# Scene-Graph ViT: End-to-End Open-Vocabulary Visual Relationship Detection

Tim Salzmann[†1,2], Markus Ryll[2],
Alex Bewley[⋆1], and Matthias Minderer[⋆1]

[1] Google DeepMind
{tsal, bewley, mjlm}@google.com
[2] Technical University Munich
{tim.salzmann, markus.ryll}@tum.de

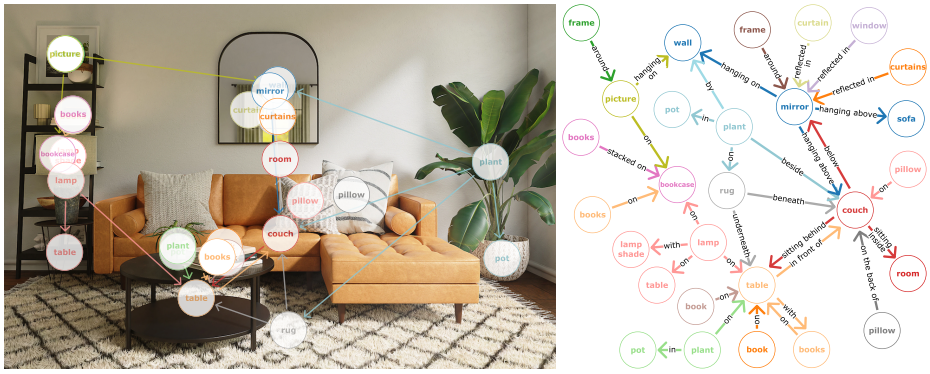



**Figure 1:** Relationships detected by our method on an unseen image. The top relationships by confidence score are shown. Photo by Spacejoy on Unsplash.

**Abstract.** Visual relationship detection aims to identify objects and their relationships in images. Prior methods approach this task by adding separate relationship modules or decoders to existing object detection architectures. This separation increases complexity and hinders end-to-end training, which limits performance. We propose a simple and highly efficient decoder-free architecture for open-vocabulary visual relationship detection. Our model consists of a Transformer-based image encoder that represents objects as tokens and models their relationships implicitly. To extract relationship information, we introduce an attention mechanism that selects object pairs likely to form a relationship. We provide a single-stage recipe to train this model on a mixture of object and relationship detection data. Our approach achieves state-of-the-art relationship detection performance on Visual Genome and on the large-vocabulary GQA benchmark at real-time inference speeds. We provide analyses of zero-shot performance, ablations, and real-world qualitative examples.

[†]Work done while at Google DeepMind.
[⋆]Advising project leads in alphabetical order.

## 1 Introduction

A fundamental goal of computer vision is to decompose visual scenes into structured semantic representations. A commonly studied task towards this goal is object detection, in which objects in an image are localized by bounding boxes and classified into semantic categories. However, a scene description also includes the semantic *relationships* between objects. This gives rise to the task of visual relationship detection (VRD, [4, 6, 32, 48, 49, 55]). In VRD, the model detects objects and infers pairwise relationships between them in the form of `<subject-predicate-object>` triplets[3] [32].

Detecting both objects and their relationships allows the construction of a *scene graph* [19, 30, 48] in which objects are represented as nodes and their relationships as edges. Scene graph generation (SGG) has wide-ranging applications in robotics [1, 5, 11, 17, 38, 42] and image retrieval [19, 27]. Increasingly, structured scene representations are also employed to provide grounding and explainability to multimodal large language models [15, 36, 59].

Prior work typically draws a distinction between object detection and relationship prediction. Detection is performed either as a wholly separate step before relationship prediction [8, 32, 48, 52, 55, 57], or by separate model parts such as "relationship decoders" that are responsible for modeling the interactions between objects [7, 49, 61]. This distinction makes it hard to optimize such models end-to-end for VRD [61]. In contrast, we propose an encoder-only architecture that models objects and relationships jointly, directly in the image encoder. Our architecture performs open-vocabulary relationship detection and can be trained end-to-end on arbitrary mixtures of object detection and relationship annotations.

Our model builds on a Transformer-based encoder-only object detector [34] in which the output tokens of the image encoder directly represent object proposals. From these tokens, class embeddings and bounding boxes are decoded with lightweight heads. Our insight is that this architecture is perfectly set up to learn relationships between objects directly in the image encoder, without the need for additional relationship-specific stages. This is because the existing self-attention in the encoder already models all-to-all pairwise interactions between the object proposal tokens.

To access information about the relationship between two of these tokens, we combine the embeddings corresponding to the `<subject>` and `<object>` token using a new *Relationship Attention* layer. Obtaining relationship embeddings for all possible pairwise combinations of object proposal tokens would be computationally infeasible. To reduce the number of combinations, we introduce a self-supervised hard attention mechanism that selects the highest-confidence `<subject-object>` pairs at a computational cost comparable to a single self-attention layer. We show how to directly supervise the attention scores of this mechanism without the need to propagate gradients through the hard selection.

[3] We use `<fixed width font>` to distinguish the grammatical terms `<subject>` and `<object>` from generic use of the word *object* as in *object detection*.

An additional benefit of our design is that it disentangles object names from relationship predicates during inference. In contrast to prior open-vocabulary methods [53, 54, 61], our model can embed object and predicate texts separately and generate confidence scores efficiently for all possible `<subject-predicate-object>` combinations.

In summary, we make the following contributions:

1. An efficient architecture for open-vocabulary visual relationship detection.
2. A single-stage recipe for joint object and relationship detection training.
3. Efficient, disentangled object and relationship inference.
4. Analysis of inference speed, ablations, and qualitative examples.

Our method achieves state-of-the-art visual relationship detection performance on the Visual Genome dataset (29.5% mR@100) and on the large-vocabulary GQA benchmark in open-vocabulary and zero-shot settings. It provides strong results while being significantly simpler than prior approaches to visual relationship detection.

## 2 Related Work

Below, we review the main approaches to VRD that are relevant to our work. For a comprehensive overview of the field, we refer to [27].

**Detector-Agnostic Relationship Detection.** Historically, many works on VRD assume that object detections are given, such that the task reduces to inferring relationships between the objects. These "detector-agnostic" methods use off-the-shelf detectors such as Faster-RCNN [39] to obtain boxes and box embeddings and infer a scene graph from them. Early work on large-scale VRD employed word embeddings to improve relationship generalization [32] and graph inference methods such as message passing for SGG [48]. More recent methods leverage relationship co-occurrence statistics [55] and address the long-tailed nature of the training data [8, 52, 57].

