Relationship Proposal Networks

Ji Zhang*[1], Mohamed Elhoseiny*[2], Scott Cohen[3], Walter Chang[3], Ahmed Elgammal[1]

[1]Department of Computer Science, Rutgers University
[2]Facebook AI Research
[3]Adobe Research

Abstract

Image scene understanding requires learning the relationships between objects in the scene. A scene with many objects may have only a few individual interacting objects (e.g., in a party image with many people, only a handful of people might be speaking with each other). To detect all relationships, it would be inefficient to first detect all individual objects and then classify all pairs; not only is the number of all pairs quadratic, but classification requires limited object categories, which is not scalable for real-world images. In this paper we address these challenges by using pairs of related regions in images to train a relationship proposer that at test time produces a manageable number of related regions. We name our model the Relationship Proposal Network (Rel-PN). Like object proposals, our Rel-PN is class-agnostic and thus scalable to an open vocabulary of objects. We demonstrate the ability of our Rel-PN to localize relationships with only a few thousand proposals. We demonstrate its performance on Visual Genome dataset and compare to other baselines that we designed. We also conduct experiments on a smaller subset of 5,000 images with over 37,000 related regions and show promising results.

1. Introduction

While object detection is progressing at an ever-faster rate, relatively little work has explored understanding visual relationships at a large scale with related objects visually grounded to image regions. Visual relationships [15, 21] are defined as ⟨subject, predicate, object⟩ tuples, where the "subject" is related to the "object" by the "predicate" relationship. Detecting visual relationships aims at not only predicting if a relationship exists in an image but also localizing the "subject" and the "object". The predicate region can be simply determined by the union of the subject and object box. There are various types of visual relationships that appear in the real world, non-comprehensively exemplified next. Positional relationships describe relative location between objects like ⟨glass, on, table⟩, ⟨bag, under, desk⟩, *etc*. Attributive relationships describe that an object is a part of another or is composed of another (e.g., ⟨brick, of, building⟩, ⟨man, with, glasses⟩). This requires an understanding beyond spatially relating the two objects. A third type of relationship describes interactions between living objects like ⟨person, dancing with, person⟩, and ⟨man, riding, horse⟩. Here, a posture-level understanding is needed since recognizing these interactions rely on how each object is posed to the other. A fourth type of relationship

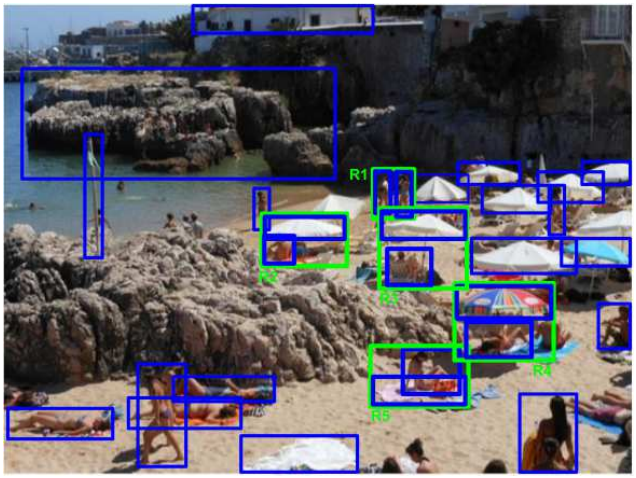


Figure 1: Given an image and its object proposals, only a handful pairs of them can form a set of meaningful relationships. Blue boxes are individual object proposals and green boxes are relationships that contain subjects and objects. In this figure the relationships are R1:⟨person, take pictures of, person⟩, R2, R3, R4:⟨person, under, umbrella⟩, and R5:⟨person, sit on, blanket⟩. Considering all pairs of object proposals is not only computationally expensive, but will include many false positives, i.e., pairs that don't form any relationship.

*The first two authors contributed equally to this work

includes interactions between living and non-living objects like ⟨kid, flying, kite⟩ and ⟨man, throwing, frisbee ⟩. In addition to difficult pose-level understanding needed for this type, the interacting objects might be far from each other which makes it further challenging (e.g., ⟨kid, flying, kite⟩). To handle all of these cases, it would be impractical to hand-write rules that can determine an arbitrary relationship between any two regions. The aforementioned challenges strongly motivate the need to learn the connection between image regions from data; this is the goal of our work.

Assuming the availability of a fixed dictionary of objects categories, the solution adopted in [15] for detecting relationship labels is to first detect all the individual objects in images and consider all pairs as potential ⟨subject, object⟩ pairs. The objects are detected by training a Faster-RCNN on a set of 100 types of objects, and similarly a predicate detector is learned to detect one out of the 70 predicates (from a closed dictionary of predicates). This limitation can be avoided by class-agnostic object proposals. However, in order to have a good recall rate, the number of proposals cannot be too small. In [25], $\sim$2000 proposals are used while the number is reduced to 1000 in [27]. In [19], they manage to use only 300 proposals at test-time. However, the complexity becomes quadratic when considering all pairs of proposals. Even if the number of proposals is as small as 300, we still need to recognize all 90,000 pairs, making it a computational bottleneck for relationship detection systems. Moreover, an image with many individual objects might only contain a handful of relationships, as shown in Figure 1. Recently, the Visual Genome dataset [12] has been released, which contains a total of 108,077 images with 33,877 object categories. Clearly, it is not straightforward to apply any closed-dictionary method at this scale, since the 33,877 object labels are too many for a CNN-based classification to perform well.

