Attention-Translation-Relation Network for Scalable Scene Graph Generation

Nikolaos Gkanatsios^{1,2}, Vassilis Pitsikalis^{1}, Petros Koutras^{2}, Petros Maragos^{2}
^{1}Deeplab, ^{2}National Technical University of Athens *
nikos.gkanatsios93@gmail.com, vpitsik@deeplab.ai, {pkoutras, maragos}@cs.ntua.gr

Abstract

We find that most Scene Graph Generation approaches suffer from two limitations as they: 1) use generic attention mechanisms and dataset-specific statistics that supersede visual features and 2) treat "no interaction" as an extra, both noisy and dominant, class and prune graph edges manually or applying simple filters. As a result, such approaches do not scale up on different settings and specifications. We propose a three-stage pipeline that employs multi-head attention driven by language and spatial features, Translation Embeddings (TransE) and multi-tasking to detect an interacting pair of objects. Our attentional scheme is able to maximize the visual features' interpretability, as well as to capture the nature of datasets of different scales, while multi-tasking robustly resolves the bias of the background class. We present an experimental overview of the related literature, unveil a multitude of evaluation inconsistencies and provide quantitative and qualitative support with experiments on a variety of datasets, where our approach performs on par or even outperforms current state-of-the-art.

1. Introduction

"A picture is worth a thousand words", but how can we read them? Parsing an image to create a structured representation, namely Scene Graph Generation, has recently captured the interest of researchers, aiming on bridging the gap between visual and semantic perception and expanding the range and capabilities of Computer Vision applications [24, 62, 54, 59, 15]. The task involves detecting visual relationships [43], i.e. $< S, P, O >$ triplets, with the predicate P being the interaction of subject S and object O, and integrating them to a graph (Fig. 1).

Former approaches have shown that, due to the density of object annotations and the large intra-class variance of predicates, pure visual information is not sufficient for modeling relationships [43, 49]. Thus, they adopt attention mechanisms [74, 77] and language priors [36, 43, 77] or benefit from the datasets' statistics [12, 65, 66, 71]. Nonetheless, we find that such strategies impose biases on the networks' focus towards visually-irrelevant predictions: as we quantitatively and qualitatively show (Section 4.3 and Fig. 5), the network fails to disambiguate between multiple pairs interacting in the same way [71] due to misleading attention.

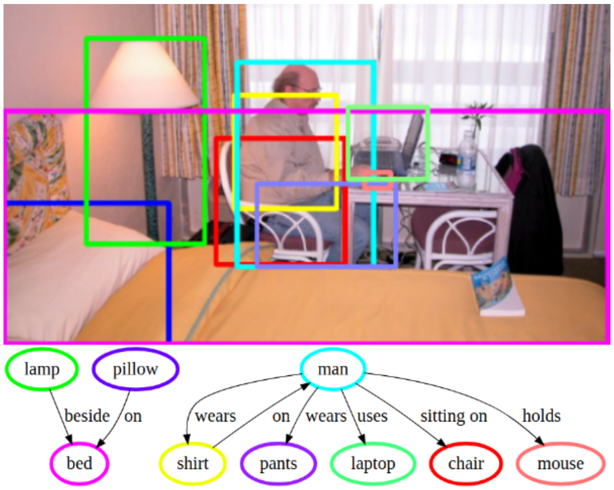
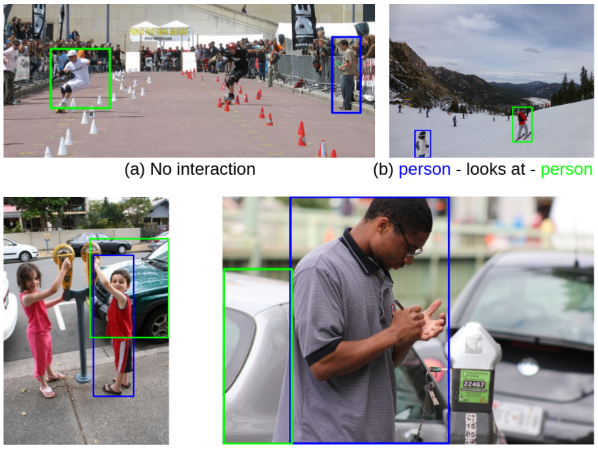


Figure 1: Scene Graph Generation refers to the formation of a visually-grounded graph structure, where related object nodes are connected via predicate edges. Despite the density of object annotations, the constructed graph is sparse.

Moreover, scene graphs tend to be sparse: only a few object pairs are related. Contemporary works often learn a spatial-semantic pair filtering network [12, 60] or consider a background class [58, 50] and sample not-interacting pairs [8, 65]. However, objects' labels and positions are not sufficient to determine their relevance without focus on specific visual cues (Fig. 2). Even worse, the two most popular datasets for visual relationships [43, 31] are sparsely and inconsistently annotated (Fig. 2): relevance is not a binary

*This project was conducted while N. Gkanatsios was an intern at the National Technical University of Athens and has been funded by deeplab.ai, as far as V. Pitsikalis and N. Gkanatsios are concerned. This is a part of the deeplab.ai research activities, such as student research-training funding, and collaborations with academic institutions.



(a) No interaction

(b) person - looks at - person

(c) No interaction

(d) person - in front of - car

Figure 2: (a)-(b): Spatial and semantic features alone are not sufficient for relationship detection. (c)-(d): The same relationship, "person - in front of - car" is annotated only in image (d). Employing visual features and decodings dataset's "preferences" is essential to edge pruning.

classification problem and an effective pair filter has to decode the annotators' "preferences" as well [42]. Yet, the unrelated pairs largely overpopulate the related ones and dominate the classification task, while sampling for negative pairs obscures the underlying distribution of relations.

Manually encoding statistics and pruning edges does not scale effectively across different datasets and challenges. Motivated by these observations, we propose a scene graph generation pipeline, where the classification of a detected pair of objects can be factorized in three steps: *Attention*: We learn a multi-head attentional scheme [55], one head per predicate class, that fuses linguistic and spatial information to mine and properly encode the most meaningful visual features. *Translation*: Predicate and subject/object features are projected to a common score space so that $P \approx O - S$ [67]. *Relation*: We perform the final classification, fusing the translated scores of the previous stage. We solve two separate tasks, one for predicate classification and one for objects' relevance, that share a similar architecture. In this way, pair filtering benefits from deep attention on visual features and is able to capture the dataset's structure without overshadowing the predicate classification task.

Our topology, the *Attention-Translation-Relation Network* (ATR-Net), is evaluated on the widely used VRD [43] and VG [31] datasets, using a variety of available splits for the latter [58, 35, 12, 67, 69], with different specifications and scales. We also shed light to several inconsistencies in the tasks' and metric's definition that make prior works incomparable and we refine literature results in order to fairly compare. Under multiple evaluation settings, ATR-Net is still compelling and outperforms most existing methods, demonstrating the efficacy of the proposed approach.

For the rest of this paper, the term "relevance" refers to the probability that two objects are related/interact. A "pair filter" is a network that outputs two objects' relevance.

2. Related Work

Visual Relationship Detection has lately emerged as a separate problem [43]. The majority of prior literature follows the two-step pipeline that first detects objects and then solves a classification task for the predicate of each object pair [11, 36, 43, 63, 70, 76, 75, 77]. Other works use a Region Proposal Network [51] to obtain candidate object regions and then jointly classify objects and predicates [12, 37, 69] or train end-to-end networks to directly predict relationships [30, 17, 67]. We follow the two-step detection method in order to disjoin the errors of object and relationship detection.

