

A Appendix

A.1 Model Details and Hyperparameters

Unless otherwise noted, we follow the public implementation⁵ of [34]. Below, we provide additional details.

Relationship Attention Architecture. To obtain query, key, and value embeddings from the image embeddings for the Relationship Attention layer, we use 3-layer MLPs with no change in feature dimensionality, GeLU hidden activations [14], and a skip connection from the input to the output embedding. LayerNorm [2] is applied to the final output in the MLPs. To obtain the relationship embedding, `<subject>` and `<object>` embeddings are summed, normalized by LayerNorm, and processed by another 2-layer MLP (not shown in Figure 2). We found model performance to be robust to the details of the Relationship Attention layer, e.g. the hyperparameters of the MLPs, and it may be possible to simplify the design further. However, as noted in the main paper, `<subject>` and

`<object>` embeddings must be computed with different projections to model the asymmetry between `<subject>` and `<object>` in the relationship. If the same projection were used to compute a single embedding to represent both `<subject>` and `<object>`, the model could not distinguish between e.g. "person riding horse" and "horse riding person". Figure 6 confirms experimentally that sharing the MLP for subjects and objects performs poorly. With a shared MLP, the model struggles to learn and reaches a low final score.

Relationship Selection. In the Relationship Attention layer, we perform two rounds of top- k selection as depicted in Figure 2: First, we select the top 512 object instances, using the diagonal entries of the relationship score matrix as "objectness" scores. This reduces the size of the relationship score matrix from $N \times N$ (where N is the number of image encoder output tokens, i.e. object proposals) to 512×512 . From this matrix, we then select k relationships, where $k = 2^{14} = 16384$ unless otherwise noted. Additionally, we always compute embeddings for the 512 self-relationships along the diagonal (which represent object instances), since these embeddings will be necessary to classify the object categories of the `<subject>` and `<object>` boxes. Model performance is remarkably robust to the value of k both during training and inference. We did not observe a significant reduction in performance for k as low as 1024 either just during inference (Section 4.5) or during training and inference.

Data Augmentation. For data preprocessing and augmentation, we follow [34] with some exceptions to account for the differences between general object detection and relationship detection data: For object detection datasets, we apply

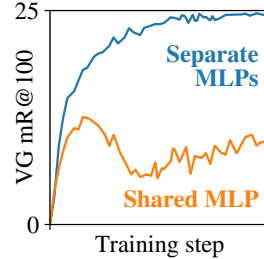


Figure 6: Computing `<subject>` and `<object>` embeddings with separate MLPs is necessary for good performance.

⁵ https://github.com/google-research/scenic/tree/main/scenic/projects/owl_vit

random left/right flip and random crop augmentation, up to 3×3 mosaics, and random negative labels. For relationship detection datasets, we replace the random crop with random resizing between $0.5\times$ and $1.0\times$ of the original size, since cropping may cause label inaccuracies if one member of a relationship is cropped off. For GQA200, we also remove random left/right flipping since the dataset contains spatial relationship annotations in the form of “<subject> to the left of <object>”. We do not use random prompt templates such as "a photo of a {}" or prompt ensembling for any datasets.

Data Rebalancing. For most of our experiments, we use the training data as-is without special treatment of the skewed object or predicate class distributions. Only for the model on the last line in Table 2, we perform simple rebalancing as follows, to show the potential of combining our method with orthogonal approaches focusing on data rebalancing: We first count the number of occurrences of each predicate in the training set to obtain its frequency. Then, during training, we randomly drop relationship annotations with a probability equal to its frequency (capped at 0.95 for the most frequent predicates).

Training Details. The B/32 and B/16 models are trained on images of size 768×768 at batch size 256 for 200'000 steps with the Adam optimizer [22] and a cosine learning rate schedule with a maximal learning rate of 5×10^{-5} and a 1000-step linear warmup. As in [34], the text encoder is trained with a learning rate of 2×10^{-6} instead. For the L/14 model, the image size is 840×840 , batch size is 128, and the maximal learning rate is 2×10^{-5} .