In this paper, we introduce Relationship Proposal Networks (Rel-PN) to extend the idea of object proposals to visual relationships. In particular, we aim to directly propose a set of potential ⟨subject, object⟩ pairs without considering every pair of individual objects. The resulting number of proposed pairs is a few thousand, which is an order of magnitude less than the number due to quadratic complexity. We call these pairs visual relationship proposals, since they are good candidates with high recall rates for relationships, and their computational cost is much lower than either exhaustive search (using a sliding window search) or by considering all object pairs. We propose an end-to-end trainable network with three branches for proposing subjects, objects and relationships, respectively. We use an efficient strategy to select candidate pairs that satisfy spatial constraints. The resulting pairs are then passed to a network module designed to evaluate the compatibility using both visual and spatial criteria, where incompatible pairs are filtered out and the remaining pairs are the final relationship proposals. We further compare our method with several intuitive baselines using individual object proposals, and we demonstrate that our method exhibits both higher recall rates and faster test-time performance.

2. Related Work

Object Proposals. Object proposal methods can be generally classified into two types: unsupervised approaches, including super-pixel merging [25, 4, 2] and objectness evaluation [1, 27], and supervised region prediction based on learned deep features from CNNs [19, 11, 3]. The latter has become increasingly popular since proposal generation can be simply performed using one CNN forward pass with near real-time running speed. With a minor sacrifice in accuracy, it is possible to integrate the proposal network into an end-to-end trainable detection system, enabling higher detection efficiency [6, 18, 14].

Object Relationship Exploration. There is significant literature that explores relationships between multiple objects, including object co-occurrence [17, 22, 13] and semantic segmentation [9, 24]. Spatial relationships have also been studied to improve both object-level and pixel-precision tasks [7, 9]. The goal of these methods is to utilize connections between objects to improve individual object recognition. In contrast, our task aims to recognize the entire relationship. Additionally, action/interaction recognition [20, 26, 16] has been a well-studied area where the "subject" is a human and "predicate" is a verb. In our work, we study general relationships with different types, where the"subject" and "predicate" are not constrained.

Visual Relationship Detection. Progress has been made on visual relationship recognition and detection tasks. In [21], the concept of visual phrases is introduced to represent relationship tuples. In [15], a new relationship detection model is proposed to not only recognize the relationship but to also locate the related objects. However, this method is restricted to a limited set of predicates/relations (i.e., 70 object labels and 100 predicate labels). In [5], a classification-free approach is proposed for visual relationship recognition, but it does not localize the objects in the predicted relationship.

We also notice that some state-of-the-art object detection methods [14, 23, 18] have removed the object proposal step and directly output detection boxes with labels. We argue that relationship proposals are still necessary and difficult to avoid for three reasons. First, the elimination of object proposals is usually realized by regressing and classifying anchor boxes (i.e., a set of location- and shape-predefined boxes), where the number of anchor boxes are at the same scale of feature maps (e.g., 8732 boxes in [14]). Simply applying this strategy to relationship detection would require considering a quadratic number of anchor boxes, which is

not tractable at large scale. Second, classification requires limited object categories, while relationship descriptions in the real-world are usually open. Third, proposing relationships involves not only localizing salient regions but also evaluating the visual connection between regions, making it more challenging than simply proposing objects.

3. Model Architecture

We consider three important aspects while designing our model. **(1) Relationship compatibility:** we model the probability of two regions being related to one-another (i.e., relationship compatibility predictor), **(2) Efficiency:** Bounding the relationship regions (i.e. ⟨subject, object⟩ pairs) that are checked for compatibility by (1), and **(3) Subjectness and objectness:** We account for the fact that the subject and object coming from different distributions. This is modeled by a different sub-network in contrast to the sub-network that models the probability of a region being an object (we call this objectness).

Subjectness and objectness sub-networks: We start to address the aforementioned aspects by modeling the probability of being an subject given a region (i.e., subjectness) and the probability of being an object given a region (i.e., objectness). It may be intuitive that subjects and objects should exist within the same category space. However, we will show later that the distributions of subject and object categories are biased differently; see section3.1. Our model discriminatively learns these two distributions by separate sub-networks that we designate as subjectness and objectness sub-networks.

Relationship compatibility module: The subjectness and the objectnness sub-networks produce regions with high probability of being subjects or objects respectively, but these regions might not have a connecting relationship. Hence, the need to learn the compatibility with the relationship becomes apparent. The relationship compatibility module takes a subject-object pair and their context (i.e., the union in our case) and produces a relationship compatibility score between the two regions. These scores are used to discard subject-object regions that do not have a relationship.

Pruning subject-object pairs: While the compatibility module could be fed regions with high subjectness and objectness scores, it is still computationally expensive to evaluate the compatibility for all subject-object pairs. This motivates further pruning of the pairs. Our solution starts by introducing a third sub-network, which is trained to detect the union-box of a relationship with ground truth annotation as the union box of subject and object pairs. We observed that this sub-network can locate the union box alone with 94% recall. Our idea is to prune the subject-object pairs by using this high-recall sub-network to generate a set of union boxes, and then select only the subject-object pairs whose union rectangles overlap with the generated union boxes by at least 50%. We found this approach to be highly effective in reducing the computational complexity.

