SCENE GRAPH PARSING AND ITS APPLICATION IN CROSS-MODAL REASONING TASKS

by

JI ZHANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Ahmed Elgammal

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2020

ABSTRACT OF THE DISSERTATION

Scene Graph Parsing And Its Application in Cross-Modal Reasoning Tasks

by JI ZHANG

Dissertation Director:
Ahmed Elgammal

Scene graph parsing aims at understanding an image as a graph where vertices are visual objects (potentially with attributes) and edges are visual relationships among objects. This task is commonly seen as an extension to the object detection task where objects are detected individually, while the former requires recognizing relationships between object pairs. Therefore, scene graphs are usually seen as a better semantic representation of images for visual reasoning. In thesis we start with an inherent issue lying in scene graph parsing: the unbearable quadratic complexity of relationship detection. We develop an efficient model that effectively reduces the complexity from quadratic down to quasi-linear and show clear superiority over intuitive and strong baselines. Then we introduce two salient issues that naturally occur in scene graphs: Ambiguity in the language dimension and ambiguity in the visual dimension. The first happens when the vocabulary of objects and relationships are significantly large, and the second happens when multiple vertices or edges in a scene graph are from the same category and confuse the model to recognize the correct relational pairing. We propose two models that tackle these two problems separately, where the first model utilizes learnable embeddings to handle the ambiguity in the language dimension, while the second adds three types of losses that we design to for the model to learn to discriminate correct instances against confusing and hard negative instances. At last, with an accurately parsed scene graph, we discuss the topic of

using scene graphs as richer feature and deeper knowledge of the input visual signals for better visual-semantic cross-modal reasoning. We design and develop a model that follows such logic and apply it on the video story understanding task, which achieves satisfying advantage over strong baseline models. In summary, we claim that scene graphs can be accurately and efficiently obtained by our models, and that we can build a sophisticated system that employs scene graphs for more explicit and interpretable cross-modal understanding.

Acknowledgements

I would like to sincerely thank my advisor, Prof. Ahmed Elgammal, for providing me a very free environment to do whatever I am interested in. For the past four years, whenever I had problems or needed helps he would respond promptly and provided as much assistance as he could. he kindness and honestness is one of the major factors that motivate me to carry on honest and solid work after my graduation.

I also want to thank Dr. Jie Shen, who is currently an assistant professor in Stevens Institute of Technology, for unselfishly offering helps and suggestions at the beginning of my PhD career. Things weren't as good as they are today back then, and he was always willing to give me a hand when I was struggling in either academic research or daily life. Wish you the best of luck in your faculty career.

I would like to thank Shuchang Liu and Shijie Geng, who are also PhD students here and colleagues of mine at our office, Hill Center 270. They made my PhD life much more interesting than I could imagine, and the productiveness of my last 3 years is definitely attributed to them to a great extent. I sincerely wish you great future in your own PhD years, and hope we can hang around even after my graduation.

In addition, I want to specifically thank Cheng Da, currently a PhD student in meteorology at University of Maryland. I have known Cheng since our 7th grade in school, roughly 17 years ago. I am most lucky to have a high school friend who is in a driving distance from me, and with his company my PhD years were full of joy and much less of loneliness.

Finally, I want to thank everyone I have known during these years, including professors, friends from the companies I have interned in, colleagues from Rutgers and from the academia, et al.. I couldn't thank you more for making my PhD years a wonderful journey that I have never imagined when I started. It is a new chapter. The adventure continues.

Dedication

This work is dedicated to my mother. Being her son is my greatest fortune ever.