Scene Graph Generation [58] constrains visual relationship detection on a graph, opening new perspectives for the task. Works like [8, 39, 58, 60] perform message-passing between nodes and edges, while others employ context from neighboring instances in order to improve a single instance's recognition, using Recurrent Neural Networks [65], self-attention [50, 53, 57] or iterative reasoning with memory [9, 56]. Complementary to these approaches, our goal is to improve single-instance prediction.

Attention has already been applied on scene graph generation [8, 16, 19, 30, 50, 74, 73, 77] in order to deal with noisy and overlapping object regions (Fig. 1). Close to us in terms of features, [19, 74, 73] combine bottom-up and top-down attention [2] with spatial and linguistic features, yet they use single-head attention, irrespective of class. On the other hand, [77] employs a per-predicate-class attentional schema, but only language-based. As we experimentally show in Sec. 4.3, such mechanisms may in fact misguide the network. Lastly, [8, 50] employ multi-head and per-class self-attention to relate object nodes in a sparse scene graph. Combining the above, we implement per-predicate class attention conditioned on language and spatial features.

Visual Translation Embeddings (VTransE) [67] are a mapping of S, P and O in a vector space where valid relationships satisfy $S + P \approx O$. [67] uses subject and object features and aligns $O - S$ with the desired P inside the loss function. [50] employs predicate features as well and applies an L2 constraint to minimize $\|S + P - O\|_2$. [21] extends the formulation into Union Visual Translation Embeddings such that $P \approx U - S - O$, where U stands for the union of subject and object regions. Contrary to these approaches, we do not directly align P and $O - S$, instead, we employ constraints in a score space to force both P and $O - S$ to match with the ground-truth.

Relevance of objects in a scene graph plays a crucial role on detection performance. Traditional approaches use

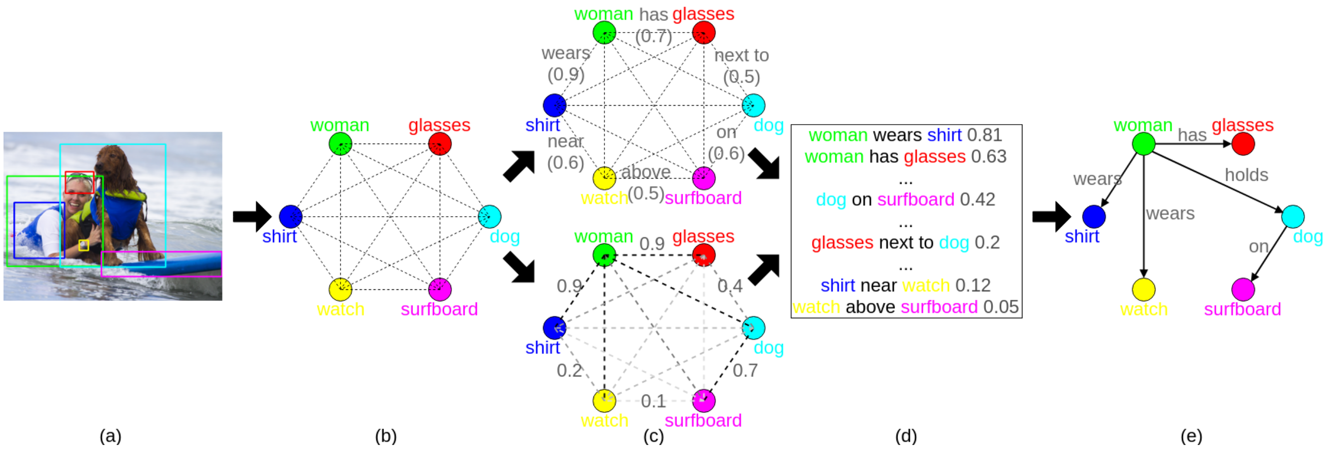


Figure 3: (a) Given an image, we first detect objects and construct a fully-connected graph (b). Then, ATR-Net solves two separate tasks to classify each edge and assign it with a relevance score that measures the probability the two objects interact (c). (d) The two scores per edge are multiplied and only the top-scoring edges are not pruned. (e) The output graph is sparse and each edge corresponds to a relationship.

ranking loss functions and filter the lowest-scoring relationships [43, 36, 71], while others train a pair filtering network based on semantic and/or spatial features [12, 39, 60, 74]. Graph-based works often consider an extra background class [58, 50] and obtain extra performance gain by manually filtering pairs with not intersectant boxes [8, 57, 65]. Our work lies close to those of [47, 66] that view relevance as an auxiliary task, yet, opposite to them, we multiply relevance and classification probabilities only during inference, avoiding the need for an extra class. Also close to us, [53] computes a pair "validity" score and applies tree-based algorithms to prune low-scoring edges. Instead, ATR-Net compares scores across the whole graph, rather than in tree neighborhoods.

3. Approach

We tackle Scene Graph Generation by detecting and ranking all $< S, P, O >$ triplets in an image and then integrating them into a sparsely-connected graph (Fig. 3). When inferring an image, we first detect objects using Faster-RCNN [51]. Considering that all objects could possibly interact, we form a fully-connected graph. Then, ATR-Net focuses on each edge to solve two tasks: assign a predicate probability $\mathrm{P}(P|\mathrm{related})$ assuming the two objects are related and predict a relevance score $\mathrm{P}(\mathrm{related})$ that measures the probability of interaction. The final predicate score is $\mathrm{P}(P) = \mathrm{P}(P|\mathrm{related})\mathrm{P}(\mathrm{related})$ and the edge's score is $\mathrm{P}(S, P, O) = \mathrm{P}(S)\mathrm{P}(P)\mathrm{P}(O)$, combining the predicate score and Faster-RCNN's scores for the subject and the object. We then rank the edges' scores and keep the top-N, where N is a threshold. We now describe how ATR-Net handles a single edge in three steps, in order to compute $\mathrm{P}(P)$ using multi-head attention and translation embeddings. The pipeline can be viewed in Fig. 4.

3.1. Attention

Given a detected pair, we extract convolutional features for both objects and the predicate region, i.e. the minimal closure of their bounding boxes. The objects' features condition the detection on the objects of interest, while predicate features view the objects in the context of the image and mine patterns of interaction or geometry. These types of features, without proper attention, fall for two shortcomings: objects' features tend to focus on class information and ignore relationship cues, e.g. in Fig. 2 (a) and (b), the persons' poses are more significant than classifying them as people; predicate features are noisy and often highly overlapping across different pairs of objects, e.g. in Fig. 1, the predicate box of the bed and the lamp includes all other objects.

Multi-head attention [55], one head for each predicate class, is employed to concentrate the network's "focus" on the discriminative visual cues that may only appear in a small fraction of the image. We constrain the attention function on the objects' classes and their spatial configuration. More specifically, we use word embeddings of the objects' categories [45], binary mask features [12] and box deltas [70]. We avoid manually precomputed probabilities [65] as they provoke dataset-specific biases. Instead, word embeddings capture both the class and the semantics of the objects, while spatial features are class-agnostic and able to generalize in unseen setups [49, 21]. Combining these two features, we have a strong discriminative attention that is able to capture the dataset's nature and scale up in open-world vocabularies [1].