Apart from these concerns, we also aim at a model that can be trained and tested end-to-end, i.e., it takes an image as input and directly outputs a set of relationship proposals. To address all these issues, we split the task into three steps which correspond to the three modules shown in Figure 2.

3.1. 3-branch RPN

We use the Region Proposal Networks (RPN) in Faster RCNN [19] to propose subjects, objects and unions respectively. In particular, we add two twin branches to RPN starting from conv3_1 down to conv5_3, resulting in a 3-branch RPN (Figure 2). The relationship branch is used to propose union boxes of subject-object pairs, while the subject and object branches propose their own boxes. This structure comes from our observation that the distribution of categories is different for subjects and objects. First, if a relationship is an interaction (i.e., the predicate is a verb) such as ⟨boy, fly, kite⟩, its subject is more likely to be a living being. In this case, the distribution of subjects' categories is more biased towards living beings than objects'. Second, for some positional relationships such as ⟨marking, on, t-shirt⟩, ⟨kite, in, sky⟩, and attributive relationships such as ⟨brick, of, building⟩, objects' category distribution is biased towards larger, coarser things while subjects' is towards smaller and finer ones. Therefore, two separated branches are necessary to learn these two different distributions.

Given an input image of size $W \times H$, we adopt VGG-16 architecture from conv_1_1 to conv_5_3 (13 layers) to convert the image into $C \times W' \times H'$ tensor of features, where $C = 512$, $W' = \lfloor \frac{W}{16} \rfloor$, and $H' = \lfloor \frac{H}{16} \rfloor$. Starting from this feature map, each branch is $N \times W' \times H'$ boxes in the form of $(x_{min}, y_{min}, x_{max}, y_{max})$, where N is the number of anchor boxes for each feature map location. Each of these boxes is associated with a confidence score for each branch. We consider 5 ratios and 7 scales for every location in the $W' \times H'$ grid, resulting in $N = 35$, where the 5 ratios are 1:4, 1:2, 1:1, 2:1, 4:1, and the 7 scales are 2, 4, 8, 16, 32, 64, 128. All the $3 \times N \times W' \times H'$ boxes and $3 \times N \times W' \times H'$ confidence scores from the three branches are passed as input to the proposal selection module.

At train-time, we feed subject and object branches with their corresponding ground-truth boxes. For the relationship branch, we use the union of subject and object box as ground-truth for each relationship. We fix the parameters of conv1_1 to conv2_2 and fine-tune conv3_1 to conv5_3.

3.2. Proposal Selection

In this module, each set of $N \times W' \times H'$ boxes are clipped to the image boundary, followed by non-maximum-suppression and sorting by their confidence scores. Then, we pick the top K_{rel}($K_{rel} = 5000$ in our model) relation-