Speed Benchmarking. For the speed benchmarking in Figure 3, we assume a scenario in which a stream of images (e.g. a video feed) needs to be processed with a fixed set of 1000 text queries (i.e. 1000 object and predicate classes). We therefore report the time needed to process a new image, given pre-computed text query embeddings. We measure the time from calling the model with a single image (batch size 1) until the predictions are ready, using an NVIDIA V100 GPU. We measure 30 trials and report the median result.

A.2 Additional Experimental Results

Zero-shot GQA. To assess zero-shot generalization to unseen classes, we report the performance on the least-frequent 1503 object and 211 predicate classes in GQA, i.e. those *not* included in GQA200 and therefore unseen during training (Table 7). Given the large vocabulary and the difficulty of zero-shot predicate classification, we evaluate the model without graph-constraint, allowing four predicate predictions per <subject-object> pair. Although the performance of our model in this scenario is nontrivial, it is significantly lower than on seen classes (Table 4). We therefore suggest using the least-frequent GQA annotations in this manner as a challenging benchmark for future work on zero-shot VRD.

Recall@K. We provide results using the Recall@K metric (i.e. pooling all classes before recall computation) in Table 8. Note that this metric weighs classes by their frequency and is therefore not suitable for assessing long-tail performance [6, 46].

Performance without Graph-Constraint. All results in the main paper for VG150 and GQA are computed with graph constraint. Table 9 provides these results *without* graph constraint.

Model	mR@50	mR@100
<i>Scene Graph Generation (unseen classes)</i>		
SG-ViT (CLIP: ViT-B/32)	1.5	2.2
SG-ViT (CLIP: ViT-B/16)	1.9	2.3
SG-ViT (CLIP: ViT-L/14)	2.2	2.8

Table 7: Performance on the GQA test set (unseen classes only). Evaluated *without* graph-constraint.

	Visual Genome 150			GQA200		
	R@20	R@50	R@100	R@20	R@50	R@100
SG-ViT (CLIP: ViT-B/32)	19.8	28.1	34.5	16.4	22.9	27.9
SG-ViT (CLIP: ViT-B/16)	20.2	28.8	35.4	16.6	23.4	28.9
SG-ViT (CLIP: ViT-L/14)	21.8	31.1	38.3	18.6	26.6	32.6

Table 8: Evaluation on Recall metrics. Evaluated without graph-constraint.

	Visual Genome 150		GQA200	
	mR@50	mR@100	mR@50	mR@100
SG-ViT (CLIP: ViT-B/32)	20.5	24.8	21.9	26.1
SG-ViT (CLIP: ViT-B/16)	21.4	26.6	23.2	27.4
SG-ViT (CLIP: ViT-L/14)	23.9	29.5	26.7	32.8

Table 9: Evaluation without graph-constraints.

A.3 Additional Qualitative Examples

Figure 7 shows additional qualitative examples of relationships predicted by SG-ViT on VG150 and illustrates some error modes. The bounding boxes for each node of the relationship edge accurately captures the extent of the object instances while in each case aligning the grammatical <subject> and <object> in the correct direction, denoted by the shaded boxes **lime** and **red** respectively. The only false positive in this set of images is in Figure 7c where the B/32 model scores the relationship “light of bike” with the subject box selecting the bright licence plate of the motorcycle instead of the brake light.

A more frequent error mode is the confusion of singular instances and groups of instances. In some cases this can be put down to ambiguity in language. For example in Figure 7d the subject text “fruit” could be referring to super-set category that includes both oranges and apples and also shares the same word for both singular and plural. In other cases we see that this confusion also appears in the human annotations, e.g. in Figure 7e and Figure 7f. Such inconsistency is likely common in the training set, which may impact the model’s ability to distinguish singular and plural.