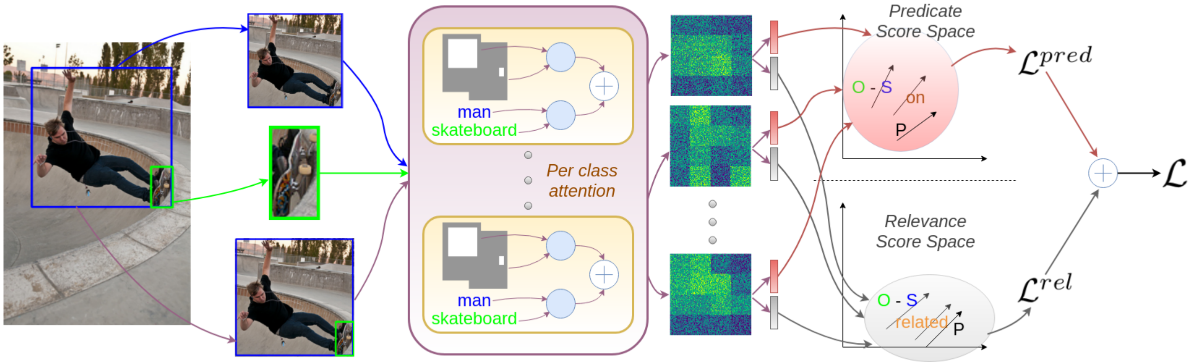


Figure 4: ATR-Net classifies and ranks separately each edge on a dense scene graph. Visual features from the subject, the object region and their union are projected in a score space as S, O and P, guided by multi-head language and spatial attention. A separate score space is created for predicate classification and object relevance, while loss constraints are imposed so that both P and $O - S$ match with the task's ground-truth. The total loss is the weighted sum of the two tasks' losses.

Language and spatial features are fused to compute a low-dimensional attention vector A_P for each predicate class P. We denote as A the attention matrix of all predicates, having as many rows as the number of classes. Conditioned on this attention, we perform attentional pooling [77] and compute weights $\mathbf{W}_S(A)$, $\mathbf{W}_P(A)$ and $\mathbf{W}_O(A)$ for the subject, predicate and object features respectively.

3.2. Translation

Having computed the attention weights, we represent S, P and O as projections $\mathbf{W}_S(A)x_S$, $\mathbf{W}_P(A)x_P$ and $\mathbf{W}_O(A)x_O$ of their visual features x_S, x_P, x_O into a score space. Thus, we reformulate VTransE equation into:

$$\mathbf{W}_P(A)x_P \approx \mathbf{W}_O(A)x_O - \mathbf{W}_S(A)x_S \qquad (1)$$

Instead of directly minimizing $\|P + S - O\|$, we impose loss constraints, $\mathcal{L}_P$ and $\mathcal{L}_{OS}$, so that both $\mathbf{W}_P(A)x_P$ and $\mathbf{W}_O(A)x_O - \mathbf{W}_S(A)x_S$ come closer to the ground-truth.

3.3. Relation

For each object pair we have to decide both about its relevance and its predicate. We view this procedure as two independent tasks and learn per-task attention weights, $\mathbf{W}^r(A) = (\mathbf{W}^r_S(A), \mathbf{W}^r_P(A), \mathbf{W}^r_O(A))$ for relevance and $\mathbf{W}^p(A) = (\mathbf{W}^p_S(A), \mathbf{W}^p_P(A), \mathbf{W}^p_O(A))$ for predicate classification. Consequently, two score spaces are created, one for each task, with the respective losses $\mathcal{L}^r_P$, $\mathcal{L}^r_{OS}$ and $\mathcal{L}^p_P$, $\mathcal{L}^p_{OS}$.

Scores per task are fused and a meta-classifier is trained to return the output scores. Let $\mathcal{L}^r_f$ and $\mathcal{L}^p_f$ be the fusion losses for the relevance and predicate task respectively. Then the total relevance loss obtains the form:

$$\mathcal{L}^r(\mathbf{W}^r) = \mathcal{L}^r_f(\mathbf{W}^r) + \mathcal{L}^r_P(\mathbf{W}^r_P) + \mathcal{L}^r_{OS}(\mathbf{W}^r_O, \mathbf{W}^r_S) \quad (2)$$

while the predicate classification loss

$$\mathcal{L}^p(\mathbf{W}^p) = \mathcal{L}^p_f(\mathbf{W}^p) + \mathcal{L}^p_P(\mathbf{W}^p_P) + \mathcal{L}^p_{OS}(\mathbf{W}^p_O, \mathbf{W}^p_S) \quad (3)$$

During training, we minimize the weighted sum of losses for both tasks:

$$\mathcal{L}(\mathbf{W}^r, \mathbf{W}^p) = \lambda_r \mathcal{L}^r(\mathbf{W}^r) + \lambda_p \mathcal{L}^p(\mathbf{W}^p) \qquad (4)$$

where λ hyperparameters balance each term's importance. The optimization problem is solved using Adam optimizer [28]. All losses are Cross-Entropy losses.

4. Evaluation and Results

4.1. Datasets, Metrics and Inconsistencies

The two most widely benchmarked datasets for Scene Graph Generation and Visual Relationship Detection are VRD [43] and Visual Genome (VG) [31]. However, we notice that current work in the field is often inconsistent in terms of dataset annotations, tasks evaluated and metrics used. We present here a literature taxonomy around the different aspects that make prior results incomparable and then we evaluate our method considering all different settings.

Datasets: VG does not offer a standard train/test split and is noisily and inconsistently annotated [58, 67, 35, 12], resulting in an immense range of classes with little variation and few examples. As an attempt to clean these annotations, prior works have explored semi-automatic ways (e.g. class merging and filtering) to construct their own VG versions. Of these, [58, 67, 35, 12] have released their cleansed annotations and are the most frequently used. Other works [33, 37, 48, 76, 64, 63, 10] use a paper-specific and non-publicly available split, disabling direct future comparisons with their experiments. Lastly, [69] presents experiments on a large-scale version of VG and [38] proposes a new split that has not been benchmarked yet.

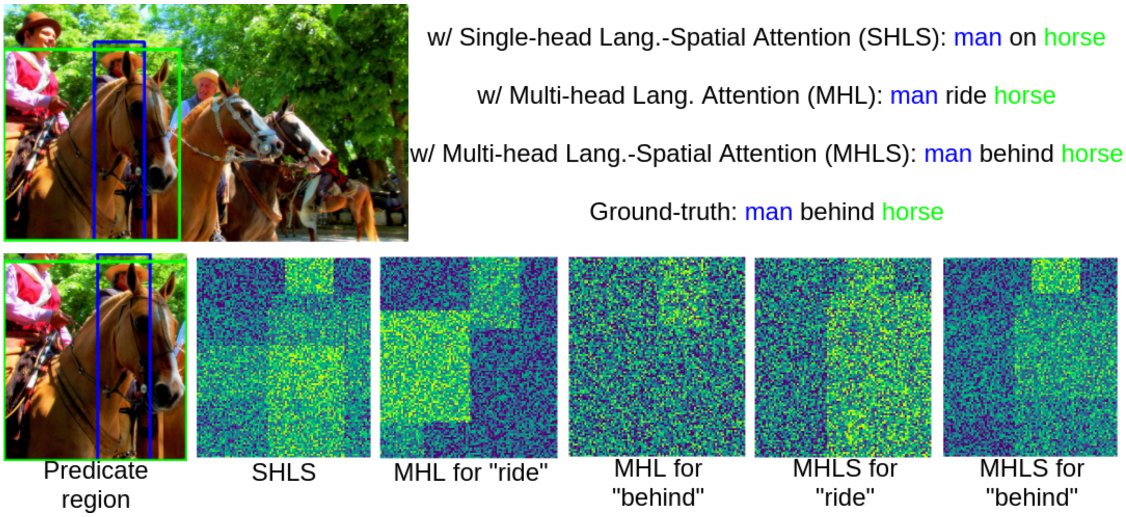

w/ Single-head Lang.-Spatial Attention (SHLS): man on horse
w/ Multi-head Lang. Attention (MHL): man ride horse
w/ Multi-head Lang.-Spatial Attention (MHLS): man behind horse
Ground-truth: man behind horse
Predicate region
SHLS
MHL for "ride"
MHL for "behind"
MHLS for "ride"
MHLS for "behind"

Figure 5: Visualization of attentional pooling weights for the same predicate region and for different mechanisms. To map weights back to the image we upsample and add random noise. Single-head attention captures generic concepts of interest irrespective of class. Multi-head Attention with language confuses the two men and gives high score to "ride". Spatial information helps to disambiguate between the two men and predict the correct predicate "behind".