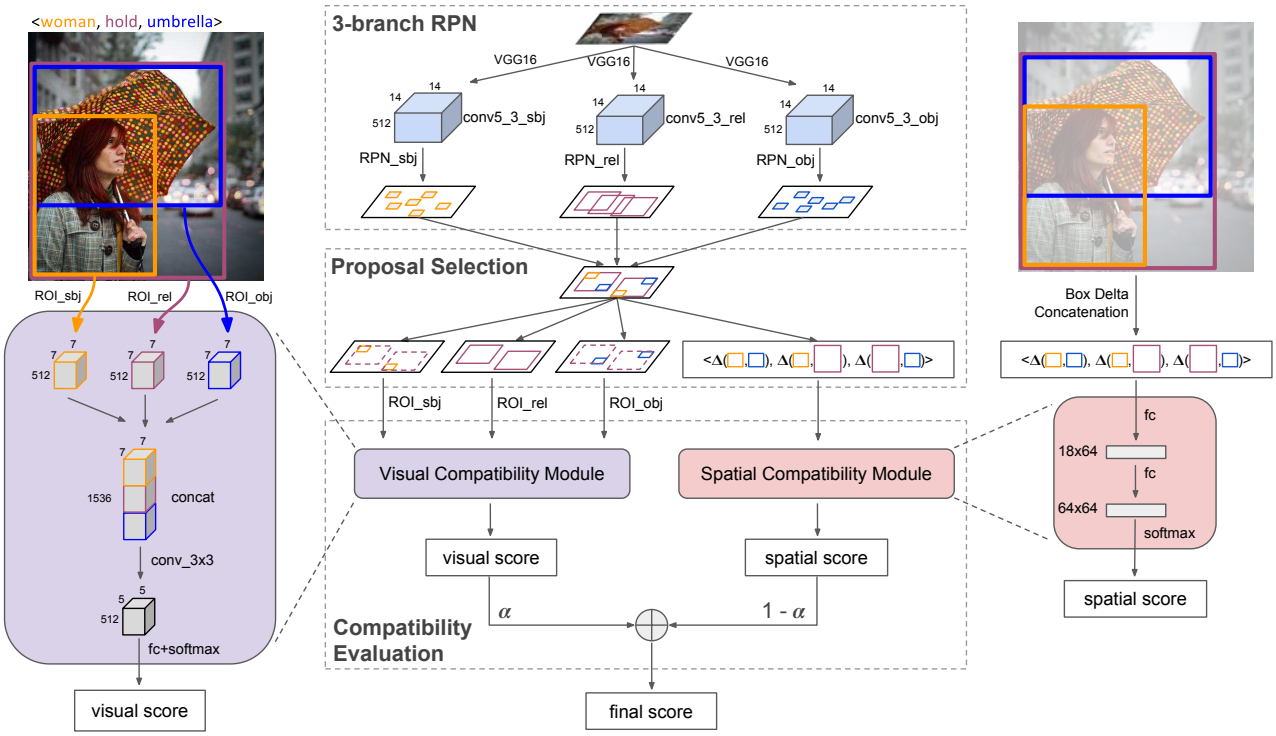


Figure 2: **Relationship Proposal Network architecture.** "sbj", "obj" and "rel" are abbreviations for "subject", "object" and "relationship". We feed an input image to a 3-branch RPN where each branch produces a set of candidate boxes. **Orange**, **purple**, **blue** boxes are subject, relationship and object proposals, respectively. The proposal selection module takes these boxes and selects qualified subject-object pairs, which are then used to generate visual and spatial features. In visual compatibility module, each subject box is ROI-pooled out as a $7 \times 7 \times 512$ feature, and so as for object and relationship boxes. The three features are then concatenated, followed by a convolutional (conv) layer, a fully-connected (fc) layer and a softmax layer to get the visual score; in spatial compatibility module, an 18-d feature is generated by concatenating the box deltas of $\langle$S, O$\rangle$, $\langle$S, P$\rangle$ and $\langle$O, P$\rangle$. Then we pass the feature to two fully-connected (fc) layers followed by a softmax layer to get the spatial score. Finally, visual and spatial scores are combined with different weights controlled by α to get the overall score.

ship boxes and do the following for each of them:

1. **Get search region:** Enlarge the relationship box by a factor (1.1 in our model) and use that as a search region;

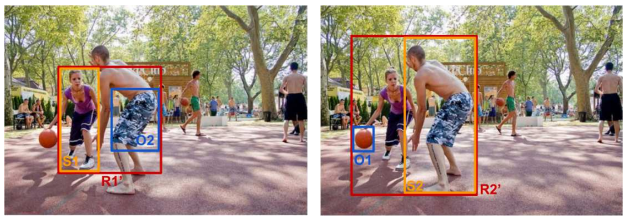
2. **Select individual subjects and objects:** Consider only those subject and object boxes that are within the search region, select top K_{sbj} of subject boxes and top K_{obj} of object boxes ($K_{sbj} = K_{obj} = 9$ in our model);

3. **Select qualified pairs:** For each of the $K_{sbj} \times K_{obj}$ subject-object pairs, we check whether its union box overlaps with the current relationship box by a threshold (0.5 in our model), and keep it only if this condition is satisfied; we also consider an additional set of K_{sbj} pairs where we pair each of the K_{sbj} subject boxes with the current relationship box. This additional set is generated specifically for those relationships whose subjects are located within objects, such as $\langle$kite, in, sky$\rangle$ and $\langle$window, of, building$\rangle$. In those cases, the object box coincides with the relationship box. We add all qualified pairs to an accumulative, duplicate-free list;

After these are done for all the K_{rel} relationship boxes, the result pairs are ranked by the average of subjectness and objectness scores, and the top N_{pair} pairs are kept. At test-time, these N_{pair} candidates are directly passed to the next module; at train-time, we need to generate positive and negative samples from them, since the compatibility module is trained as a binary classifier, which is fed with a batch of subject-object pairs as training samples, with binary labels indicating whether each pair is compatible or not.

For a positive sample, we define it as a pair satisfying *all* the following three conditions: 1) the subject box S overlaps with its closest ground-truth subject box S^{gt} by at least 0.5; 2) the object box O overlaps with its closest ground-truth object box O^{gt} by at least 0.5; 3) the two ground-truth boxes S^{gt} and O^{gt} should be a ground-truth relationship pair. The first two conditions ensure localization accuracy of each box, while the third condition excludes those pairs

(a) Two positive samples



(b) Two negative samples

Figure 3: **Sampling strategy for training.** Sampling on an example image with a) two positive pairs: $R_1 = \langle S_1, O_1 \rangle = \langle$girl, play, basketball$\rangle$, $R_2 = \langle S_2, O_2 \rangle = \langle$boy, wear, pants$\rangle$, and b) the corresponding negative pairs: $R'_1 = \langle S_1, O_2 \rangle$, $R'_2 = \langle S_2, O_1 \rangle$, which are obtained by pairing unrelated subjects and objects.

that are well located but mismatched.

For a negative sample, the definition is a pair satisfying *any* of the following three: 1) the subject box S overlaps with the ground-truth S^{gt} by less than 0.5; 2) the object box O overlaps with the ground-truth O^{gt} by less than 0.5; 3) both the subject and object overlaps are at least 0.5, but the two ground-truth boxes $\langle S^{gt}, O^{gt} \rangle$ is not a ground-truth relationship pair. The third condition is critical, since it enables the compatibility module to contrast correctly matched pairs against mismatched ones and learn the visual connection between subjects and objects in positive pairs. The sampling strategy is illustrated in Figure 3.