Table of Contents

Abstract ii

Acknowledgements iv

Dedication v

List of Tables 1

List of Figures 4

1. Introduction 8

1.1. Relationship Proposal and Scene Graph Parsing 8

1.2. Ambiguities in Scene Graph Parsing 8

1.2.1. Semantic Ambiguity 9

1.2.2. Visual Ambiguity 9

1.3. Visual-Semantic Understanding via Scene Graphs 9

2. Background and Related Work 11

2.1. Visual Relationships 11

2.2. Object Proposals 12

2.3. Object Relationship Exploration 12

2.4. Visual Relationship Detection 12

2.5. Semantically Guided Visual Recognition 14

2.6. Scene Graph Parsing 14

2.7. Phrase Grounding and Referring Expressions 15

2.8. Contrastive Training 15

2.9. Visual Question Answering (VQA) 16

2.10. Relational Reasoning 16

2.11. Video Story Question Answering . . . 17
2.12. Visual Relation Detection . . . 18
2.13. Character Naming . . . 18

3. Relationship Proposal Networks . . . 19
3.1. Introduction . . . 19
3.2. Model Architecture . . . 21
3.2.1. 3-branch RPN . . . 22
3.2.2. Proposal Selection . . . 23
3.2.3. Compatibility Evaluation . . . 25
3.3. Implementation Details . . . 27
3.4. Experiments . . . 28
3.4.1. Experimental Setup . . . 28
3.4.2. Visual Genome . . . 30
3.4.3. Visual Relationship Detection dataset . . . 34
3.5. More Qualitative Results on VG . . . 35
3.6. Limitations . . . 36

4. Large-Scale Visual Relationship Understanding . . . 41
4.1. Introduction . . . 41
4.2. Method . . . 43
4.2.1. Visual Module . . . 43
4.2.2. Semantic Module . . . 44
4.2.3. Training Loss . . . 45
4.3. Experiments . . . 49
4.3.1. Evaluation of Relationship Detection on VRD . . . 51
4.3.2. Scene Graph Classification & Detection on VG200 . . . 51
4.3.3. Relationship Recognition on VG80k . . . 51
4.3.4. Ablation Study . . . 53
4.3.5. Qualitative results . . . 57

4.4. Summary . . . 57

5. Graphical Contrastive Losses for Scene Graph Parsing . . . 58

5.1. Introduction . . . 58

5.2. Graphical Contrastive Losses . . . 61

5.2.1. Class Agnostic Loss . . . 61

5.2.2. Entity Class Aware Loss . . . 62

5.2.3. Predicate Class Aware Loss . . . 63

5.2.4. Complexity Analysis . . . 64

5.3. RelDN . . . 64

5.4. Implementation Details . . . 67

5.5. Experiments . . . 68

5.5.1. Evaluation Settings . . . 68

5.5.2. Loss Analysis . . . 70

5.5.3. Loss Analysis with the Official mAP metrics . . . 71

5.5.4. Model Analysis . . . 72

5.5.5. Comparison to State of the Art . . . 73

5.5.6. Qualitative Results . . . 75

5.6. Summary . . . 75

6. Video Story Understanding with Character-Aware Relations . . . 78

6.1. Introduction . . . 78

6.2. Method . . . 81

6.2.1. Character-Aware Frame Understanding . . . 81

6.2.2. Character-Aware Relation Detection . . . 83

6.2.3. Character-Aware Reasoning Network . . . 84

6.2.4. Implementation Details . . . 87

6.3. Experiments . . . 87

6.3.1. Dataset . . . 87

6.3.2. Baselines . . . 88

6.3.3. Experimental Setup 88
6.3.4. Comparison to State-of-the-Art 89
6.3.5. Ablation Study 90
6.3.6. Qualitative Results 90
6.4. Analysis on the Influence of Question Type 91
6.5. Summary 91

7. Conclusions 93

8. Future Direction 95

References 97

List of Tables

3.1. **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. 29

3.2. **Recall rates on VG with IoU≥0.5**. Abbreviations are the same with Table 3.1. 29

3.3. **Recall rates on VG with different values of α.** The number of proposals is fixed as 5000 33

3.4. **Recall rates on VRD with IoU≥0.5.** 34

3.5. **Recall rates on VRD by 5000 proposals.** Abbreviations are the same with Table 1 in the paper. 35

4.1. Comparison with state-of-the-art on the VRD dataset. 50

4.2. Comparison with state-of-the-art on the VG200 dataset. 50

4.3. Results on all relation classes and tail classes ($\#occurrence \leq 1024$) in VG80k. Note that since VG80k is extremely imbalanced, classes with no greater than 1024 occurrences are still in the tail. In fact, there are more than 99% of relation classes but only 10.04% instances of these classes that occur for no more than 1024 times. 52