**End-to-End Relationship Detection.** A fundamental limitation of the detector-agnostic approach is that object detection and relationship prediction are treated separately and cannot learn from each other. Overcoming this limitation has recently motivated the development of end-to-end architectures in which objects and relationships are predicted jointly by a single model [7, 20, 29, 61]. All of these models have in common that they use a Transformer [47] decoder to predict object and relationship embeddings. The decoder represents object proposals with *query embeddings* that can cross-attend into image embeddings. The literature on object detection suggests that this choice may be problematic, because the query embeddings can be difficult to initialize, which can lead to unstable and slow optimization [3, 34, 43, 50, 58, 63]. To avoid this issue, we design an encoder-only architecture in which both object and relationship embeddings are computed directly from the output tokens of a Vision Transformer [10] image encoder, building on an idea from object detection [34]. In contrast to prior models [61], this allows us to train the model end-to-end on object and relationship annotations simultaneously.

**Long-Tailed and Open-Vocabulary Relationship Detection.** Object classes in the natural world follow a long-tailed distribution [12,62], and this is compounded in VRD due to the combination of subject, object, and predicate in a relationship [27,48,49]. A large number of VRD methods specifically aim to address this issue, for example with debiasing losses [52], data augmentation [57], or data resampling [8]. Additionally, while VRD models historically assumed a fixed set of object and predicate classes, recent works add open-vocabulary capabilities [13,41,54,61]. These models leverage pretrained vision-language models such as CLIP [18, 37, 51, 56] to inherit their natural-language understanding. Instead of learning a fixed set of classes, open-vocabulary models predict class embeddings that can be compared to the text embeddings of arbitrary object or predicate descriptions for classification. Here, we employ a pretrained vision-language backbone and combine it with a carefully designed, lightweight relationship detection head to preserve and transfer semantic knowledge from the backbone pretraining to VRD. Without adding further complexity to handle rare classes, this allows our model to achieve strong results on the large-vocabulary QGA benchmark [16] (Section 4.3).

While we focus on architectural improvements, another line of work addresses the scarcity of training data for VRD. For example, RLIP [53] and RLIPv2 [54], which focus on human-object interaction [4], propose relationship-focused image-text pretraining and self-training, which yield large improvements in VRD performance. These works are orthogonal to our contributions and can be combined.

## 3 Scene-Graph ViT

We propose an architecture for open-vocabulary relationship prediction in which both objects and the relationships between them are handled as "first-class citizens" directly in a single-stage process within the model backbone. We build on an encoder-only architecture for object detection [34] and adapt it to relationship prediction by adding a specialized attention layer. This layer exploits the pairwise structure between existing object embeddings to obtain relationship embeddings as schematized in Figure 2 and described below.

### 3.1 Encoder-Only Open-Vocabulary Object Detection

We briefly review the detection architecture that forms the basis of our model [34]. The model consists of Transformer-based [10] image and text encoders that are contrastively pretrained on a large number of image and text pairs [37]. The image encoder is adapted to detection by removing the final pooling layer and adding heads that predict bounding boxes and class embeddings directly from the output tokens of the image encoder. For open-vocabulary object classification, the embeddings computed by the class prediction head are compared to text encoder embeddings of object descriptions. This architecture achieves strong open-vocabulary object detection performance [33, 34] and does not suffer from the training instabilities observed in some encoder-decoder detection models [3, 34, 43, 50, 58, 63].

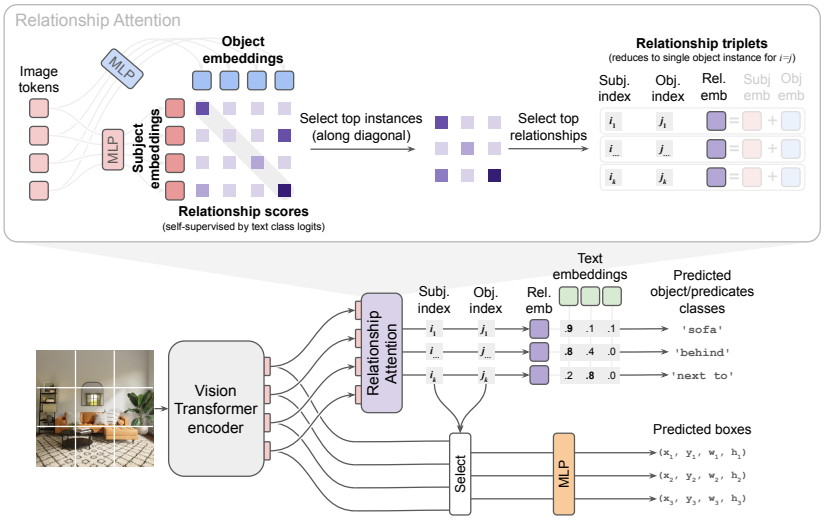



**Figure 2:** For relationship selection, image tokens are first projected using two lightweight MLPs to produce `<subject>` and `<object>` embeddings. A relationship score is then computed as the inner product between all `<subject>` and `<object>` embeddings. Relationships are filtered by first selecting the top object instances, using the scores along the diagonal to represent instance likelihood. Among the remaining instances, the top `<subject-object>` pairs are selected using the off-diagonal scores. This yields a set of relationship triplets, each consisting of a `<subject>` index, an `<object>` index, and a relationship embedding that is computed by summing the respective `<subject>` and `<object>` embeddings. For classification, the relationship embeddings are compared against text embeddings of object class or predicate text descriptions.

### 3.2 Extension to Relationship Prediction

In the architecture described above, each image encoder token represents an object proposal that captures per-object information. Importantly, the encoder consists of self-attention layers which, by design, model all pairwise interactions between these tokens. We therefore hypothesize that information about object relationships can be learned directly in the image encoder. To extract this information in the form of `<subject-predicate-object>` triplets, we introduce two MLPs that transform vision encoder output tokens into `<subject>` embeddings $\mathbf{s}_i$ and `<object>` embeddings $\mathbf{o}_j$ (Figure 2). Using different MLPs for `<subject>` and `<object>` is crucial to break symmetry in subsequent processing.