3.3. Compatibility Evaluation

The compatibility module is designed to evaluate the likelihood of a given box pair being a true relationship. We consider two aspects of the likelihood – visual compatibility, which analyzes coherence of the two boxes' appearance; spatial compatibility, which explores the locations and shapes of the two boxes. We designed two branches for these two purposes, get a visual score and spatial score from each branch, then integrate them into a final score(as shown in "Compatibility Evaluation" of Figure 2). The following paragraphs introduce the two components of this module.

Visual Compatibility: The input to this component is visual features of the samples selected from the last module. Each feature is obtained by extracting the conv5_3 features within the subject, object and the union box using ROI-pooling, then concatenating the three features into one. Since the feature of each box is $512 \times 7 \times 7$, we end up with a $1536 \times 7 \times 7$ concatenated feature map. Note that we also integrate the feature of the union box since it provides contextual information (i.e., visual feature of the whole relationship region). On this feature map we apply a convolution layer using a 3×3 filter with no zero-padding, shrinking the feature map from 7×7 to 5×5. We do this for two reasons: one is to learn a representative feature for the concatenation, the other is to reduce the size of parameters. After that, we append one fully-connected layer with 2048-d output and a softmax layer to generate a probability as the visual score.

Spatial Compatibility: The spatial feature of each sample is obtained by considering the difference between subject, object and relationship boxes. Specifically, a spatial feature is a vector of 18 dimensions concatenating three 6-d vectors, each indicating the difference of subject and object boxes $\Delta(S, O)$, subject and relationship boxes $\Delta(S, P)$, object and relationship boxes $\Delta(O, P)$. We adopt the idea of box regression [8] and use box delta as the metric of box difference. Specifically, $\Delta(S, O) = (t_x^{SO}, t_y^{SO}, t_w^{SO}, t_h^{SO}, t_x^{OS}, t_y^{OS})$ where each dimension is given by

$$\begin{aligned} t_x^{SO} &= (x^S - x^O)/w^S, \quad t_y^{SO} = (y^S - y^O)/h^S, \\ t_w^{SO} &= log(w^S/w^O), \quad t_h^{SO} = log(h^S/h^O), \\ t_x^{OS} &= (x^O - x^S)/w^O, \quad t_y^{OS} = (y^O - y^S)/h^O, \end{aligned} \tag{1}$$

where x^S, y^S, w^S, h^S denotes the center coordinates of a subject box, and similarly x^O, y^O, w^O, h^O is for an object box. The first 4 dimensions $(t_x^{SO}, t_y^{SO}, t_w^{SO}, t_h^{SO})$ is the box delta that regresses the subject box to the object box, while the last 2 dimensions (t_x^{OS}, t_y^{OS}) comes from the box delta $(t_x^{OS}, t_y^{OS}, t_w^{OS}, t_h^{OS})$ that regresses the object box to the subject, excluding $t_w^{OS} = log(w^O/w^S)$ and $t_h^{OS} = log(h^O/h^S)$ since $t_w^{OS} = 1 - t_w^{SO}$ and $t_h^{OS} = 1 - t_h^{SO}$. Similarly, we define $\Delta(S, P) = (t_x^{SP}, t_y^{SP}, t_w^{SP}, t_h^{SP}, t_x^{PS}, t_y^{PS})$, and $\Delta(O, P) = (t_x^{OP}, t_y^{OP}, t_w^{OP}, t_h^{OP}, t_x^{PO}, t_y^{PO})$. We concatenate $\Delta(S, O)$, $\Delta(S, P)$ and $\Delta(O, P)$ to get the 18-d feature, which is then passed to two consecutive fully-connected layers with 64 outputs. A softmax layer is appended in the end to produce the spatial score.

Once we have the visual score p_v and spatial score p_s, we integrate them by a convex combination defined as

$$p = \alpha p_v + (1 - \alpha) p_s \tag{2}$$

where p is the combined score, α is the ratio of visual compatibility, which can be learned using existing linear programming methods. We empirically set $\alpha = 0.8$ for all experiments and found that this fixed value works just as well. We also conduct a comprehensive evaluation on different values of α in section 4.2.

4. Experiments

We evaluate our model by localizing relationships in images. To our best knowledge we are the first to study rela-

5000 proposals	IoU≥0.5	IoU≥0.6	IoU≥0.7
SS, pairwise, 71×71	18.4	12.3	7.2
SS, nns, 100×50	19.5	12.6	7.1
SS, nns, 200×25	17.5	10.5	5.5
SS, nns, 400×13	14.8	8.4	4.2
EB, pairwise, 71×71	20.8	14.7	8.3
EB, nns, 100×50	21.9	14.8	7.5
EB, nns, 200×25	21	13	5.8
EB, nns, 400×13	18.7	10.5	4.2
RPN, pairwise, 71×71	27.3	19.2	9.4
RPN, nns, 100×50	32.5	22.5	9.8
RPN, nns, 200×25	34	21.1	8.1
RPN, nns, 400×13	28.3	15.8	5.2
Rel-PN, pro_sel	37.1	22	8.5
Rel-PN, pro_sel + spt	34.2	20.2	7.8
Rel-PN, pro_sel + vis	39.1	24	9.7
Rel-PN, pro_sel + vis + spt	**39.4**	**24.2**	**9.9**

Table 1: **Recall rates on VG by 5000 proposals.** "IoU≥t" means *both* subject and object boxes overlap with ground-truth by at least t. "Rel-PN" represents our model, "nns" denotes nearest neighbors search, "pro sel" denotes proposal selection, "vis" and "spt" stand for visual and spatial compatibility.

tionship proposals, hence we demonstrate the necessity and superiority of our method over several strong baselines derived from individual object proposals. We conduct experiments and report state-of-art results on two datasets: Visual Genome (VG) relationships [12] and Visual Relationship Detection (VRD) dataset [15].

