of classes for VG8K-LT, this is because these are the classes present in the test set only.

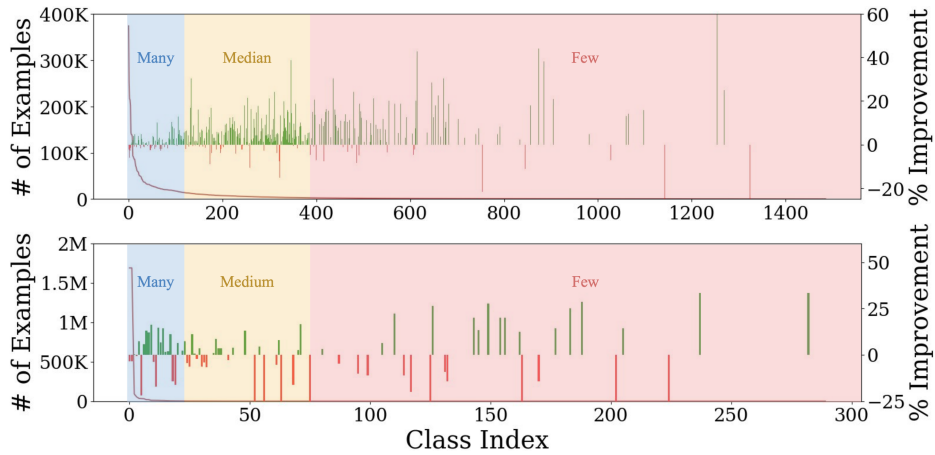
Figure 9 shows the average precision metric on the tail for W2V trained on Google News (W2V-GN). It shows the same patterns as in Figure 5 in the main paper. Note that the scores using W2V-GN is less than when using W2V-VG. This is because W2V-VG is trained on more a relevant data to the task (Visual Genome) than W2V-GN.

Figure 10 shows the same analysis done in the main paper section 4.5 Figure 5 but repeated for the head (top 20% of classes). We can observe the same patterns shown in the main paper, all the models are doing much better than the exact matching metric implies.

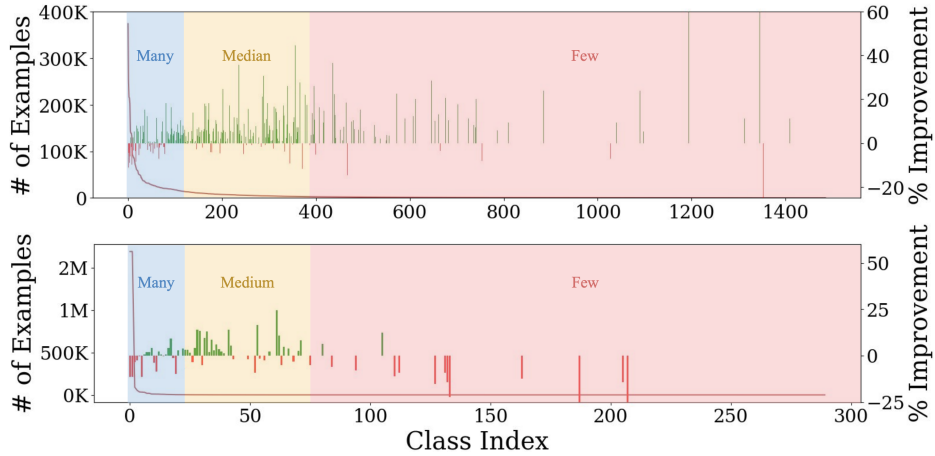
Figure 11 shows the average precision analysis for tail classes using the Relmix approach in combination with ViLHub.