4.4. Ablation study of our model on VG80k. 53

4.5. Performances of our model on VG80k validation set with different values of the scaling factor. We use scaling factor $\lambda = 5.0$ for all our experiments on VG80k. 54

4.6. Performances of triplet loss on VG80k validation set with different values of margin m. We use margin $m = 0.2$ for all our experiments in the main paper. . 55

5.1. Ablation Study on our losses with the official mAP$_{rel}$, mAP$_{phr}$ and score metrics. Metric marked with a * means "under" and "hits" are excluded from evaluation. The fluctuating numbers in mAP$_{rel}$, mAP$_{phr}$ and score indicate that the mAP metrics are unstable and unreliable, while when "under" and "hits" are excluded, all the results become consistent with Table 5.3. 69

5.2. Comparison of our model with Graphical Contrastive Loss vs. without the loss on 100 images containing the 5 classes that suffer from the two aforementioned confusions, selected via visual inspection on a random set of images. The metrics are the official mAP$_{rel}$, mAP$_{phr}$ and the score. The "under" and "hits" predicates are not in this 100 image subset. 69

5.3. Ablation Study on our losses. We report a frequency-balanced wmAP instead of mAP, as the test set is extremely imbalanced and would fluctuate wildly otherwise (see fluctuations in columns "under" and "hits"). We also report score$_{wtd}$, which is the official OI scoring formula but with wmAP in place of mAP. "Under" and "hits" are not highlighted due to having too few instances. . 72

5.4. Comparison of our model with Graphical Contrastive Loss vs. without the loss on 100 images containing the 5 classes that suffer from the two aforementioned confusions, selected via visual inspection on a random set of images. 72

5.5. Ablation Study on RelDN modules. *sem only* means using only the semantic module without training any model; $\langle$*S,P,O*$\rangle$ means using only the $\langle$S,P,O$\rangle$ concatenation without the separate S,O layers in the visual module; *vis* means our full visual module, and *spt* means spatial module. "Under" and "hits" are not highlighted due to having too few instances. 73

5.6. Ablation Study on the margin threshold m. We use $m = 0.2$ everywhere in our experiments. 73

5.7. Comparison with state-of-the-arts on VG. L_0 **only** is the RelDN without our losses. We also include results of our model with ResNeXt-101-FPN as the backbone for future work reference. 73

5.8. Comparison with state-of-the-art on VRD ($-$ means unavailable / unknown). Same with Table 5.7, L_0 **only** is the RelDN without our losses. "Free k" means considering k as a hyper-parameter that can be cross-validated. 73

5.9. Comparison with models from OpenImages Challenge. RelDN* means using the same entity detector from *Seiji*, the champion model. Overall is computed as 0.3*Public+0.7*Private. Note that this table uses the official mAP$_{rel}$ and mAP$_{phr}$ metrics. 74

6.1. Results on the TVQA test set for models that use time-stamp annotation ('w/ ts'). We compare to other baselines on the six TVQA sub-datasets individually. "V only" means using only global CNN features without subtitles. 86

6.2. Results on the TVQA test set for models that do not use time-stamp annotations ('w/o ts'). We compare to other baselines on the six TVQA sub-datasets individually. "V only" means using only global CNN features without subtitles. 86

6.3. Ablation study both with and without the time stamps. "Sub", "Objs", "Rels" and "nm" represent subtitles, objects, relationships and names, respectively. . . 89

6.4. The influence of question type for different methods with time stamps. "Sub", "Objs", "Rels" and "nm" represent subtitles, objects, relations and names, respectively. 91

List of Figures

3.1. **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 ⟨S, O⟩, ⟨S, P⟩ and ⟨O, P⟩. 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. 22

3.2. **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. . 25

3.3. **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. . . . 31

3.4. **Example relationship proposals on VG.** Red and blue boxes are ground-truth subject and object, yellow and green boxes are outputs from our model. 32

3.5. **Recall vs IoU on VRD with various numbers of proposals.** 36

3.6. **More example relationship proposals on VG.** Red and blue boxes are ground-truth subject and object, yellow and green boxes are outputs from our model. . 39

3.7. **Illustration of our model's limitation.** These four output proposals are ranked prior to the one shown in Figure 3.6m, which is supposed to be ranked at the top according to human's perceptual intuition about this image, since it is mostly about a man lies beside a dog. 40