4.1. Experimental Setup

Baseline Models. Our goal of studying the following baseline models is to evaluate the performance of relationship proposals generated by some intuitive strategies. Given a set of N object proposals $P = \{P_1, P_2, ..., P_N\}$, the first strategy is to simply pair every two object proposals (denoted as "pairwise"). A more sophisticated strategy is to pair each object with its geometric nearest neighbors (denoted as "nns"), since intuitively speaking, closer objects are more likely to be related. Specifically, our second baseline is to pair each proposal with each of the top K nearest neighbors $Q = \{Q_1, Q_2, ..., Q_K\}$, resulting in $N \times K$ relationship proposals. Euclidean distance between box centers is used as the distance metric. Every pair of $\langle P_i, Q_j \rangle (i = 1, ..., N, j = 1, ..., K)$ is used twice: one with P_i as subject and Q_j as object, and the other with Q_i as subject and P_j as object. Duplicate pairs are removed if exist.

We consider three object proposal methods for each of these two strategies: Selective Search (SS) [25], EdgeBoxes (EB) [27] and Region Proposal Network (RPN) [19]. For SS and EB, we directly apply them on our testing images. For RPN, we use both subject and object boxes as ground-truth for training, then use the trained model to generate individual object proposals.

IoU≥0.5	2000	5000	8000	10000
SS, pairwise	14.9	18.4	20.5	21.5
EB, pairwise	16.4	20.8	23.3	24.4
RPN, pairwise	18.1	27.3	32.6	35.3
Rel-PN, pro_sel	29.7	37.1	39.5	40.3
Rel-PN, pro_sel + spt	25.2	34.2	39	41.2
Rel-PN, pro_sel + vis	29.3	39.1	42.3	43.1
Rel-PN, pro_sel + vis + spt	**29.8**	**39.4**	**42.8**	**43.2**

Table 2: **Recall rates on VG with IoU≥0.5**. Abbreviations are the same with Table 1.

Our Model. We perform ablation studies on our model and compare results with the baselines. Specifically, we consider the following variants of our model:

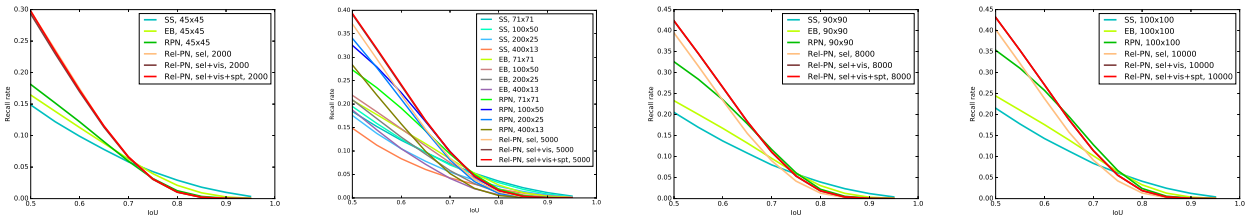
- **Proposal Selection.** We select top N proposals by the average of subjectness and objectness scores from the proposal selection module without feeding it to the compatibility module.
- **Proposal Selection + Spatial Compatibility.** We use only spatial confidence scores for the final proposals.
- **Proposal Selection + Visual Compatibility.** We use only visual confidence scores for the final proposals.
- **Proposal Selection + Visual + Spatial Compatibility.** This is our complete model. Visual and spatial scores are combined as shown in section 3.3.

Evaluation Settings. We design the following two experiments and evaluate recall rates in various settings:

1. **5000 proposals, varying IoU thresholds** We fix the number of relationship proposals as 5000, leading to $N = \lceil \sqrt{5000} \rceil = 71$ object proposals for the pairwise strategy. For the nearest-neighbor strategy, we generate 1) $N = 100$ object proposals with $K = 50$ nearest neighbors for each; 2) $N = 200$ object proposals with $K = 25$ nearest neighbors for each; 3) $N = 400$ object proposals with $K = 13$ nearest neighbors for each. We use 0.5, 0.6, 0.7 for Intersection over Union (IoU) thresholds and report recall rates of relationship proposals where *both* subject and object overlap with ground-truth by at least the threshold.
2. **IoU≥0.5, varying number of proposals** We fix the baseline strategy as pairwise and generate N_{rel} =2000, 5000, 8000 and 10000 relationship proposals for baselines and our models. For the baselines, the corresponding numbers of object proposals are $N = \lceil \sqrt{N_{rel}} \rceil = 45$, 71, 90 and 100. For our models, we directly select the top 2000, 5000, 8000 and 10000 proposals ranked by scores from our different modules.