An embedding representing the relationship between two object proposals (encoder tokens) can then be obtained simply by element-wise addition of their `<subject>` and `<object>` embeddings. Obtaining relationship embeddings for all $N^2$ `<subject-object>` pairs, where $N$ is the number of object proposals, would be computationally infeasible. We therefore introduce a *Relationship Attention* layer that performs hard attention to select the `<subject-object>` pairs most likely to form a relationship. This layer computes an attention-like score $p_{ij} =$

$\mathbf{s}_i \mathbf{o}_j^T$, where $\mathbf{s}_i$ and $\mathbf{o}_j$ are the embedding vectors of `<subject>` $i$ and `<object>` $j$, and $p_{ij}$ represents the likelihood that a relationship between `<subject>` $i$ and `<object>` $j$ exists. By computing this score for all `<subject-object>` pairs, we obtain an $N \times N$ matrix, from which we select the top $k$ pairs.[4] For these $k$ pairs, we compute relationship embeddings $\mathbf{r}_{ij} = \text{LayerNorm}(\mathbf{s}_i + \mathbf{o}_j)$. To classify the relationship of a pair, its relationship embedding is compared to text embeddings of relationship predicates. We use the embeddings where `<subject>` and `<object>` are identical ($\mathbf{r}_{i=j}$) to represent object instances, and predict object classes from them. Boxes are predicted from the corresponding image encoder output tokens (see Figure 2).

The Relationship Attention layer therefore identifies object pairs to focus on for relationship classification, by computing a hard attention with `<subject>` embeddings as queries and `<object>` embeddings as keys. Since the hard selection is not differentiable, gradients from the relationship prediction cannot be used directly to train this layer. Instead, the relationship score is self-supervised to predict the maximum `<predicate>` class probability that will later be predicted for the relationship at the classification stage (Section 3.3).

### 3.3 Training

The image and text encoders of our model are initialized from a vision-language model that was contrastively pretrained on a large number of image-text pairs [37]. After adding the Relationship Attention as well as the object bounding box and class prediction heads, the model is jointly trained on a mixture of object and relationship detection datasets in a single training stage (Section 4.1), using the losses described below.

**Bipartite Matching.** Bipartite matching between object predictions ($\mathbf{r}_{i=j}$) and ground-truth annotations is performed based on a cost consisting of the object classification loss and the box prediction losses as in [3]. This matching between objects also establishes a matching of predicted to ground-truth relationship predicates, since a relationship is uniquely identified by the `<subject>` and `<object>` indices $i$ and $j$. Unmatched predictions are trained to predict low scores for all classes and incur no box prediction loss.

**Box Prediction Losses.** For bounding box regression, we use the L1 and generalized intersection-over-union (gIoU) losses described in [3].

**Object and Predicate Classification Loss.** For both object category and predicate classification, we use a sigmoid cross-entropy loss as in [34]. This loss is computed between the ground truth classes and logits obtained by computing the inner product of the relationship embeddings $\mathbf{r}_{ij}$ selected by the Relationship Attention layer with the text embeddings of the class names. For embeddings corresponding to individual objects ($\mathbf{r}_{i=j}$), the class name is the object category or description. For embeddings corresponding to relationships ($\mathbf{r}_{i \neq j}$), we use the predicate text as class name. This differs from prior work [53, 54, 61], which

[4] This selection reduces the number of relationships that need to be processed e.g. from $N^2 = 3600^2$ to $k = 16386$ (99.9% reduction) for our ViT-L/14 model.

uses the full `<subject-predicate-object>` triplet description. Our approach has the advantage of disentangling object category and predicate names, which allows for more efficient inference since object category and predicate texts are embedded separately.

**Relationship Score Loss.** To select only the most promising embeddings for further processing, the Relationship Attention layer predicts a score $p_{ij}$ that represents the likelihood that the corresponding `<subject-object>` pair forms a relationship (if $i \neq j$), or that the corresponding object exists in the image (if $i = j$). This score is trained with a sigmoid cross-entropy loss against targets provided by the model itself, namely the maximum probability predicted for any class for the corresponding $\mathbf{r}_{ij}$ embedding [33]. In this way, the relationship score is trained to predict the class probability of a potential embedding $\mathbf{r}_{ij}$ *before* actually computing and classifying the embedding. Note that this loss can only be computed for those objects and relationships that ultimately get selected for further processing. We found this to provide sufficient supervision.

The final loss is an equally weighted sum of four components: (1) classification loss, (2) L1 box loss, (3) gIoU box loss, (4) relationship score loss.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We use the following datasets for training or evaluation:

*Visual Genome* [26], is the largest VRD dataset, labeled with 2.3M triplet relationships across 108K images. However, as Visual Genome includes noisy annotations, the community commonly evaluates VRD on a cleaned version of the dataset [55] where the label space is reduced to the 150 most frequent object classes and the 50 most frequent predicate classes. This dataset is commonly referred to as Visual Genome 150 (VG150).

*GQA* [16] uses the same image corpus as Visual Genome, but is more diversely labeled, with 1703 object classes (1704 including a background class) and 310 predicates. Spatial relationships like "to the left of" are automatically labeled based on the 2-D spatial arrangement of bounding boxes. Similar to VQ150, GQA200 is a reduced and cleaned version of GQA with 200 object classes and 100 predicate classes [9].

*HICO-DET* [4] is a dataset which focuses on a more specialized subset of visual relationships, specifically interactions between humans and objects. HICO-DET contains 50k images and is exhaustively annotated for 600 defined human-object interaction (HOIs).

*Objects365* [40] (O365) is a large scale object-detection dataset with 365 object categories across two million images.

*Open X-Embodiment* [35] (OXE) dataset is targeted towards learning vision language action policies for robotics. OXE lacks bounding box annotations which precludes quantitative evaluation, but it represents both a visually different setting compared to the standard VRD benchmarks and a promising application area, so we use it for qualitative evaluation.

| | VG150 | VG | HICO | GQA200 | GQA | Objects365 |
|---|---|---|---|---|---|---|
| % in training mixture | 12.5% | 12.5% | 12.5% | 12.5% | 0% | 50% |
| # removed | 6587 | 17727 | 729 | 15361 | 18801 | 15306 |

**Table 1: Training data mixture.** *# removed* indicates the number of images that were removed from the official training split due to overlap with data we evaluate on.