Dataset	Train/test images	Pred. Classes	Obj. Classes
VRD [43]	4k/1k	70	100
VG-MSDN [35]	46.2k/10k	50	150
VG-VTE [67]	73.8k/25.8k	100	200
sVG [12]	64.7k/8.7k	24	399
VG200 [58]	75.6k/32.4k	50	150
VG80K [69]	99.9k/4.8k	29086	53304

Table 1: Statistic of different datasets we use for evaluation.

The effectiveness and scalability of the proposed method is validated on VG200 [58], VG-VTE [67], VG-MSDN [35], sVG [12], VRD [43] and VG80K [69]. The statistics of these datasets are summarized in Table 1, where it becomes clear that they have distinct annotations and sizes.

Tasks: There are totally 5 tasks evaluated in Scene Graph Generation, but prior works often evaluate on less. We preserve the tasks' names as defined in [43] and [58], despite inconsistencies on whether they are in fact classification or detection tasks:

Predicate Detection (PredDet) [43]: given objects' categories and boxes, as well as which pairs do interact, classify each pair's predicate.

Predicate Classification (PredCls) [58]: given objects' categories and boxes, decide which pairs interact and classify each one's predicate.

Scene Graph Classification (SGCls) [58]: given objects' boxes, classify objects, decide which pairs interact and classify each one's predicate.

Scene Graph Generation (SGGen) [58], also mentioned as Relationship Detection (RelDet) [43]: nothing is known in prior, detect objects and classify the predicates of the interacting pairs.

Phrase Detection (PhrDet) [43]: same to SGGen but evaluates the IoU of the predicate box for each pair.

We evaluate ATR-Net on all the above tasks per dataset, provide comparisons where available and present results for the rest of the tasks in Table 8.

Metrics: As explained in [43], a suitable metric for these problems is $\text{Recall}@x$, that counts the fraction of times the correct relationship is included in the top x confident predictions. A hyperparameter k, often not specified by prior works, measures the maximum predictions allowed per pair. Most works have seen Predicate Detection as a multiclass problem and they use $k = 1$ to reward the correct top-1 prediction for each pair [43, 46, 76, 67, 77]. Motivated by the fact that there are pairs annotated with more than one predicate classes, other works [12, 36, 11] tackle this as a multilabel problem and they use a k equal to the number of predicate classes to allow for predicate co-occurrences.

We re-formulate the metric as $\text{Recall}_k@x$ ($R_k@x$): for n examined subject-object pairs in an image, $R_k@x$ keeps the top-k predictions per pair and examines the x most confident out of nk total. Some past works [65, 69] have also identified this inconsistency and interpret k as the maximum number of edges allowed between a pair of object nodes. Thus, $k = 1$ is equivalent to 'graph constraints" and a larger k to "no graph constraints".

Notably, there is another inconsistency in Recall's definition: whether it is a micro- or macro-Recall. Let N be the number of testing images and gt_i the number of ground-truth pair annotations (directed scene graph edges)

Method	PredDet				PhrDet				SGGen			
	k = 1		k = 70		k = 1		k = 70		k = 1		k = 70	
	50	100	50	100	50	100	50	100	50	100	50	100
[67]	44.76	44.76	-	-	19.42	22.42	-	-	14.07	15.2	-	-
[68]	47.43	47.43	-	-	19.62	23.15	-	-	14.41	15.72	-	-
[43]	47.87	47.87	-	-	16.17	17.03	-	-	13.86	14.7	-	-
[61]	48.03	48.03	-	-	-	-	-	-	-	-	-	-
[10]	49.16	50.47	-	-	-	-	-	-	-	-	24.98	25.48
[19]	49.22	49.22	-	-	19.07	21.65	-	-	16.03	17.74	-	-
[76]	51.5	51.5	-	-	16.94	18.89	-	-	14.31	15.77	-	-
[22]	52.3	52.3	-	-	17.4	19.1	-	-	15.2	16.8	-	-
[46]	52.6	52.6	-	-	17.9	-	-	-	15.8	-	-	-
[3]	53.14	53.14	-	-	18.25	19.36	-	-	16.03	17.12	-	-
[77]	53.59	53.59	-	-	23.88	25.26	-	-	20.14	23.39	-	-
[74]	56.56	56.56	-	-	20.82	24.5	-	-	13.81	16.01	-	-
[66]	58.2	58.2	-	-	**31.5**	**36.1**	-	-	**23.9**	**26.8**	-	-
[5]	-	-	61.19	-	-	-	-	-	-	-	16.71	17.58
[12]	-	-	80.78	81.9	-	-	19.93	23.45	-	-	17.73	20.88
[17]	-	-	-	82.1	-	-	-	23.5	-	-	-	15.98
[48]	-	-	78.6	87.6	-	-	-	-	-	-	-	-
[13]	-	-	85.21	90.33	-	-	-	-	-	-	20.41	24.17
[4]	-	-	86.58	92.58	-	-	26.32	28.96	-	-	19.06	20.96
[40]	-	-	84.92	92.65	-	-	-	-	-	-	20.81	22.22
[36]	-	-	86.01	93.18	-	-	-	-	-	-	19.03	23.29
[11]	-	-	87.57	93.76	-	-	-	-	-	-	21.46	26.14
[64]	55.16	55.16	85.64	94.65	23.14	24.03	26.32	29.43	19.17	21.34	22.68	31.89
[63]	55.98	55.98	89.03	94.56	25.21	28.89	29.64	38.39	19.54	22.39	22.34	28.52
[25]	55.16	55.16	88.88	95.18	17.0	19.03	18.95	23.06	15.05	16.73	16.83	20.54
[16]	56.14	56.14	89.79	96.26	-	-	-	-	-	-	-	-
[73]	56.6	56.6	90.65	96.66	-	-	-	-	-	-	-	-
[26]	-	-	-	-	14.5	18.3	-	-	8.6	11.3	-	-
[75]	-	-	-	-	22.67	23.95	-	-	17.4	18.33	-	-
[32]	-	-	-	-	20.53	24.12	-	-	14.23	16.26	-	-
[49]	-	-	-	-	-	-	-	-	15.08	18.37	-	-
[33]	-	-	-	-	22.78	27.91	-	-	17.32	20.0	-	-
[37]	-	-	-	-	21.37	22.6	-	-	18.19	20.79	-	-
[34]	-	-	-	-	-	-	26.03	30.77	-	-	18.32	21.2
[6]	-	-	-	-	-	-	28.92	33.48	-	-	22.9	26.01
Ours micro	**58.4**	**58.4**	**91.0**	**96.97**	29.74	34.63	**33.2**	**41.01**	22.83	24.87	**26.04**	**31.94**
[21]	55.5	55.5	-	-	30.11	36.1	31.76	39.77	**25.74**	**29.82**	27.38	**34.12**
[18]	-	-	-	-	27.39	34.38	-	-	20.31	25.01	-	-
[70]	-	-	-	-	31.09	36.42	33.29	41.25	24.3	27.91	26.67	32.55
[69]	-	-	-	-	28.93	32.85	32.9	39.64	23.68	26.67	26.98	32.59
[71]	-	-	-	-	31.34	36.42	34.45	42.12	25.29	28.62	28.15	33.91
Ours macro	**58.78**	**58.78**	**95.32**	**98.47**	**31.96**	**36.54**	**36.06**	**43.45**	25.32	26.57	**28.28**	33.26