References

- [1] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. 6, 7, 8, 9
- [2] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 4
- [3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6, 7, 8, 9
- [4] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019. 6
- [5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3
- [6] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1946–1955. IEEE, 2017. 2
- [7] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan. Equalization loss for long-tailed object recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11659–11668, 2020. 6
- [8] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9185–9194, 2019. 2, 4, 6, 7, 8, 9, 10



(a) FL vs. FL + ViLHub for S/O (upper) and R (lower) on GQA-LT



(b) DCPL vs. DCPL + ViLHub for S/O (upper) and R (lower) on GQA-LT

Figure 8: Comparisons of subject/object (upper) and relations (lower) performances between several models with and without ViLHub on GQA-LT dataset. We report the performance for all classes sorted by frequency. The distribution of classes for both figures is shown in the background. Note that the number of classes is slightly less than the listed number of classes for GQA-LT, this is because these are the classes present in the test set only.

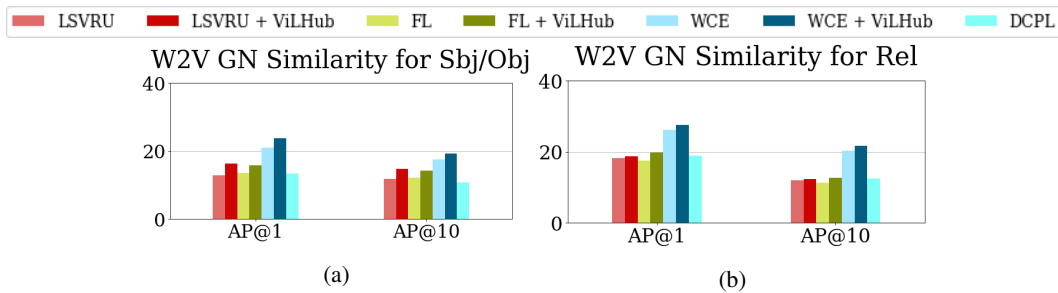


Figure 9: Average precision analysis on the tail classes (lower 80% on GQA-LT dataset) using a variety of metrics, calculated using W2V trained Google News, showing the same pattern as the figures in the main paper