**Training.** Unless otherwise noted, we train models on a mixture of VG, VG150, GQA200, HICO, and O365 in the proportions show in Table 1. The B/32 model is trained for 200'000 steps at batch size 256 (further details in Appendix A.1). Since the datasets we use share similar image sources, there is some overlap between the official training and evaluation splits. To ensure that no evaluation images are used for training, we rigorously filter all training datasets to remove images present in any of the test splits for all datasets we evaluate on, i.e. VG150, GQA, GQA200, and HICO. We use an image similarity filter that also identifies non-exact matches [24]. Table 1 shows the number of images removed from each training split. Note that all prior work performs similar deduplication.

**Evaluation Procedure and Metrics.** Prior work presents a diverse range of evaluation metrics often tailored to specific datasets. For non-exhaustively labelled datasets, like Visual Genome and GQA, using precision-based metrics is inherently inconclusive. Thus, the community has adapted a *Recall@K* metric for these datasets, where $K$ denotes a fixed budget of `<subject-predicate-object>` triplets on which the recall is computed. However, this metric is biased towards the more frequent `<predicate>` classes in the dataset [6, 46]. Therefore, recent approaches use *mean Recall@K*, which calculates *Recall@K* separately for each `<predicate>` class in the test data and then averages the results.

For exhaustively labeled datasets like HICO, it is feasible to use precision-based metrics such as *mean Average Precision* (mAP), a well-established metric in object detection. For HICO, the mean is taken over the 600 possible HOI triplets. Besides the overall metric, we also report results separately for rare ($< 10$ occurrences) and non-rare ($\geq 10$ occurrences) HOIs as mAPr and mAPn.

Further, evaluation can be "graph-constrained", where only a single prediction can be made per detected object pair, or "graph-unconstrained", where any number of predictions can be made per pair. We report unconstrained results in the main paper and results with graph constraint in Appendix A.2.

Multiple implementations of the aforementioned metrics exist, often showing differences in results. To compare fairly and directly against a wide range of prior approaches we identified the PyTorch evaluation procedure of [45] as the most prevalent routine for recall-based metrics and replicated their procedure in JAX with numerical accuracy. Similarly, for HICO, we numerically reproduce the original Matlab evaluation procedure outlined in [4] in JAX.

**Exhaustive Relationship Evaluation.** Previous methods [53, 54, 61] entangle objects and relationships in a single representation. Consequently, these methods score their embeddings against the full `<subject-predicate-object>` triplet. While this is feasible for datasets with a low number of possible triplets, like

| Model | Backbone | mR@50 | mR@100 |
|---|---|---|---|
| RelTR [7] | ResNet-50 | 6.8 | 10.8 |
| Transformer + CFA [28] | ResNeXt-101 | 12.3 | 14.6 |
| VCTree [46] + IETrans + Rwt [57] | ResNeXt-101 | 12.0 | 14.9 |
| Motif [55] + IETrans + Rwt [57] | ResNeXt-101 | 15.5 | 18.0 |
| GPS-Net [31] + IETrans + Rwt [57] | ResNeXt-101 | 16.2 | 18.8 |
| SG-Transformer [52] + IETrans + Rwt [57] | ResNeXt-101 | 16.2 | 18.8 |
| Sgtr [29] | ResNet-101 | 15.8 | 20.1 |
| DT2-ACBS [8] | ResNet-101 | 22.0 | 24.4 |
| UniVRD | CLIP: ViT-B/32 | 9.6 | 12.1 |
| UniVRD | CLIP: ViT-B/16 | 10.9 | 13.2 |
| UniVRD | CLIP: ViT-L/14 | 12.6 | 14.5 |
| SG-ViT | CLIP: ViT-B/32 | 20.5 | 24.8 |
| SG-ViT | CLIP: ViT-B/16 | 21.4 | 26.6 |
| SG-ViT | CLIP: ViT-L/14 | **23.9** | **29.5** |

**Table 2: Performance on the Visual Genome test set.** Best results are highlighted in bold.

HICO with 600 defined relationships, it quickly becomes computationally intractable for datasets with a larger vocabulary of objects and predicates (e.g. VG, GQA). Prior works [61] therefore evaluate only on triplets which are known to be present in the test split, which may inflate metrics. In contrast, our method disentangles objects and predicates (Section 3). This allows for an exhaustive evaluation that considers all object and predicate combinations and enables applications where knowledge of possible relationships is unavailable.

### 4.2 Relationship Detection Performance

We use Visual Genome as our main benchmark for relationship detection because it is widely adopted in the literature. Table 2 shows results for our models and prior work. Our method significantly improves over the prior best method DT2-ACBS [8] by 5.1 points mR@100 (29.5 vs. 24.4). We also note the large difference in performance to UniVRD [61], which builds on the same detection architecture as our method, but adds a Transformer decoder-based relationship module. To disentangle whether the improvements over UniVRD are due to differences in architecture or training data, we also trained our B/32 model on the UniVRD data mixture, and still observe a large improvement (23.9% VG mR@100 for our method vs. 12.1% for UniVRD when trained on the same data; Table 6). This result suggests that our encoder-only architecture is better suited for VRD than prior decoder-based architectures [7, 20, 29, 61].

### 4.3 Scaling to Large Vocabularies

Improving long-tail performance has been critical to the VRD field since the natural distribution of relationship triplets is highly skewed [27]. A substantial body of literature is devoted to this goal, often proposing complex loss debiasing or data augmentation approaches [8, 27, 52, 57]. It is therefore of interest whether our method, which includes no special treatment of rare classes, works well in this regime.

| Model | mR@50 | mR@100 |
|---|---|---|
| LLM4SGG [21] | 5.3 | 6.5 |
| Neural Motif w/ GCL [9] | 16.8 | 18.8 |
| VCTree w/ GCL [9] | 15.6 | 17.8 |
| SG-Transformer w/ CFA [28] | 13.4 | 15.3 |
| SHA w/ GCL [9] | 17.8 | 20.1 |
| SG-ViT (CLIP: ViT-B/32) | 21.9 | 26.1 |
| SG-ViT (CLIP: ViT-B/16) | 23.2 | 27.4 |
| SG-ViT (CLIP: ViT-L/14) | **26.7** | **32.8** |

**Table 3: Performance on the GQA200 test set.** Best results are shown in bold.

| Model | mR@50 | mR@100 |
|---|---|---|
| *Scene Graph Classification* | | |
| IMP [25, 44, 48] | 0.5 | 0.7 |
| Neural Motif [25, 44, 55] | 0.8 | 1.2 |
| Unbiased TDE [25, 45] | – | 0.7 |
| RTN [25] | – | 1.4 |
| MP [23] | – | 2.8 |
| Transformer w/ EBM [44] | 1.3 | 1.8 |
| *Scene Graph Generation* | | |
| SG-ViT (CLIP: ViT-B/32) | 9.5 | 11.5 |
| SG-ViT (CLIP: ViT-B/16) | 10.1 | 11.7 |
| SG-ViT (CLIP: ViT-L/14) | **13.5** | **16.6** |
| *Scene Graph Generation (unseen classes)* | | |
| 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 4: Performance on the GQA test set.** Best results are shown in bold.