4.1. Relationships predicted by our approach on an image. Different relationships are colored differently with a relation line connecting each subject and object. Our model is able to recognize relationships composed of over $53,000$ object categories and over $29,000$ relation categories. 42

4.2. (a) Overview of the proposed approach. L_s, L_p, L_o are the losses of subject, relation and object. Orange, purple and blue colors represent subject, relation, object, respectively. Grey rectangles are fully connected layers, which are followed by ReLU activations except the last ones, i.e. w_3^s, w_5^p, w_3^o. We share layer weights of the subject and object branches, i.e. w_i^s and w_i^o, $i = 1, 2...5$. 43

4.3. Top-5 relative accuracies against the 3-branch Fast-RCNN baseline in the tail intervals. The intervals are defined as bins of 32 from 1 to 1024 occurrences of the relation classes. 52

4.4. Qualitative results. Our model recognizes a wide range of relation ship triples. Even if they are not always matching the ground truth they are frequently correct or at least reasonable as the ground truth is not complete. 56

5.1. Example of failure of models without our losses and success of our losses. (a) RelDN learned with only multi-class cross-entropy loss incorrectly relates the man with the microphone, while (b) RelDN learned with our *Graphical Contrastive Losses* detects the correct relationship $\langle man, holds, guitar \rangle$. . . . 59

5.2. Examples of Entity Instance Confusion and Proximal Relationship Ambiguity. Red boxes highlight relationships our baseline model predicts incorrectly. (a) the man is not holding the predicted wine glass. (b) the guitar player on the right is not playing drum. 60

5.3. The RelDN model architecture. The structures of conv_body_det and conv_body_rel are identical. We freeze the weights of the former and only train the latter. . . . 65

5.4. Visualization of CNN features by averaging over the channel dimension of convolution feature maps [130]. (a) shows the image ground truth relationships, (b) shows the convolution feature from the entity detector backbone, and (c) shows the feature from the predicate backbone. In all the three examples there are clear shifts of salience from large entities to small areas that strongly indicate the predicates (highlighted in white boxes). 67

5.5. Example results of RelDN with L_0 only and with our losses. The top row shows RelDN outputs and the bottom row visualizes the learned predicate CNN features of the two models. Red and green boxes highlight the wrong and right outputs (the first row) or feature saliency (the second row). As it shows, our losses force the model to attend to the representative regions that discriminate the correct relationships against unrelated entity pairs, thus is able to disentangle entity instance confusion and proximal relationship ambiguity. 70

5.6. Example images where RelDN with only L_0 predicts incorrectly while our loss succeeds. For each image we check the number of its ground truth relationships, then we output the same number of top predictions from a model to see its ranking accuracy. Red boxes in (b) highlight the false predictions from RelDN with L_0 only and green boxes in (c) highlight the correct ones from RelDN with all losses. 76

5.7. Example images of the 100 image subset with ground truth relationships. The subset contains five predicates where the Entity Instance Confusion and Proximal Relationship Ambiguity commonly occur. 77

6.1. Illustration of how fine-granular features may help in VSQA. Character names, visual objects, and their relationships are all necessary factors in answering this question. Character-aware relationships are detected in video frames, where references to humans such as “woman” and “man” are replaced with predicted character names, determined by finding the face bounding box that overlaps the most with the human bounding box. 79

6.2. An illustration of our weakly-supervised character identification pipeline. The face bounding boxes of all characters are first detected. The extracted face features are then predicted by fully-connected feed-forward layer and Softmax. After broadcasting the character names in subtitle to be a distribution sequence that has the same length of predicted name distribution sequence, a weight KL divergence loss is utilized to conduct multi-instance co-occurrence matching. . 82

6.3. Examples of correctly answered questions that benefit from the proposed strategy. Orange and blue boxes are subjects and objects, while white boxes are objects with no detected relationships. Boxes with names are our detected characters, which substitute for the human-referring words in the relationships to obtain a character-aware understanding. 89

8.1. Illustration of previous common pipeline and the potential future model. 95

Chapter 1

Introduction

1.1 Relationship Proposal and Scene Graph Parsing

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 these 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.