Table 2: Comparison with state-of-the-art on VRD [43], bold font indicates best results, underlined second-best. We compare using micro-Recall and macro-Recall (bottom table) with the respective methods.

in each image i. Then, having detected tp_i true positives in the image i, micro-Recall micro-averages these positives as $\frac{\sum_i^N tp_i}{\sum_i^N gt_i}$ and rewards correct predictions across dataset. Macro-Recall macro-averages detections in terms of images $\frac{1}{N}\sum_i^N \frac{tp_i}{gt_i}$, favoring correct predictions in images with fewer ground-truth annotations. Early works evaluate micro-Recall on VRD [43] and macro-Recall on VG200 [58], but later works often use the two types interchangeably and without consistency.

4.2. Results

We present and discuss evaluation results of ATR-Net on 6 datasets/splits. In all experiments we set $\lambda_r = \lambda_p = 1$, leaving further fine-tuning for future work. Our PyTorch code is publicly available[1].

Method	PredDet				PhrDet				SGGen			
	k = 1		k = 50		k = 1		k = 50		k = 1		k = 50	
	50	100	50	100	50	100	50	100	50	100	50	100
[17]	-	-	-	77.18	-	-	-	14.96	-	-	-	10.95
[13]	-	-	84.96	91.5	-	-	-	-	-	-	20.77	22.12
[40]	-	-	85.21	91.56	-	-	28.58	31.69	-	-	**21.49**	23.51
[35]	-	-	-	-	-	-	-	-	10.72	14.22	-	-
[52]	-	-	-	-	-	-	-	-	12.66	14.85	-	-
[34]	-	-	-	-	22.84	28.57	-	-	13.06	16.47	-	-
[18]	-	-	-	-	23.51	30.04	-	-	13.65	17.57	-	-
Ours micro	**66.31**	**66.52**	**90.6**	**95.97**	**28.13**	**33.91**	**29.98**	**37.75**	**16.66**	**19.35**	20.86	**23.66**
Ours macro	**66.44**	**66.44**	**95.53**	**98.52**	**37.13**	**42.54**	**40.43**	**48.51**	**24.06**	**27.29**	**26.47**	**31.67**

Table 3: Comparison with state-of-the-art on VG-MSDN [35], bold font indicates best results, underlined second-best. We compare using micro-Recall but include macro-Recall results as well for future reference.

Method	PredDet				PhrDet				SGGen			
	k = 1		k = 100		k = 1		k = 100		k = 1		k = 100	
	50	100	50	100	50	100	50	100	50	100	50	100
[67]	62.63	62.87	-	-	9.46	10.45	-	-	5.52	6.04	-	-
[61]	62.71	62.94	-	-	-	-	-	-	-	-	-	-
[19]	64.41	64.53	-	-	9.72	9.97	-	-	6.02	6.28	-	-
[68]	64.17	64.86	-	-	10.62	11.08	-	-	6.02	6.91	-	-
[5]	65.27	66.45	-	-	-	-	-	-	12.64	14.62	-	-
[74]	68.63	68.91	-	-	10.6	12.05	-	-	5.96	6.64	-	-
[66]	71.9	72.2	-	-	26.6	32.1	-	-	14.4	16.5	-	-
[75]	-	-	-	-	13.08	15.61	-	-	6.82	8.0	-	-
[36]	-	-	69.06	74.37	-	-	-	-	-	-	-	-
[11]	-	-	70.42	74.92	-	-	-	-	-	-	-	-
[6]	-	-	-	-	14.62	18.13	-	-	7.93	9.41	-	-
[21]	-	-	-	-	17.53	21.92	-	-	9.55	11.74	-	-
Ours micro	**71.92**	**72.23**	**91.65**	**96.61**	**27.84**	**33.47**	**29.49**	**36.99**	**15.81**	**18.27**	**17.31**	**21.08**
Ours macro	**72.3**	**72.32**	**96.74**	**98.8**	**36.05**	**41.32**	**38.85**	**46.37**	**21.99**	**24.86**	**24.13**	**28.66**

Table 4: Comparison with state-of-the-art on VG-VTE [67], bold font indicates best results, underlined second-best. We compare using micro-Recall but include macro-Recall results as well for future reference.

Method	PredDet				PhrDet				SGGen			
	k = 1		k = 24		k = 1		k = 24		k = 1		k = 24	
	50	100	50	100	50	100	50	100	50	100	50	100
[12]	-	-	88.26	91.26	-	-	23.95	27.57	-	-	20.79	23.76
[34]	-	-	-	-	-	-	26.91	32.63	-	-	19.88	23.95
Ours micro	**77.12**	**77.32**	**98.72**	**99.76**	**37.15**	**43.3**	**39.6**	**47.39**	**26.12**	**29.58**	**28.33**	**33.16**
Ours macro	**76.2**	**76.21**	**99.94**	**99.99**	**44.91**	**51.17**	**48.1**	**56.01**	**34.67**	**39.25**	**37.55**	**43.63**

Table 5: Comparison with state-of-the-art on sVG [12], bold font indicates best results, underlined second-best. We compare using micro-Recall but include macro-Recall results as well for future reference.

	Predicate Accuracy		
Method	top-1	top-5	top-10
[69]	52.0	79.37	85.6
Ours	**59.81**	**84.23**	**89.24**

Table 6: Comparison with state-of-the-art on VG80K [69], bold font indicates best results, underlined second-best. We compare using predicate accuracy as in [69].

Results on VRD [43]: Comparative results on VRD are presented on Table 2. Due to inconsistencies to the metric's definition, we make our best attempt to split the results into those that use the original micro-Recall [43] and those who use macro-Recall [58]. In the first setting, we outperform most other methods or achieve second-best results on all tasks. The main competitor [66] uses a similar strategy with multi-tasking, but employs external training