To evaluate the performance of our method on large object and predicate vocabularies with a long tail of rare classes, we use the Graph Question Answering (GQA) dataset [16]. This dataset builds on VG and expands the number of classes and annotations. A simplified version of GQA with 200 object and 100 predicate classes, called GQA200 [9], is part of our training mixture. Our method surpasses prior results on the GQA200 test split by a large margin (Table 3).

The full GQA dataset has an even larger vocabulary with 1703 object and 311 predicate classes. To our knowledge, no prior methods evaluate scene graph *generation* on the full GQA dataset. Some prior methods report results on scene graph *classification*, where ground-truth boxes are provided. As shown in Table 4, our model achieves higher performance on the much harder task of scene graph *generation* than prior methods on *classification*.

To assess zero-shot generalization to unseen classes, we separately 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 4, bottom). Although performance of our model is significantly lower than on seen classes, it is still higher than the performance of prior methods performing scene graph *classification* on all GQA classes. We suggest using the least-frequent GQA annotations in this manner as a challenging benchmark for future work on zero-shot VRD.

We believe that two factors contribute to the strong performance in the open vocabulary regime: First, the simple design of the method does not impede transfer of semantic knowledge from the pre-trained VLM backbone because it adds only lightweight heads. In particular, we use no decoder, which could be difficult to train and could lead to "forgetting" of pretrained representations. Instead, most of the relationship modeling happens in the backbone, which has had the benefit of large-scale vision-language pretraining. Second, the open-vocabulary design allows end-to-end training on a mix of datasets, which was not possible for all prior methods.

| Model | Backbone | mAP | mAPr | mAPn |
|---|---|---|---|---|
| HOTR [20] | ResNet50 | 25.10 | 20.33 | 25.86 |
| RLIP [53] | ResNet50 | 32.84 | 26.85 | 34.63 |
| RLIPv2 [54] | ResNet50 | 35.38 | 29.61 | 37.10 |
| RLIPv2 [54] | Swin-L | **45.09** | **43.23** | **45.64** |
| UniVRD [60] | CLIP: ViT-B/32 | 29.98 | 22.94 | 32.02 |
| UniVRD [60] | CLIP: ViT-B/16 | 29.98 | 22.94 | 32.02 |
| UniVRD [60] | CLIP: ViT-L/14 | 37.41 | 28.90 | 39.95 |
| SG-ViT | CLIP: ViT-B/32 | 28.86 | 23.72 | 30.39 |
| SG-ViT | CLIP: ViT-B/16 | 31.98 | 26.83 | 33.52 |
| SG-ViT | CLIP: ViT-L/14 | 38.11 | 33.71 | 39.42 |

**Table 5: Performance on the HICO test set.** Our method is on par with the most comparable prior work, UniVRD, for overall mAP. However, our method performs better on "rare" classes (mAPr), indicating better generalization to rarely seen concepts. For reference, we also show RLIPv2, the state-of-the-art method on HICO. RLIPv2 proposes a self-training approach that is orthogonal and compatible to our method.

### 4.4 Human-Object Interaction

We also evaluate our model on the specialized task of human-object interaction, using the HICO benchmark [4]. The performance of our model is on par with the most comparable prior method [61], but does not show similarly large improvements as for VG or GQA200 (Table 5). HICO has a much narrower and more specialized vocabulary than VG or GQA (HICO evaluates on 600 pre-specified triplets, whereas VG150 has $1.1 \times 10^6$ possible triplet combinations). Performance on this specialized task may benefit less from transfer of pretrained representations and may instead be limited by the amount of task-specific training data. This is supported by the fact that the recent state-of-the-art method for HICO pretrains on large amounts of person-focused pseudo-labels [54].

### 4.5 Ablations

In the following, we ablate training datasets and model components to study the interplay of data and architecture on the model performance.

**Inference Speed and Number of Predicted Relationships.** The primary hyperparameter of our method is the number of relationships $k$ that are selected by the Relationship Attention layer from all possible `<subject-object>` combinations for further processing. Figure 3 shows how VRD performance and inference speed depend on $k$. For training, we use $k = 2^{14}$, which results in 71% (B/32) to 81% (L/14) of the speed of a pure object detection model [34] of the same size. Without top-$k$ selection, the model would

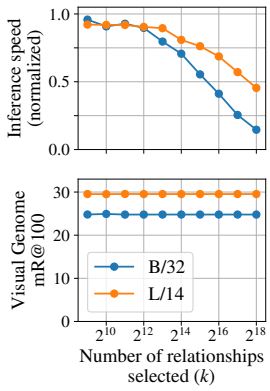



**Figure 3:** Model speed and VRD accuracy by number of selected relationships $k$. Speed is relative to a non-VRD object detector [34].

| | Mixture | | | | | | VG150 | GQA200 | HICO |
|---|---|---|---|---|---|---|---|---|---|
| | VG | VG150 | GQA200 | HICO | O365 | COCO | mR@100 | mR@100 | mAP |
| SG-ViT (ours) | 12.5% | 12.5% | 12.5% | 12.5% | 50% | – | 24.8 | 26.1 | 28.9 |
| UniVRD [61] | – | 10% | – | 20% | 50% | 20% | 23.9 ↓ 4% | 7.4 ↓ 72% | 26.6 ↓ 8% |
| w/o O365 | 25% | 25% | 25% | 25% | – | – | 21.9 ↓ 12% | 23.7 ↓ 9% | 23.8 ↓ 18% |
| w/o VG | – | 25% | 12.5% | 12.5% | 50% | – | 21.2 ↓ 15% | 21.7 ↓ 17% | 26.8 ↓ 7% |

**Table 6: Dataset Ablation.**

be several times slower (right end of top plot in Figure 3). Since the vast majority of `<subject-object>` pairs do not form relationships, VRD performance is essentially unaffected by top-$k$ selection at inference. For inference, we can therefore use $k = 2^{11}$ at no loss in accuracy to achieve 90% of the speed of the object detector (for B/32 and L/14). At this $k$, the B/32 model achieves 52.8 FPS at batch size 1 on an NVIDIA V100 GPU and is therefore suitable for real-time applications.