1.2 Ambiguities in Scene Graph Parsing

The task of Scene Graph Parsing aims at parsing the given image into a graph whose nodes are objects of interest and edges are meaningful pairwise relationships. Even with the aforementioned proposal method utilized, there could still be scenarios where two major ambiguities happen that might significantly harm the task. Concretely, this thesis studies two cases:

1.2.1 Semantic Ambiguity

It is a common practice in academia to restrict the number of object and predicate categories when doing the scene graph parsing. However, in real application the vocabulary could be very large or even totally open. In that case, ambiguity might exist among these categories, i.e., the model is not 100% sure which category to output since there are several candidates with very similar semantic meanings. In this thesis we propose to use learnable embeddings to solve this issue. The embeddings are learned in a way that under the context of subjects and objects, the model learns to not only discriminate rights against wrongs, but also preserve semantic similarities between categories.

1.2.2 Visual Ambiguity

Visual ambiguity happens in a more common setting since it is an intrinsic issue of scene graphs. Concretely there are two major visual ambiguities: 1) Entity Instance Confusion: the subject or object is related to one of many instances of the same class, and the model fails to distinguish between the target instance and the others; 2) Proximal Relationship Ambiguity: it occurs when the image contains multiple subject-object pairs interacting in the same way, and the model fails to identify the correct pairing. The primary cause of these two failures lies in the inherent difficulty of inferring relationships. It is challenging for any model to learn to attend to these details precisely, and it would be impractical to specify which details to focus on for all kinds of relationships, let alone to learn all these details. These challenges motivate the need for a mechanism that can automatically learn fine details that determine visual relationships, and explicitly discriminate related entities from unrelated ones, for all types of relationships. This is one of the goals of this thesis.

1.3 Visual-Semantic Understanding via Scene Graphs

Visual Semantic Understanding is a task that given both visual and textual signals, the machine needs to learn to understand the association between them and complete the required mission, such as question answering or image-to-text matching. In order to do it accurately and reasonably, it is critical for a model to ground visual and textual signals into solid elements as well

as figuring out how these elements are composed and related. This is where the motivation of scene graphs lies in, since scene graphs are a grounded and associated representation for salient elements in the given visual signals. In this thesis we explore one of the visual-semantic understanding tasks, i.e., video story understanding, by leveraging scene graphs as richer representation of the input video, and analyze how and why it could help on comprehending the video and answering the given questions.

Chapter 2

Background and Related Work

2.1 Visual Relationships

Visual relationships [62, 90] 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 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 one of the major motivations of this thesis.

2.2 Object Proposals

Object proposal methods can be generally classified into two types: unsupervised approaches, including super-pixel merging [101, 10, 4] and objectness evaluation [2, 147], and supervised region prediction based on learned deep features from CNNs [87, 47, 9]. 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 [18, 85, 61].

2.3 Object Relationship Exploration

There is significant literature that explores relationships between multiple objects, including object co-occurrence [70, 91, 49] and semantic segmentation [30, 94]. Spatial relationships have also been studied to improve both object-level and pixel-precision tasks [23, 30]. 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 [89, 120, 67] has been a well-studied area where the "subject" is a human and "predicate" is a verb. In this thesis, we study general relationships with different types, where the"subject" and "predicate" are not constrained.

2.4 Visual Relationship Detection

In [90], the concept of visual phrases is introduced to represent relationship tuples. In [62], 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 [16], a classification-free approach is proposed for visual relationship recognition, but it does not localize the objects in the predicted relationship.

Some object detection methods [61, 93, 85] have removed the object proposal step and directly output detection boxes with labels. However, 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 [61]). 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.

Looking back at more recent literature, almost all of the relationship detectors are built for small vocabularies, e.g., 100 object and 70 relation categories from the VRD dataset [62], or a subset of VG with the most frequent object and relation categories [134, 114, 140, 138, 141]. In one of the earliest works, [62] utilize the object detection output of an an R-CNN detector and leverage language priors from semantic word embeddings to fine-tune the likelihood of a predicted relationship. Very recently, [146] use language representations of the subject and object as "context" to derive a better classification result for the relation. However, similar to [62] their language representations are pre-trained. Unlike these approach, we fine-tune subject and object representations *jointly* and employ the interaction between branches also at an earlier stage before classification.