False negatives are another error mode, in which the model predicts no boxes, or assigns very low confidence to predictions. Two such examples are shown in Figure 7g and Figure 7h where the relationship descriptions are “snow on mountain” and “elephant near giraffe” respectively. On these examples, the model predicts no relationships with a score above 0.001, which we use as a threshold for visualization for all examples shown here. In the first case, the snow is only recognized by L/14 model and for the latter case neither model recognizes the decorations on the cup-cakes as either an elephant or giraffe.

A.4 Additional Graph Visualization Examples

Figure 8 illustrates the ability of SG-ViT to generate entire scene graphs on novel images. Each image shows all relationships with a score above 0.06, using the object categories and predicates from the full Visual Genome dataset to query the model. Nodes on the left are drawn at the center of the corresponding bounding box. Relationship predicates are shown in the graph visualization on the right. These visualizations are intended for qualitative assessment. For downstream use of the scene graph, further post-processing, e.g. non-maximum suppression, may be applied.










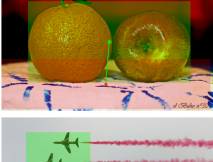

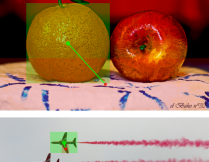
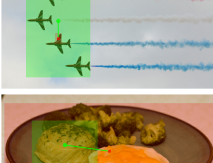
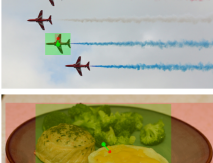
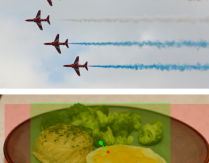






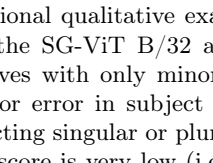
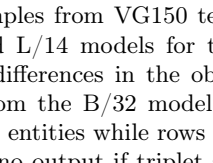
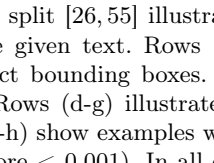
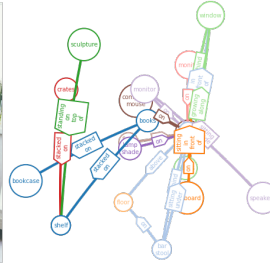
Relation	GT	SG-ViT (B/32)	SG-ViT (L/14)
(a) bottle on sink			
(b) number on post			
(c) light of bike			
(d) fruit on table			
(e) plane with wing			
(f) food on plate			
(g) snow on mountain			
(h) elephant near giraffe			

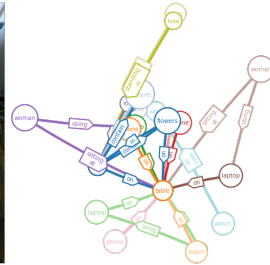
Figure 7: Additional qualitative examples from VG150 test split [26, 55] illustrating box outputs of the SG-ViT B/32 and L/14 models for the given text. Rows (a-b) show true positives with only minor differences in the object bounding boxes. Row (c) shows a minor error in subject from the B/32 model. Rows (d-g) illustrate the challenge of selecting singular or plural entities while rows (g-h) show examples where the relationship score is very low (i.e. no output if triplet score < 0.001). In all cases the <subject> is lime and the <object> is red.



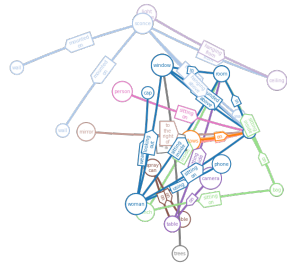
(a) Photo by Joanna Boj on Unsplash.



(b) Photo by Vadim Sherbakov on Unsplash.



(c) Photo by CoWomen on Unsplash.



(d) Photo by Nachele Nocom on Unsplash.

Figure 8: Additional graph visualizations on unseen images.