[1] https://github.com/deeplab-ai/atr-net

		PredDet				PredCls				SGCls				SGGen			
		k = 1		k = 50		k = 1		k = 50		k = 1		k = 50		k = 1		k = 50	
	Method	50	100	50	100	50	100	50	100	50	100	50	100	50	100	50	100
w/o context	[29]	-	-	-	-	-	-	-	-	-	-	-	-	6.84	9.85	-	-
	[47]	-	-	-	-	-	-	67.71	77.6	-	-	35.55	42.74	-	-	-	-
	[48]	-	-	-	-	-	-	72.5	81.7	-	-	-	-	-	-	-	-
	[72]	-	-	-	-	65	67.1	-	-	36.3	37.1	-	-	26.6	29.5	-	-
	[21]	-	-	-	-	65.3	67.3	-	-	35.9	36.6	-	-	**30.1**	**33.6**	-	-
	[70]	68.3	68.3	93.7	97.7	-	-	-	-	36.7	36.7	48.9	50.8	28.1	32.5	30.1	36.4
	[69]	68.4	68.4	-	-	-	-	-	-	36.7	36.7	-	-	27.9	32.5	-	-
	[71]	68.4	68.4	93.8	97.8	-	-	-	-	**36.8**	36.8	**48.9**	**50.8**	28.3	32.7	**30.4**	**36.7**
	Ours	**68.61**	**68.61**	**98.34**	**99.52**	**65.87**	**67.8**	**81.63**	**89.03**	36.0	**37.01**	44.71	48.24	21.43	26.39	22.7	28.14
w/ context	[58]	-	-	-	-	44.75	53.08	-	-	21.72	24.38	-	-	3.44	4.24	-	-
	[41]	-	-	-	-	45.55	53.66	-	-	23.37	26.29	-	-	4.33	5.57	-	-
	[27]	-	-	-	-	46.85	55.63	-	-	23.8	26.78	-	-	6.36	7.54	-	-
	[14]	-	-	-	-	56.65	57.21	-	-	23.71	24.66	-	-	13.18	13.45	-	-
	[56]	-	-	-	-	53.2	57.9	-	-	27.8	29.5	-	-	11.4	13.9	-	-
	[22]	-	-	-	-	51.9	58.3	-	-	24.3	26.6	-	-	4.8	6.0	-	-
	[60]	-	-	-	-	54.2	59.1	-	-	29.6	31.6	-	-	11.4	13.7	-	-
	[50]	-	-	-	-	56.6	61.3	-	-	38.2	40.4	-	-	-	-	-	-
	[20]	-	-	-	-	65.1	66.9	80.8	88.2	36.5	38.8	45.5	**50.8**	-	-	-	-
	[65]	-	-	**96.0**	**98.4**	65.2	67.1	81.1	88.3	35.8	36.5	44.5	47.7	27.2	30.3	30.5	35.8
	[8]	-	-	-	-	65.8	67.6	**81.9**	**88.9**	36.7	37.4	**45.9**	49.0	27.1	29.8	**30.9**	**35.8**
	[39]	-	-	-	-	64.2	66.4	-	-	38.6	39.7	-	-	**32.3**	**35.4**	-	-
	[53]	-	-	-	-	66.4	68.1	-	-	38.1	38.8	-	-	27.9	31.3	-	-
	[7]	-	-	-	-	66.4	68.1	-	-	38.5	39.3	-	-	27.9	31.2	-	-
	[57]	-	-	-	-	**67.0**	**68.5**	-	-	**41.0**	**41.7**	-	-	27.4	30.1	-	-
	[44]	-	-	-	-	-	-	68.0	75.2	-	-	26.5	30.0	-	-	9.7	11.3
	[23]	-	-	-	-	-	-	80.4	88.4	-	-	43.7	47.6	-	-	-	-

Table 7: Comparison with state-of-the-art on VG200 [58], bold font indicates best results, underlined second-best. We use only macro-Recall as all other methods in VG200 do. The table splits methods from those that do not, including us. Our results on PredDet and PredCls are better than or comparable to other methods, even those that employ context.

data (Wikipedia). However, we still outperform them on PredDet, due to our attentional scheme that allows ATR-Net to optimally exploit visual features. We notice a similar trend in macro-Recall results, outperforming other works on PredDet and PhrDet and performing on par on SGGen. Again, the main competitor [21] uses external data (VG) to train their object detector. VRD is a small dataset and using external data highly boosts performance.

Results on VG-MSDN [35]: Table 3 shows comparative results on VG-MSDN, where the original metric used is micro-Recall [35]. Our method clearly outperforms all other methods across tasks, with the margins being higher on PredDet and PhrDet, indicating that ATR-Net captures higher-level concepts of relationships. The main competitor [40] achieves higher R_{50}@50 on SGGen, but our difference depends on the training of the object detector, which is beyond our focus.

Results on VG-VTE [67]: VG-VTE is a much larger dataset and our method is able to scale up and achieve state-of-the-art results across tasks (Table 4), even outperforming [66] and [21] that were the main competitors on VRD using external data.

Results on sVG [12]: sVG is a peculiar dataset that has a wide variety of object categories and a large number of training images but only 24 predicate classes. ATR-Net attentional mechanism that uses language and spatial features proves that it can scale up to such a setting and dramatically improve over the current state-of-the-art (Table 5).

Results on VG200 [58]: VG200 is the most widely used VG split on Scene Graph Generation. Most related works solve the problem from a graph perspective and often employ context to improve their predictions. Since our work does not use context, we split Table 7 in two parts. ATR-Net belongs to first part, the methods that do not use context, achieving better or on par results on most tasks. We notice that although ATR-Net achieves the best results on PredDet and PredCls, it malperforms on SGCls and especially on SGGen. Following [65]'s observations, recent works on VG200 train a scene graph generator using perturbed bounding boxes. This approach both augments the data and provides robustness to noisy object detections. On the other hand, [71, 70, 69] train two Faster-RCNNs, one for objects and one for predicates. Far from our goal, we do not adopt any of these strategies and instead focus on improving the attentional scheme and pair filtering, managing to outperform or perform on par on PredDet and PredCls with most methods, independently of whether they use context or not.

Results on VG80K [69]: Lastly, we evaluate ATR-Net on the large-scale VG80K, where the metric is predicate accuracy. To handle the immense number of classes, [69] avoid cross-entropy and instead use a nearest-neighbor search on a multimodal space. We insist on softmax and

Recall	Dataset	PredCls				SGCls			
		k = 1		k = max		k = 1		k = max	
		50	100	50	100	50	100	50	100
micro	[35]	60.12	64.34	68.77	79.76	31.18	32.9	35.99	40.96
	[67]	60.3	66.88	66.56	78.56	33.3	36.53	37.06	43.19
	[12]	77.09	77.32	96.54	98.88	41.0	41.05	51.2	52.33
	[43]	56.37	57.55	73.18	83.92	39.09	39.85	51.01	58.83
macro	[35]	62.35	65.27	74.99	84.35	31.58	32.72	38.42	42.39
	[67]	65.73	69.51	78.0	86.51	33.7	35.45	40.3	44.37
	[12]	76.2	76.21	99.05	99.71	38.04	38.05	49.59	49.87
	[43]	57.09	57.7	80.16	88.71	39.42	39.75	55.88	61.78

Table 8: Results of ATR-Net on tasks that other methods do not evaluate on VG-MSDN [35], VG-VTE [67], sVG [12] and VRD [43], where "max" is the maximum value of k per dataset, equal to the number of predicate classes.

Ablation		PredCls				SGCls			
		k = 1		k = 50		k = 1		k = 50	
		50	100	50	100	50	100	50	100
A	No attention	56.43	59.8	68.69	78.28	32.44	33.91	39.4	43.85
	Single-head	62.39	64.76	76.6	84.81	34.5	35.55	42.43	46.22
	Spatial only	58.81	61.39	73.54	82.59	33.59	34.7	41.57	45.66
	Language only	64.24	66.77	78.41	86.33	35.17	36.26	43.27	46.97
T	W/o extra losses	64.58	66.79	79.58	87.32	35.37	36.28	43.67	47.35
R	W/o relevance task	53.12	61.86	57.6	70.34	29.55	33.84	32.24	38.62
	Filter of [12]	60.23	64.84	71.13	82.07	31.2	35.61	39.04	45.98
	ATR-Net Full	**65.87**	**67.8**	**81.63**	**89.03**	**36.0**	**37.01**	**44.71**	**48.24**

Table 9: Ablations of ATR-Net on VG200 [58]. A, T and R denote the respective pipeline steps under study, Attention, Translation and Relation respectively.

cross-entropy, achieving noticeably better results (Table 6), proving that our approach can scale up to settings with 300 times more classes than standard datasets.