4.2. Visual Genome

The Visual Genome dataset (VG) contains 108,077 images with 21 relationships on average per image. Each rela-



(a) 2000 proposals (b) 5000 proposals (c) 8000 proposals (d) 10000 proposals

Figure 4: **Recall vs IoU on VG with various numbers of proposals.** We compare against the pairwise baselines for 2000, 8000 and 10000 proposals while considering both pairwise and nearest-neighbor baselines for 5000 proposals.

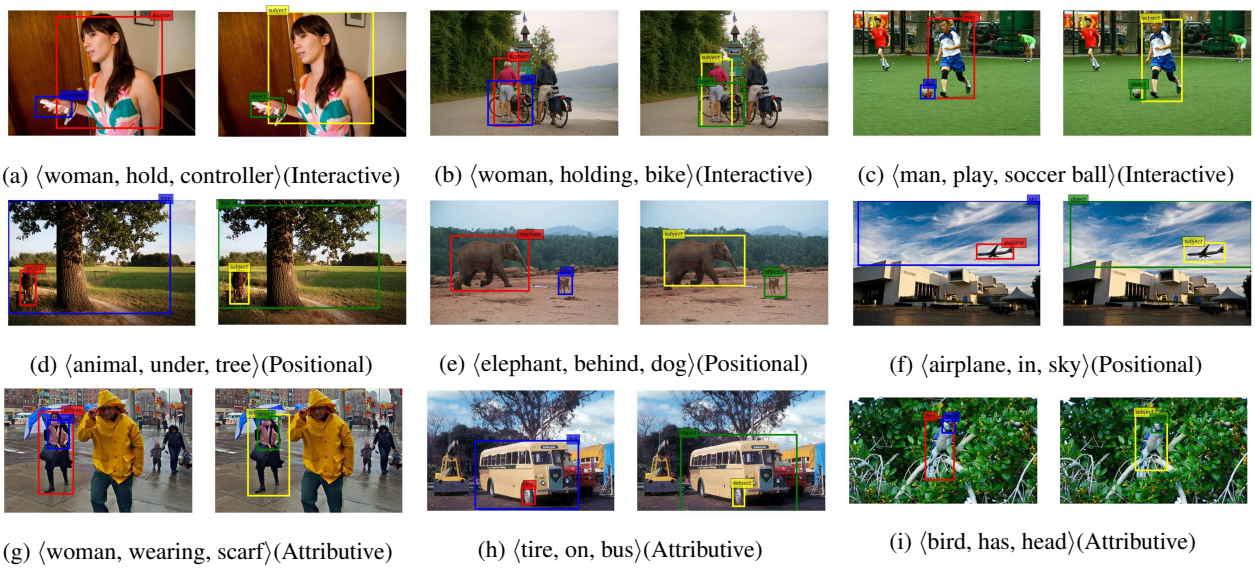
tionship is of the form ⟨subject, predicate, object⟩ with annotated subject and object bounding boxes. We follow [10] and split the data into $103,077$ training images and $5,000$ testing images. We train the model for 300k iterations with a learning rate of 0.001 for the first 200k and 0.0001 for the last 100k.

Quantitative Results. The results of the first experiment are shown in Table 1, while the second experiment is reported in Table 2. We also show Recall vs IoU curves with 2000, 5000, 8,000 and 10,000 proposals in Figure 4. We make the following observations:

- Table 1 shows that using 5000 proposals, which is of a reasonable complexity, our complete model achieves the highest recall against all baselines and variants of our model.
- Even without compatibility evaluation, the proposal selection module alone ("Rel-PN, pro_sel" in Table 1) can achieve 37.1% recall, due to the accuracy of union box localization, and the efficient strategy of selecting qualified subject-object pairs using the union boxes.
- The visual compatibility is clearly more important than spatial. Using only visual compatibility can lead to a suboptimal performance (39.1%), while using spatial compatibility alone exhibits an obvious drop in recall. This is mainly because for general relationships, the distribution of spatial features are usually more uniform and thus less discriminating than visual features. For example, the appearance of ⟨man, fly, kite⟩ usually involves a human holding the string of a kite in the sky. However, the man's size, the kite's shape and the distance between the man and kite often varies across different scenes, making it harder to learn by using spatial features alone. That said, the spatial compatibility is still better than the best nearest-neighbor baseline (37.1% vs 32.5%), since our spatial evaluation module learns to cover various relationships with different spatial layouts, while nearest-neighbor methods naively treat closer objects as providing better relationships.
- With a proper number of neighbors, the nearest-neighbor strategy is better than the pairwise strategy. For example, using Edgeboxes by 100 object proposals with 50 neighbors ("EB, nns, 100×50") has a higher recall (21.9%) than using Edgeboxes in a pairwise manner ("EB, pairwise, 71×71"). This benefit arises from considering more object proposals than pairwise (100 vs 71) and pairing with closest objects, which are intuitively more likely to be related. However, when the number of nearest neighbors K is much smaller than the number of object proposals N, there is an obvious decrease in performance. This is because a small number of nearest neighbors cannot cover medium or long distance relationships, such as ⟨boy, fly, kite⟩, where "boy" is on the ground and "kite" is high in the sky.
- As shown in Figure 4, our model works better for smaller IoU thresholds. We found that this is mainly due to the same reason why RPN is not good when IoU values are high (see Figure 2 in [19]), when unsupervised proposal methods (SS and EB) utilize pixel level clues (e.g., superpixels in SS and edges in EB) to determine object boundaries, while RPN-like networks regress proposals from anchor boxes using smaller size features (i.e., 7x7 from conv5_3). Therefore, the regressed proposals have less ability to guarantee that object boundaries can be exactly located in the original image. Nevertheless, our model still outperforms others when using a moderate number of proposals (e.g., 5000) with a reasonable IoU (e.g., IoU$\geq$0.7).