In [128], the authors employ knowledge distillation from a large Wikipedia-based corpus and get state-of-the-art results for the VRD [62] dataset. In ViP-CNN [55], the authors pose the problem as a classification task on limited classes and therefore cannot scale to the open-vocabulary scenarios. In this thesis we exploit co-occurrences at the relationship level to model such knowledge. Our approach directly targets the large category scale and is able to utilize semantic associations to compensate for infrequent classes, while at the same time achieves competitive performance in the smaller and constrained VRD [62] dataset.

Approaches like [144, 84] target open-vocabulary for scene parsing and visual relationship detection, respectively. In [84], the related work closest to ours, the authors learn a CCA model on top of different combinations of the subject, object and union regions and train a Rank SVM. They however consider each relationship triplet as a class and learn it as a whole entity,

thus cannot scale to our setting. Our approach embeds the three components of a relationship separately to the independent semantic spaces for object and relation, but implicitly learns connections between them via visual feature fusion and semantic meaning preservation in the embedding space.

2.5 Semantically Guided Visual Recognition

Another parallel category of vision and language tasks is known as zero-shot/few-shot, where class imbalance is a primary assumption. In [21], [78] and [96], word embedding language models (e.g., [71]) were adopted to represent class names as vectors and hence allow zero-shot recognition. For fine-grained objects like birds and flowers, several works adopted Wikipedia Articles to guide zero-shot/few-shot recognition[51, 17]. However, for relations and actions, these methods are not designed with the capability of locating the objects or interacting objects for visual relations. Several approaches have been proposed to model the visual-semantic embedding in the context of the image-sentence similarity task (e.g., [46, 19, 106, 28]). Most of them focused on leaning semantic connections between the two modalities, which we not only aim to achieve, but with a manner that does not sacrifice discriminative capability since our task is detection instead of similarity-based retrieval. In contrast, visual relationship also has a structure of $\langle$subject, relation, object$\rangle$ and we show in our results that proper design of a visual-semantic embedding architecture and loss is critical for good performance.

2.6 Scene Graph Parsing

A scene graph is defined as a graph where nodes are objects with their attributes and edges are relationships between objects. The task of scene graph parsing is to extract the scene graph from the given image. Recent scene graph parsers use the same pipeline that first either uses off-the-shelf detectors [62, 146, 134, 13, 128, 117] or detectors fine-tuned with relationship datasets [55, 114, 132, 136, 137, 122, 115] to detect entities, then predicts the predicate using proposed methods. Most of them [62, 146, 134, 13, 128, 122, 55, 114, 132, 136, 139, 140] model the second step as a classification task that takes features of each entity pair as input and output a label independently from other pairs. [137] instead learn embeddings for subjects,

predicates and objects and use nearest neighbor searching during testing to predict predicates. Nevertheless, the prediction is still done on each entity pair individually. We show that this pipeline struggles with two major scenarios. We find that ignoring the intrinsic graph structure of relationships and predicting each predicate separately is the main cause. Our proposed losses compensate for such drawback by contrasting positive against negative edges for each node, providing global supervision to the classifier and significantly alleviating those two issues.

The scene graph parsing work most related to ours is Associative Embedding [76]. They use use a *push* and *pull* contrastive loss to train embeddings for entities within a visual genome scene graph. Our work differs in that we propose to have different sets of hard negatives to target specific error types within scene graph parsing.

2.7 Phrase Grounding and Referring Expressions

Phrase grounding and referring expression models aim to localize the region described by a given expression, with the latter focusing more on cases of possible reference confusion [126, 69, 127, 74, 35, 65, 88, 107, 60, 11, 34, 83]. It can be abstracted as a bipartite graph matching problem, where nodes on the visual side are the regions and nodes on the language side are the expressions, and the goal is to find all matched pairs. In contrast, scene graphs are arbitrarily connected graphs whose nodes are visual entities and edges are predicates with rich semantic information. Our losses are designed to leverage that information to better discriminate between related and non-related entities.