**Dataset Mixture.** To assess how training data and architectural improvements contribute to the performance of our model compared to prior work, we train the models on a range of dataset mixtures (Table 6). The full mixture, which we use for all experiments unless otherwise noted, includes VG, VG150, GQA200, HICO and O365. To allow direct comparison to UniVRD [61], we also train on their mixture (VG150, HICO, O365, COCO) and find that our model still performs better on VG150 by a large margin (23.9 mR@100 vs 12.1 for UniVRD for B/32 models). Since UniVRD uses the same image and text encoding backbones as our model, this result suggests that our encode-only model with Relationship Attention is an advance over decoder-based relationship architectures. We further find that including both pure object detection (O365) and large-vocabulary VRD data (VG) benefit all metrics.

**Object Detection Performance.** A necessary step towards good relationship detection performance is accurate object detection. In our architecture, object detection is treated as a special case of relationship detection in which the `<subject>` is identical to the `<object>`. We validate this design by training the model only on detection datasets without relationship annotations, using the same dataset mixture as the OWL-ViT [34] model (Objects365 and Visual Genome, [34]). We find that our model achieves a similar object detection performance as OWL-ViT (21.9% vs. 22.1% mAP and 18.3% vs. 18.9% mAPr on LVIS [12]), indicating that the Relationship Attention design does not interfere with object detection.

### 4.6 Qualitative Results

In Figure 4, we visualize qualitative examples to highlight difficult edge cases. Even where the model output differs from the ground truth, it is often plausibly correct, e.g. where either the extent of an object differs with the ground truth or


**Figure 4:** Qualitative examples of difficult edge-cases from VG150 test split [26, 55]. From left to right: Ground Truth, SG-ViT (B/32), SG-ViT (L/14). In all cases the `<subject>` is lime and the `<object>` is red.

there are multiple appropriate relations which are not annotated. For example, both models select the headlight on the bus in Figure 4b for "light on bus" as opposed to the sun reflected off the windshield. In Figure 4c both models focus more on the `<predicate>` "on" in "snow on hill" than the distinction between a mountain and a hill. Figure 4d shows shows a challenging scene where the B/32 model selects the bottle inside the open fridge while the L/14 correctly selects a water bottle in the door for "bottle in door" while also capturing the extent of the door instance.

Furthermore, to qualitatively assess the utility of this work beyond standard relation benchmark datasets, we also visualize some example relations applied to data from the robotics domain, namely the OXE dataset [35]. Figure 5 shows a real-life scene where multiple instances of the same semantic object categories are present within the scene. The model is able to match the `<predicate>` to the instances in the desired configuration. One subjective failure-case is that the "banana outside bowl" text query had its highest scoring triplet involving a bowl on a shelf in the background rather than the adjacent bowl. Another characteristic is that the model occasionally places boxes over groups with multiple instances of the same semantic class when a singular word is used for either the


**Figure 5:** Qualitative examples showing SG-ViT (L/14) on out-of-distribution data from the OXE dataset [35]. In all cases the `<subject>` is lime and the `<object>` is red. Note how the model correctly disambiguates several instances of the same class (e.g. "bananas" and "bottle") depending on their relationships.

`<object>` or `<subject>`. This can be attributed to the similarity between word and their plural in the CLIP embeddings and that many human annotations from VG also confuse singular and plural (see Appendix A.3).

## 5 Limitations

While our model performs strongly on large-vocabulary VRD, its performance on specialized human-object interaction detection is only on par with prior models. We discuss this limitation in Section 4.4. Further error modes are explored qualitatively in Section 4.6 and Appendix A.3.

A challenge for open-vocabulary relationship detection models as a whole is zero-shot generalization to unseen objects and predicates. While our model improves over prior approaches, there is still a large gap between the performance on seen and unseen classes (Tables 3 and 4). Future research should focus on closing this gap.

## 6 Conclusion

We present an architecture for open-vocabulary visual relationship detection. By combining an encoder-only design with a novel Relationship Attention layer for efficient selection of high-confidence relationships, our architecture leverages VLM pretraining and multi-dataset VRD training. It achieves strong performance on standard and large-vocabulary VRD benchmarks while maintaining pure object detection performance and adding little extra computational cost. Due to its simplicity and strong performance, we believe that our method will be a a useful basis for further research on visual relationship detection.

## References


1. Amiri, S., Chandan, K., Zhang, S.: Reasoning with scene graphs for robot planning under partial observability. IEEE Robotics and Automation Letters **7**(2), 5560–5567 (2022)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 213–229. Springer (2020)
4. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to Detect Human-Object Interactions. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (2018)
5. Chen, B., Xia, F., Ichter, B., Rao, K., Gopalakrishnan, K., Ryoo, M.S., Stone, A., Kappler, D.: Open-vocabulary queryable scene representations for real world planning. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 11509–11522. IEEE (2023)
6. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6163–6171 (2019)
7. Cong, Y., Yang, M.Y., Rosenhahn, B.: Reltr: Relation transformer for scene graph generation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
8. Desai, A., Wu, T.Y., Tripathi, S., Vasconcelos, N.: Learning of visual relations: The devil is in the tails. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15404–15413 (2021)
9. Dong, X., Gan, T., Song, X., Wu, J., Cheng, Y., Nie, L.: Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19427–19436 (2022)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR (2021)
11. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. arXiv preprint arXiv:2309.16650 (2023)
12. Gupta, A., Dollar, P., Girshick, R.: LVIS: A Dataset for Large Vocabulary Instance Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
13. He, T., Gao, L., Song, J., Li, Y.F.: Towards Open-Vocabulary Scene Graph Generation with Prompt-Based Finetuning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
14. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
15. Herzig, R., Mendelson, A., Karlinsky, L., Arbelle, A., Feris, R., Darrell, T., Globerson, A.: Incorporating Structured Representations into Pretrained Vision \& Language Models Using Scene Graphs. In: The 2023 Conference on Empirical Methods in Natural Language Processing (2023)

16. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6700–6709 (2019)
17. Hughes, N., Chang, Y., Carlone, L.: Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. Robotics: Science and Systems (RSS) (2022)
18. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
19. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image Retrieval Using Scene Graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
20. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: Hotr: End-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 74–83 (2021)
21. Kim, K., Yoon, K., Jeon, J., In, Y., Moon, J., Kim, D., Park, C.: LLM4SGG: Large Language Model for Weakly Supervised Scene Graph Generation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Knyazev, B., de Vries, H., Cangea, C., Taylor, G.W., Courville, A., Belilovsky, E.: Graph Density-Aware Losses for Novel Compositions in Scene Graph Generation. In: British Machine Vision Conference (BMVC) (2020)
24. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 491–507 (2020)
25. Koner, R., Shit, S., Tresp, V.: Relation transformer network. arXiv preprint arXiv:2004.06193 (2020)
26. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**, 32–73 (2017)
27. Li, H., Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Zhao, X., Shah, S.A.A., Bennamoun, M.: Scene Graph Generation: A comprehensive survey. Neurocomputing (2024). https://doi.org/https://doi.org/10.1016/j.neucom.2023.127052
28. Li, L., Chen, G., Xiao, J., Yang, Y., Wang, C., Chen, L.: Compositional feature augmentation for unbiased scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 21685–21695 (2023)
29. Li, R., Zhang, S., He, X.: Sgtr: End-to-end scene graph generation with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19486–19496 (2022)
30. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1261–1270 (2017)
31. Lin, X., Ding, C., Zeng, J., Tao, D.: Gps-net: Graph property sensing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3746–3753 (2020)

32. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 852–869. Springer (2016)
33. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. Advances in Neural Information Processing Systems **36** (2024)
34. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple open-vocabulary object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 728–755. Springer (2022)
35. Open X-Embodiment Collaboration: Open X-Embodiment: Robotic Learning Datasets and RT-X Models (2023)
36. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Ye, Q., Wei, F.: Grounding Multimodal Large Language Models to the World. In: International Conference on Learning Representations (2024)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
38. Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., Suenderhauf, N.: Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In: 7th Annual Conference on Robot Learning (2023)
39. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
40. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8430–8439 (2019)
41. Shi, H., Hayat, M., Cai, J.: Open-vocabulary object detection via scene graph discovery. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 4012–4021 (2023)
42. Singh, K.P., Salvador, J., Weihs, L., Kembhavi, A.: Scene graph contrastive learning for embodied navigation. In: PProceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10884–10894 (2023)
43. Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., Yang, M.H.: ViDT: An Efficient and Effective Fully Transformer-based Object Detector. In: International Conference on Learning Representations (2022)
44. Suhail, M., Mittal, A., Siddiquie, B., Broaddus, C., Eledath, J., Medioni, G., Sigal, L.: Energy-based learning for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13936–13945 (2021)
45. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3716–3725 (2020)
46. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6619–6628 (2019)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

48. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5419 (2017)
49. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 670–685 (2018)
50. Yao, Z., Ai, J., Li, B., Zhang, C.: Efficient DETR: Improving End-to-End Object Detector With Dense Prior. arXiv preprint arXiv:2104.01318 (2021)
51. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: CoCa: Contrastive Captioners are Image-Text Foundation Models. Transactions on Machine Learning Research (2022)
52. Yu, J., Chai, Y., Wang, Y., Hu, Y., Wu, Q.: CogTree: Cognition Tree Loss for Unbiased Scene Graph Generation. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-21. International Joint Conferences on Artificial Intelligence Organization (8 2021)
53. Yuan, H., Jiang, J., Albanie, S., Feng, T., Huang, Z., Ni, D., Tang, M.: Rlip: Relational language-image pre-training for human-object interaction detection. Advances in Neural Information Processing Systems **35**, 37416–37431 (2022)
54. Yuan, H., Zhang, S., Wang, X., Albanie, S., Pan, Y., Feng, T., Jiang, J., Ni, D., Zhang, Y., Zhao, D.: RLIPv2: Fast Scaling of Relational Language-Image Pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 21649–21661 (2023)
55. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5831–5840 (2018)
56. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18123–18133 (2022)
57. Zhang, A., Yao, Y., Chen, Q., Ji, W., Liu, Z., Sun, M., Chua, T.S.: Fine-grained scene graph generation with data transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 409–424. Springer (2022)
58. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
59. Zhang, Y., Ma, Z., Gao, X., Shakiah, S., Gao, Q., Chai, J.: GROUNDHOG: Grounding Large Language Models to Holistic Segmentation. In: Conference on Computer Vision and Pattern Recognition 2024 (2024), https://openreview.net/forum?id=PWVW3ux1Ju
60. Zhao, L., Yuan, L., Gong, B., Cui, Y., Schroff, F., Yang, M.H., Adam, H., Liu, T.: Unified Visual Relationship Detection with Vision and Language Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
61. Zhao, L., Yuan, L., Gong, B., Cui, Y., Schroff, F., Yang, M.H., Adam, H., Liu, T.: Unified Visual Relationship Detection with Vision and Language Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
62. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 350–368 (2022)

63. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: International Conference on Learning Representations (2021)

# A Appendix

## A.1 Model Details and Hyperparameters

Unless otherwise noted, we follow the public implementation[5] 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.

**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 $512 \times 512$. 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 random left/right flip and random crop augmentation, up to $3 \times 3$ 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.

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

**Training Details.** The B/32 and B/16 models are trained on images of size $768 \times 768$ 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 $5 \times 10^{-5}$ and a 1000-step linear warmup. As in [34], the text encoder is trained with a learning rate of $2 \times 10^{-6}$ instead. For the L/14 model, the image size is $840 \times 840$, batch size is 128, and the maximal learning rate is $2 \times 10^{-5}$.

**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

**Recall@K.** We provide results using the Recall@K metric (i.e. pooling all classes before recall computation) in Appendix A.2. Note that this metric weighs classes by their frequency and is therefore not suitable for assessing long-tail performance [6, 46].

**Graph-Constrained Performance.** All results in the main paper are computed without graph constraint [27]. Appendix A.2 provides the main results *with* 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 7: Evaluation on Recall metrics.**

| | Visual Genome 150 | | GQA200 | | HICO |
|---|---|---|---|---|---|
| | mR@50 | mR@100 | mR@50 | mR@100 | mAP |
| SG-ViT (CLIP: ViT-B/32) | 13.2 | 16.2 | 12.3 | 15.1 | 2.3 |
| SG-ViT (CLIP: ViT-B/16) | 14.3 | 17.0 | 14.2 | 16.8 | 2.4 |
| SG-ViT (CLIP: ViT-L/14) | 17.6 | 20.8 | 16.6 | 19.3 | 3.1 |

**Table 8: Evaluation with graph-constraints.**

### A.3 Additional Qualitative Examples

Figure 6 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 6c 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 6d 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 6e and Figure 6f. 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 6g and Figure 6h 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 7 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.