Qualitative Results. In Figure 5, we show example proposals generated by our model with their corresponding ground-truth. The phrase of each ground-truth relationship (e.g., ⟨girl, chasing, bubble⟩) is also shown for better illustration. Our model is able to cover all three types of relationships (interactive, positional, attributive). Note that subject and object boxes have various shapes and distances, while our model correctly finds meaningful relationships and accurately localizes subjects and objects by boxes.

Visual Compatibility Weight. In Table 3, we show recall rates with different values of the visual compatibility weight α. We can see that results are close as long as visual compatibility weighs are more than the spatial, since the spatial scores are generally less discriminating than visual scores.



(a) ⟨woman, hold, controller⟩(Interactive) (b) ⟨woman, holding, bike⟩(Interactive) (c) ⟨man, play, soccer ball⟩(Interactive)

(d) ⟨animal, under, tree⟩(Positional) (e) ⟨elephant, behind, dog⟩(Positional) (f) ⟨airplane, in, sky⟩(Positional)

(g) ⟨woman, wearing, scarf⟩(Attributive) (h) ⟨tire, on, bus⟩(Attributive) (i) ⟨bird, has, head⟩(Attributive)

Figure 5: **Example relationship proposals on VG.** Red and blue boxes are ground-truth subject and object, yellow and green boxes are outputs from our model.

5000 proposals	IoU≥0.5	IoU≥0.6	IoU≥0.7
1.0 visual, 0.0 spatial	39.1	24	9.7
0.9 visual, 0.1 spatial	39.3	24.2	9.8
0.8 visual, 0.2 spatial	**39.4**	**24.3**	**9.9**
0.7 visual, 0.3 spatial	39.3	24.2	9.9
0.6 visual, 0.4 spatial	39	24	9.9
0.5 visual, 0.5 spatial	38.5	23.8	9.7

Table 3: **Recall rates on VG with different values of α.**. The number of proposals is fixed as 5000.

IoU≥0.5	2000	5000	8000	10000
SS, pairwise	22.1	28	31.4	33
EB, pairwise	15.1	20.6	24.2	25.2
RPN, pairwise	28.9	36.2	41	43
Rel-PN, pro_sel	35.1	41.9	43.9	44.5
Rel-PN, pro_sel + spt	27.2	38.6	44	46.1
Rel-PN, pro_sel + vis	36.8	44.1	45.5	47
Rel-PN, pro_sel + vis + spt	**38.3**	**44.3**	**46.4**	**47.3**

Table 4: **Recall rates on VRD with IoU≥0.5.**

However, combining a moderate amount of spatial information with visual scores improves the performance (e.g., 0.3% gain from 39.1% of "1.0 visual, 0.0 spatial" to 39.4% of "0.8 visual, 0.2 spatial").

4.3. Visual Relationship Detection dataset

In this section we conduct experiments on the Visual Relationship dataset (VRD) from [15]. We use the same settings with the Visual Genome experiments. In Table 4 we observed that our model outperforms baselines on small datasets as well. We also notice that here our spatial module has an obviously better performance on VRD than on Visual Genome (e.g., 44% vs 39% for 8,000 proposals and 46% vs 41% for 10,000). This is mainly because the annotated relationships in this dataset are usually denser than Visual Genome, i.e., distances between subjects and objects are smaller. Hence, the spatial distribution of relationships is more biased and easier to learn by our spatial compatibility module.

For completeness, we include additional results for this dataset in the supplementary material.

5. Conclusion

We introduced the task of proposing visual relationships, which requires simultaneously localizing two regions that are related. Challenges include visual and spatial variation among all types of relationships and quadratic complexity if all pairs of individual objects are considered. We developed a new Rel-PN architecture, which addresses these challenges by utilizing both a heuristic spatial constraint and a learned compatibility metric to select a manageable number of relationship proposals. Our experiment demonstrated our model's efficiency and significant improvement over several baseline models. Future work includes applying our relationship proposals to a detection system that outputs linguistic descriptions of subject, object and predicates.

6. Acknowledgements

This research is partially funded by a gift from Adobe Research and National Science Foundation (NSF) under NSF-USA award #1409683.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, 2014.

[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016.

[4] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

[5] M. Elhoseiny, S. Cohen, W. Chang, B. Price, and A. Elgammal. Sherlock: Scalable fact learning in images. *arXiv preprint arXiv:1511.04891*, 2015.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition*, pages 2155– 2162, 2014.

[7] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[8] R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[9] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[10] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[13] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*, pages 239–253. Springer, 2010.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[15] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[16] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184. IEEE, 2011.

[17] T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[20] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013.

[21] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1745–1752. IEEE, 2011.

[22] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481–1488. IEEE, 2011.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[24] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[25] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[26] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24. IEEE, 2010.

[27] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.