2.8 Contrastive Training

Contrastive training using a triplet loss [46] has wide application in both computer vision and natural language processing. Representative works include Negative Sampling [71] and Noise Contrastive Sampling [72]. More recent work also utilizes it to solve multi-modal tasks such as phrase grounding, image captioning, VQA, and vector embeddings [107, 32, 126, 76]. Our setting differs in that we define hard negative contrastive margins along the known structure of the annotated scene graph, allowing us to specifically target entity instance and proximal relationship confusion. By adding our losses as additional supervision on top of the N-way

cross-entropy loss, we are able to improve the model by significant margins.

2.9 Visual Question Answering (VQA)

The current dominant framework for VQA systems consists of an image encoder, a question encoder, multimodal fusion, and an answer predictor. In lieu of directly using visual features from CNN-based feature extractors, [118, 20, 80, 63, 99, 75, 145, 68] explored various image attention mechanisms to locate regions that are relevant to the question. To learn a better representation of the question, [63, 75, 20] proposed to perform question-guided image attention and image-guided question attention collaboratively, to merge knowledge from both visual and textual modalities in the encoding stage. [22, 43, 129, 7, 42] explored higher order fusion methods to better combine textual information with visual information (e.g., using bilinear pooling instead of simpler first-order methods such as summation, concatenation and multiplication).

To make the model more interpretable, some literature [53, 124, 52, 110, 111, 109] also exploited high-level semantic information in the image, such as attributes, captions and visual relation facts. Most of these methods applied VQA independent models to extract semantic knowledge from the image, while [64] built a Relation-VQA dataset and directly mined VQA-specific relation facts to feed additional semantic information to the model. A few recent studies [97, 66, 52] investigated how to incorporate memory to aid the reasoning step, especially for difficult questions. However, the semantic knowledge brought in by either memory or high-level semantic information is usually converted into textual representation, instead of directly used as visual representation, which contains richer and more indicative information about the image. Our work is complementary to these prior studies in that we encode object relations directly into image representation, and the relation encoding step is generic and can be naturally fit into any state-of-the-art VQA model.

2.10 Relational Reasoning

We name the visual relationship aforementioned as explicit relation, which has been shown to be effective for image captioning [121]. Specifically, [121] exploited pre-defined semantic relations learned from the Visual Genome dataset [48] and spatial relations between objects.

A graph was then constructed based on these relations, and a Graph Convolutional Network (GCN) [45] was used to learn representations for each object.

Another line of research focuses on implicit relations, where no explicit semantic or spatial relations are used to construct the graph. Instead, all the relations are implicitly captured by an attention module or via higher-order methods over the fully-connected graph of an input image [92, 33, 8, 119], to model the interactions between detected objects. For example, [92] reasons over all the possible pairs of objects in an image via the use of simple MLPs. In [8], a bilinear fusion method, called MuRel cell, was introduced to perform pairwise relationship modeling.

Some other work [100, 77, 108] have been proposed for learning question-conditioned graph representations for images. Specifically, [77] introduced a graph learner module that is conditioned on question representations to compute the image representations using pairwise attention and spatial graph convolutions. [100] exploited structured question representations such as parse trees, and used GRU to model contextualized interactions between both objects and words. A more recent work [108] introduced a sparser graph defined by inter/intra-class edges, in which relationships are implicitly learned via a language-guided graph attention mechanism. However, all these work still focused on implicit relations, which are less interpretable than explicit relations.

2.11 Video Story Question Answering

The task of video question answering has been explored in many recent studies. While some of them [54, 24, 36] principally focus on factual understanding in short videos, another research direction aims at understanding videos that contain story-lines and answering questions about them. Read Write Memory Networks (RWMN) [73] rely on Compact Bilinear Pooling to fuse individual captions with corresponding frames and store them in memory slots. Multi-layered CNNs are then employed to represent adjacent slots in time. PAMN [41] proposes a progressive attention memory to progressively prune out irrelevant temporal parts in memory and utilizes dynamic modality fusion to adaptively determine the contribution of each modality for answering questions. ES-MTL [40] introduces additional temporal retrieval and modality alignment networks to predict the time when the question was generated and to find associations of video

and subtitles. However, these methods merely extract visual features from video frames or parts of video frames with pre-trained CNNs while ignoring the characters inside video scenes, making their models lack the ability of deep scene understanding.