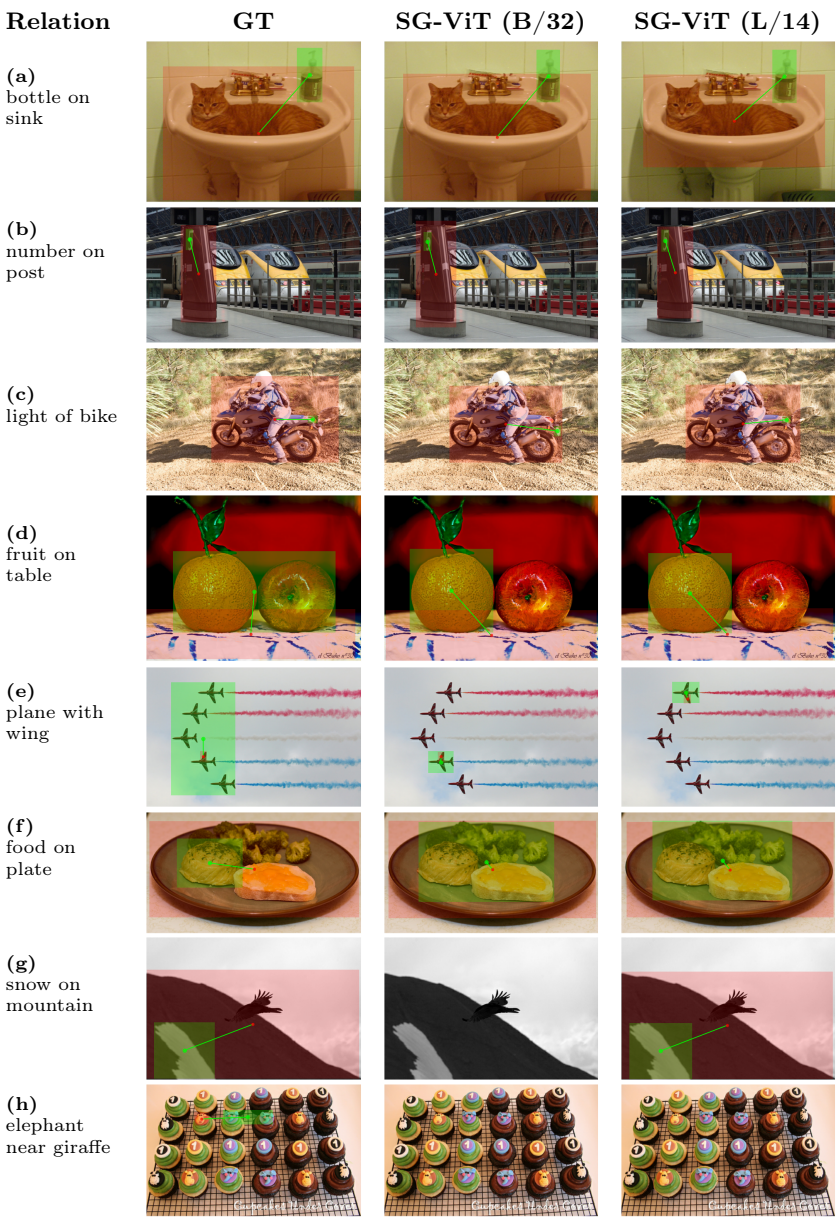
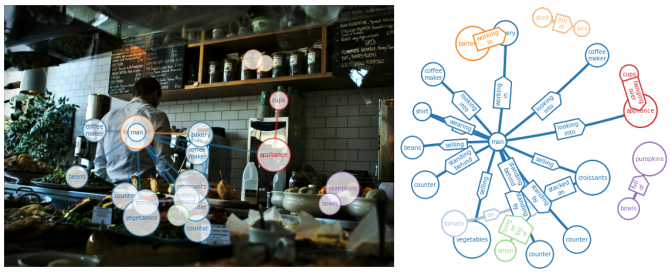
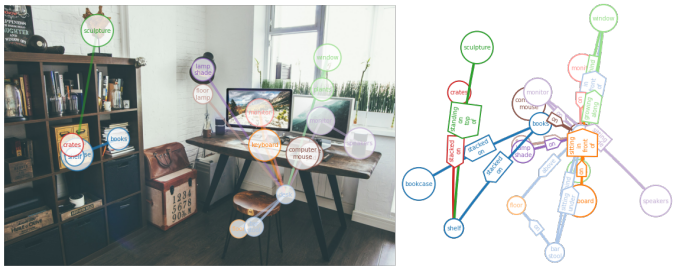


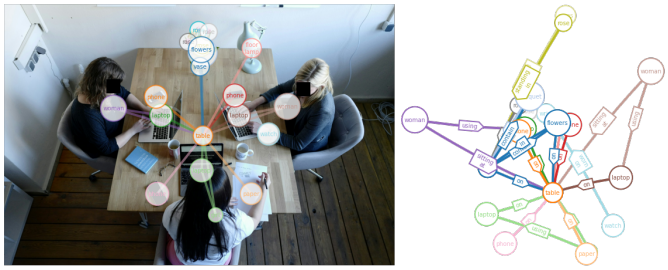

**Figure 6:** Additional qualitative examples from VG150 test split [26, 55] illustrating box outputs of the SG-ViT B/32 and L/14 models for the given text. Rows (a-b) show true positives with only minor differences in the object bounding boxes. Row (c) shows a minor error in subject from the B/32 model. 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.



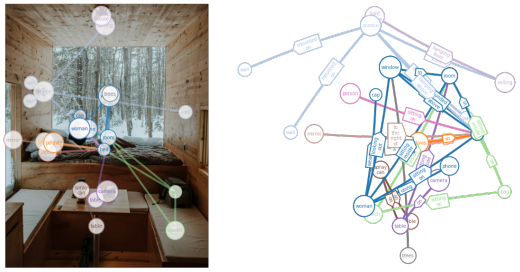
**(a)** Photo by Joanna Boj on Unsplash.



**(b)** Photo by Vadim Sherbakov on Unsplash.



**(c)** Photo by CoWomen on Unsplash.



**(d)** Photo by Nachelle Nocom on Unsplash.

**Figure 7:** Additional graph visualizations on unseen images.