Filling the gaps: For future reference, we report micro- and macro-Recall results of ATR-Net on tasks previously not evaluated across all datasets on Table 8.

4.3. Ablation Study

We question the importance of all steps in ATR-Net's pipeline both quantitatively and qualitatively. The variants and their comparison to our full model are summarized on Table 9. We present ablative results on VG200 [58], for PredCls and SGCls, as these settings exploit all ATR-Net's components and are not prone to object detection errors.

Attention: Attention is vital to ATR-Net since it leads the network's "focus" on the most discriminative visual features, while removing it results in an absolute 8% and 3.1% R_1@100 drop on PredCls and SGCls respectively. Single-head attention, conditioned on language and spatial features tends to focus on the specific details of the objects, yet irrespectively of class (Fig. 5) and thus does not capture the per-class important details of interaction, causing a 3.04% R_1@100 drop on PredCls. Multi-head attention with spatial features captures per-class important regions of the objects of interest, but ignoring the semantic information results in large drops in recall. Lastly, multi-head attention conditioned on language only is the strongest attentional variant, but focuses on per-class important characteristics of all present objects of the same category as the objects of interest, struggling to disambiguate cases when there are multiple instances of the same class (Fig. 5).

Translation: Do all losses contribute to the overall result? We train a variant without $\mathcal{L}_P^r$, $\mathcal{L}_{OS}^r$ and $\mathcal{L}_P^p$, $\mathcal{L}_{OS}^p$, thus supervising only the fusion ($\mathcal{L}_f^r$ and $\mathcal{L}_f^p$). We notice an absolute 1.01% and 0.73% R_1@100 drop on PredCls and SGCls respectively (Table 9), as these losses are responsible for supporting Eq. 1.

Relation: Another crucial factor of ATR-Net's effectiveness is that it separately handles the relevance task. If we instead train a single-task with the extra background class, we measure a significant drop in all recall and task settings, often worse than the (Table 9). This shows that the classifier cannot effectively minimize false positives without extra supervision. It is interesting that multi-tasking provides much larger gains in smaller recall thresholds (R_1@50, R_{50}@50), indicating that it forces the network to properly rank the different relationships. Note that without the relevance task we obtain close results to [50] (Table 7), that, similar to us, does not manually filter background pairs. Lastly, we train a variant of the pair filter of [12, 39] that employs only semantic and spatial information. Although reasonably more robust than with no filtering, this network fails to compete with the full ATR-Net, since the latter employs visual features with attention.

Discussion: Comparing the variants altogether, it becomes clear that language attention is the core component of ATR-Net. Combining with spatial attention boost the network's capability to deal with difficult examples where multiple object instances of the same classes occur, while multi-head shifts the focus of each head to class-specific characteristics. Multi-tasking for the relevance task is essential in order to avoid dataset-specific filters of negative samples, while the translation losses offer an extra recall margin by approximating the formulation of Translation Embeddings.

5. Conclusion

Misleading attention and dataset-specific biases hamper networks' scalability to significantly different settings. We address these challenges with a three-step pipeline that employs multi-head attention conditioned on language and spatial features, translation embeddings and multi-tasking to vote whether two objects interact and classify their interaction. We demonstrate the proposed approach's benefits on multiple datasets, metrics and task settings that capture the wide range of prior literature experiments. Interesting future directions would be to integrate ATR-Net's concepts with context or investigate ways to exploit the not annotated data to further boost single-instances' prediction.

References

[1] H. Akbari, S. Karaman, S. Bhargava, B. Chen, C. Vondrick, and S.-F. Chang. Multi-level Multimodal Common Semantic Space for Image-Phrase Grounding. In *Proc. CVPR*, 2019.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. CVPR*, 2018.

[3] S. Baier, Y. Ma, and V. Tresp. Improving Visual Relationship Detection Using Semantic Modeling of Scene Descriptions. In *Proc. ISWC*, 2017.

[4] H. Ben-younes, R. Cadène, N. Thome, and M. Cord. BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection. *ArXiv*, abs/1902.00038, 2019.

[5] Y. Bin, Y. Yang, C. Tao, Z. Huang, J. Li, and H. T. Shen. MR-NET: Exploiting Mutual Relation for Visual Relationship Detection. In *AAAI*, 2019.

[6] D. Chen, X. Liang, Y. Wang, and W. Gao. Soft Transfer Learning via Gradient Diagnosis for Visual Relationship Detection. In *Proc. WACV*, 2019.

[7] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang. Scene Dynamics: Counterfactual Critic Multi-Agent Training for Scene Graph Generation. *ArXiv*, abs/1812.02347, 2018.

[8] T. Chen, W. Yu, R. Chen, and L. Lin. Knowledge-Embedded Routing Network for Scene Graph Generation. In *Proc. CVPR*, 2019.

[9] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta. Iterative Visual Reasoning Beyond Convolutions. In *Proc. CVPR*, 2018.

[10] W. Cong, W. Wang, and W.-C. Lee. Scene Graph Generation via Conditional Random Fields. *ArXiv*, abs/1811.08075, 2018.

[11] Z. Cui, C. Xu, W. Zheng, and J. Yang. Context-Dependent Diffusion Network for Visual Relationship Detection. In *Proc. ACMMM*, 2018.

[12] B. Dai, Y. Zhang, and D. Lin. Detecting Visual Relationships with Deep Relational Networks. In *Proc. CVPR*, 2017.

[13] Y. Dai, C. Wang, J. Dong, and C. Sun. Visual Relationship Detection Based on Bidirectional Recurrent Neural Network. *Multimedia Tools and Applications*, 2019.

[14] A. Dornadula, A. Narcomey, R. Krishna, M. S. Bernstein, and L. Fei-Fei. Visual Relationships as Functions: Enabling Few-Shot Scene Graph Prediction. *ArXiv*, abs/1906.04876, 2019.

[15] S. Ghosh, G. Burachas, A. Ray, and A. Ziskind. Generating Natural Language Explanations for Visual Question Answering using Scene Graphs and Visual Attention. *ArXiv*, abs/1902.05715, 2019.

[16] N. Gkanatsios, V. Pitsikalis, P. Koutras, A. Zlatintsi, and P. Maragos. Deeply Supervised Multimodal Attentional Translation Embeddings for Visual Relationship Detection. *ArXiv*, abs/1902.05829, 2019.

[17] Y. Goutsu. Region-Object Relevance-Guided Visual Relationship Detection. In *Proc. BMVC*, 2018.

[18] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene Graph Generation with External Knowledge and Image Reconstruction. In *Proc. CVPR*, 2019.

[19] C. Han, F. Shen, L. Liu, Y. Yang, and H. T. Shen. Visual Spatial Attention Network for Relationship Detection. In *ACM Multimedia*, 2018.

[20] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In *Proc. NeurIPS*, 2018.

[21] Z.-S. Hung, A. Mallya, and S. Lazebnik. Union Visual Translation Embedding for Visual Relationship Detection and Scene Graph Generation. *ArXiv*, abs/1905.11624, 2019.

[22] S. J. Hwang, S. Ravi, Z. Tao, H. J. Kim, M. D. Collins, and V. Singh. Tensorize, Factorize and Regularize: Robust Visual Relationship Learning. In *Proc. CVPR*, 2018.

[23] G. Jaume, B. Bozorgtabar, H. K. Ekenel, J.-P. Thiran, and M. Gabrani. Image-Level Attentional Context Modeling Using Nested-Graph Neural Networks. *ArXiv*, abs/1811.03830, 2018.

[24] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proc. CVPR*, 2015.