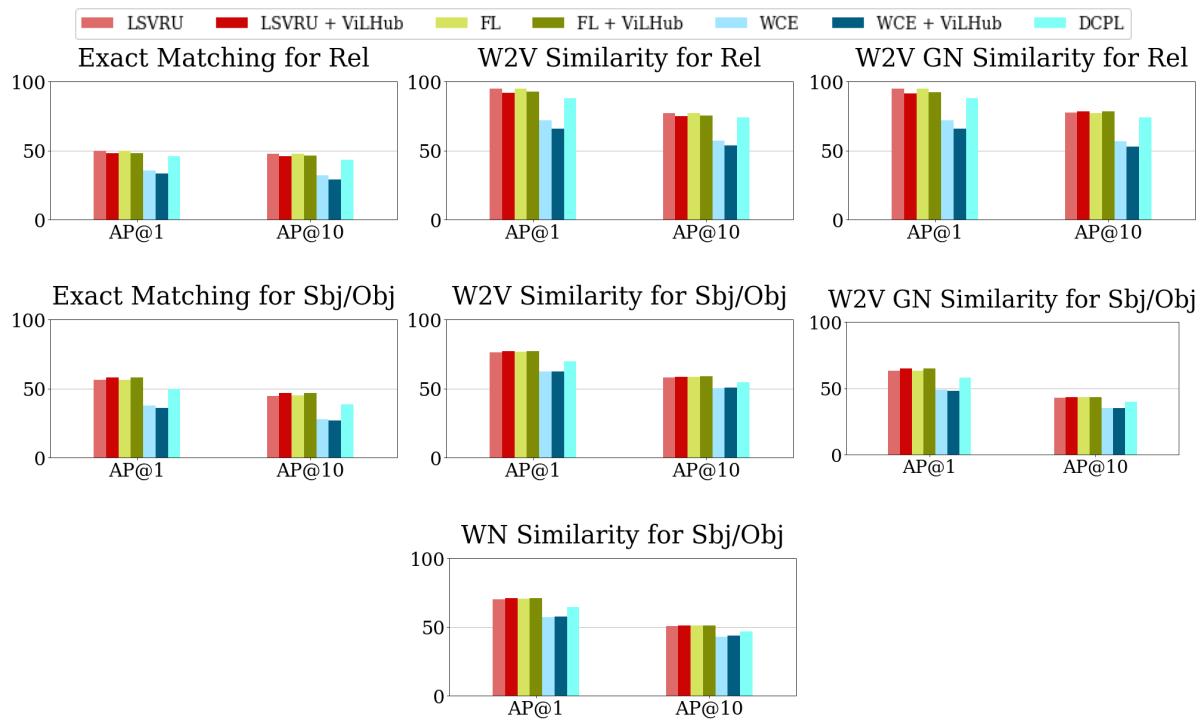


Figure 10: **Average precision analysis on the head classes (top 20% on GQA-LT dataset using a variety of metrics.** We visualize results using exact similarity metrics, W2V-VG, and average of 6 WordNet metrics. The models using ViLHub show consistently superior performance on the tail, when compared to similar models without the ViLHub.

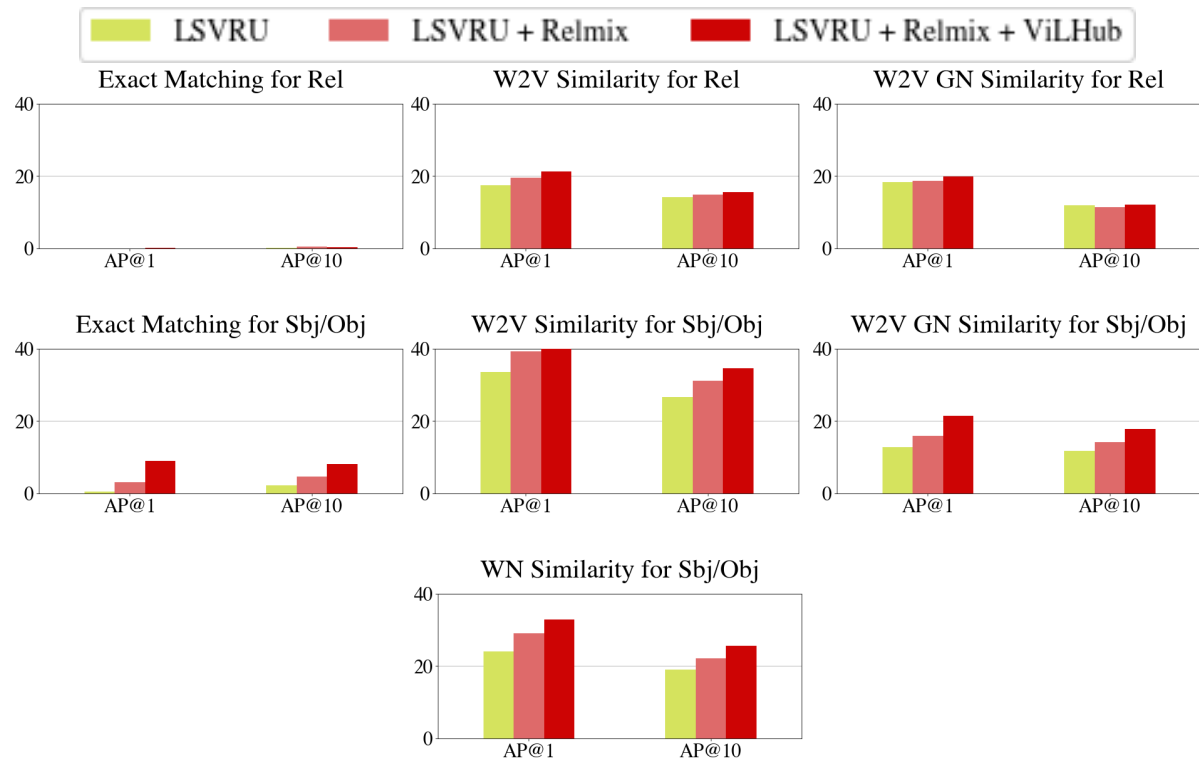


Figure 11: Average precision analysis on the tail classes (bottom 80% of classes) on GQA-LT dataset using the Relmix approach combined with ViLHub the figure shows the incremental improvement from adding Relmix augmentation and then ViLHub regularization