[25] J. Jung and J. Park. Visual Relationship Detection with Language prior and Softmax. In *Proc. IPAS*, 2018.

[26] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Joint Learning of Object and Action Detectors. In *Proc. ICCV*, 2017.

[27] M. Khademi and O. Schulte. Dynamic Gated Graph Neural Networks for Scene Graph Generation. In *Proc. ACCV*, 2018.

[28] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

[29] M. Klawonn and E. Heim. Generating Triples With Adversarial Networks for Scene Graph Construction. *ArXiv*, abs/1802.02598, 2018.

[30] A. Kolesnikov, C. H. Lampert, and V. Ferrari. Detecting Visual Relationships Using Box Attention. *CoRR*, abs/1807.02136, 2018.

[31] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 123, 2016.

[32] B. Li. Visual Relationship Detection Using Joint Visual-Semantic Embedding. In *Proc. ICPR*, 2018.

[33] Y. Li, W. Ouyang, X. Wang, and X. Tang. ViP-CNN: Visual Phrase Guided Convolutional Neural Network. In *Proc. CVPR*, 2017.

[34] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang. Factorizable Net: An Efficient Subgraph-Based Framework for Scene Graph Generation. In *Proc. ICCV*, volume abs/1806.11538, 2018.

[35] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene Graph Generation from Objects, Phrases and Region Captions. In *Proc. ICCV*, 2017.

[36] K. Liang, Y. Guo, H. Chang, and X. Chen. Visual Relationship Detection With Deep Structural Ranking. In *Proc. AAAI*, 2018.

[37] X. Liang, L. Lee, and E. P. Xing. Deep Variation-Structured Reinforcement Learning for Visual Relationship and Attribute Detection. In *Proc. CVPR*, 2017.

[38] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu, and T. Mei. Rethinking Visual Relationships for High-level Image Understanding. *ArXiv*, abs/1902.00313, 2019.

[39] W. Liao, C. Lan, W. Zeng, M. Y. Yang, and B. Rosenhahn. Exploring the Semantics for Visual Relationship Detection. *ArXiv*, abs/1904.02104, 2019.

[40] W. Liao, S. Lin, B. Rosenhahn, and M. Y. Yang. Natural Language Guided Visual Relationship Detection. *CoRR*, abs/1711.06032, 2017.

[41] X. Lin, Y. Li, C. Liu, Y. Ji, and J. Yang. Scene Graph Generation Based on Node-Relation Context Module. In *Proc. ICONIP*, 2018.

[42] Q. Liu, F. Yu, S. Wu, and L. Wang. A Convolutional Click Prediction Model. In *Proc. CIKM*, 2015.

[43] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual Relationship Detection with Language Priors. In *Proc. ECCV*, 2016.

[44] A. Newell and J. Deng. Pixels to Graphs by Associative Embedding. In *Proc. NeurIPS*, 2017.

[45] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *Proc. EMNLP*, 2014.

[46] J. Peyre, I. Laptev, C. Schmid, and J. Sivic. Weakly-Supervised Learning of Visual Relations. In *Proc. ICCV*, 2017.

[47] F. Plesse, A. Ginsca, B. Delezoide, and F. J. Prêteux. Learning Prototypes for Visual Relationship Detection. In *Proc. CBMI*, 2018.

[48] F. Plesse, A. Ginsca, B. Delezoide, and F. J. Prêteux. Visual Relationship Detection Based on Guided Proposals and Semantic Knowledge Distillation. In *Proc. ICME*, 2018.

[49] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik. Phrase Localization and Visual Relationship Detection with Comprehensive Image-Language Cues. In *Proc. ICCV*, 2017.

[50] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo. Attentive Relational Networks for Mapping Images to Scene Graphs. In *Proc. CVPR*, 2019.

[51] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *Proc. NeurIPS*, 2015.

[52] D. Shin and I. Kim. Deep Image Understanding Using Multilayered Contexts. *Mathematical Problems in Engineering*, 2018.

[53] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to Compose Dynamic Tree Structures for Visual Contexts. In *Proc. CVPR*, 2019.

[54] S. Tripathi, A. Bhiwandiwalla, A. Bastidas, and H. Tang. Using Scene Graph Context to Improve Image Generation. *ArXiv*, abs/1901.03762, 2019.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Proc. NeurIPS*, 2017.

[56] W. Wang, R. Wang, S. Shan, and X. Chen. Exploring Context and Visual Pattern of Relationship for Scene Graph Generation. In *Proc. CVPR*, 2019.

[57] S. Woo, D. Kim, D. Cho, and I.-S. Kweon. LinkNet: Relational Embedding for Scene Graph. In *Proc. NeurIPS*, 2018.

[58] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Proc. CVPR*, pages 3097–3106, 2017.

[59] N. Xu, A.-A. Liu, J. Liu, W. Nie, and Y. Su. Scene Graph Captioner: Image Captioning Based on Structural Visual Representation. *J. Visual Communication and Image Representation*, 2019.

[60] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph R-CNN for Scene Graph Generation. In *Proc. ECCV*, 2018.

[61] X. Yang, H. Zhang, and J. Cai. Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features. In *Proc. ECCV*, 2018.

[62] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring Visual Relationship for Image Captioning. In *Proc. ECCV*, 2018.

[63] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. C. Loy. Zoom-Net: Mining Deep Feature Interactions for Visual Relationship Recognition. In *Proc. ECCV*, 2018.

[64] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation. In *Proc. ICCV*, 2017.

[65] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *Proc. CVPR*, 2018.

[66] Y. Zhan, J. M. Yu, T. Yu, and D. Tao. On Exploring Undetermined Relationships for Visual Relationship Detection. In *Proc. CVPR*, 2019.

[67] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual Translation Embedding Network for Visual Relation Detection. In *Proc. CVPR*, 2017.

[68] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang. PPR-FCN: Weakly Supervised Visual Relation Detection via Parallel Pairwise R-FCN. In *Proc. ICCV*, 2017.

[69] J. Zhang, Y. Kalantidis, M. Rohrbach, M. Paluri, A. M. Elgammal, and M. Elhoseiny. Large-Scale Visual Relationship Understanding. In *Proc. AAAI*, 2019.

[70] J. Zhang, K. Shih, A. Tao, B. Catanzaro, and A. Elgammal. An Interpretable Model for Scene Graph Generation. *CoRR*, abs/1811.09543, 2018.

[71] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical Contrastive Losses for Scene Graph Generation. In *Proc. CVPR*, 2019.

[72] L. Zhang, S. Zhang, P. Shen, G. Zhu, S. A. A. Shah, and M. Bennamoun. Relationship Detection Based on Object Semantic Inference and Attention Mechanisms. In *Proc. ICMR*, 2019.

[73] H. D. Zhou, C. Hu, C. Zhang, and S. Shen. Visual Relationship Recognition via Language and Position Guided Attention. In *Proc. ICASSP*, 2019.

[74] L. Zhou, J. Zhao, J. Li, L. Yuan, and J. Feng. Object Relation Detection Based on One-shot Learning. *ArXiv*, abs/1807.05857, 2018.

[75] Y. Zhu and S. Jiang. Deep Structured Learning for Visual Relationship Detection. In *Proc. AAAI*, 2018.

[76] Y. Zhu, S. Jiang, and X. Li. Visual Relationship Detection with Object Spatial Distribution. In *Proc. ICME*, 2017.

[77] B. Zhuang, L. Liu, C. Shen, and I. D. Reid. Towards Context-Aware Interaction Recognition for Visual Relationship Detection. In *Proc. ICCV*, 2017.