2.12 Visual Relation Detection

Visual relation detection has recently emerged as a task that goes one step further than object detection towards a holistic semantic understanding of images [62, 48, 114, **?**]. The task involves first detecting any visually related pairs of objects and recognizing the predicate that describes their relations. Most recent approaches achieve this goal by learning classifiers that predict relations based on different types of features of the object pairs [114, 128, 132, 137, 141]. It has been demonstrated in recent works that scene graphs can provide rich knowledge of image semantics and help boost high-level tasks such as Image Captioning and Visual Question Answering [121, 57, 119]. We are interested in how relations can be exploited not just for images but also for video understanding with a character-based relation representation, which to the best of our knowledge has not been fully explored yet.

2.13 Character Naming

The goal of character naming is to automatically identify characters in TV shows or movies. Previous methods tend to train a face assignment model based on extracted face tracklets. Some approaches rely on semi-supervised learning for person identification [98, 79, 5]. Meanwhile, [38] propose an unsupervised method to address the task. In this work, we train the character naming and question answering modules in a multi-task scheme. Our approach does not require any explicit annotations on faces. We only rely on weak supervision from the subtitles that contain speakers' names and exploit the co-occurrence distribution between appearing faces and names in subtitles.

Chapter 3

Relationship Proposal Networks

3.1 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 [62, 90] 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 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 [62] 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 [101], $\sim$2000 proposals are used while the number is reduced to 1000 in [147]. In [87], 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. Recently, the Visual Genome dataset [48] 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 chapter, 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.

3.2 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. We later show that there is a gain when the probability of a region is a subject (we call this subjectness); 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 section3.2.1. 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

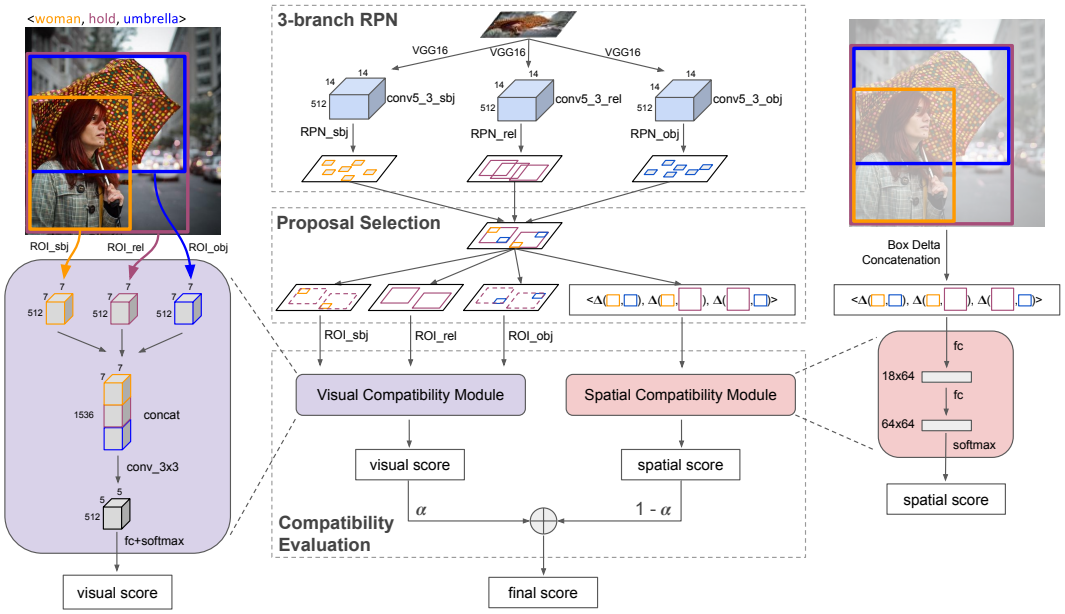


Figure 3.1: **Relationship Proposal Network architecture.** "sbj", "obj" and "rel" are abbreviations for "subject", "object" and "relationship". We feed an input image to a 3-branch RPN where each branch produces a set of candidate boxes. **Orange**, **purple**, **blue** boxes are subject, relationship 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.

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 3.1.

3.2.1 3-branch RPN

We use the Region Proposal Networks (RPN) in Faster RCNN [87] 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 3.1). 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.

3.2.2 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) relationship 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);

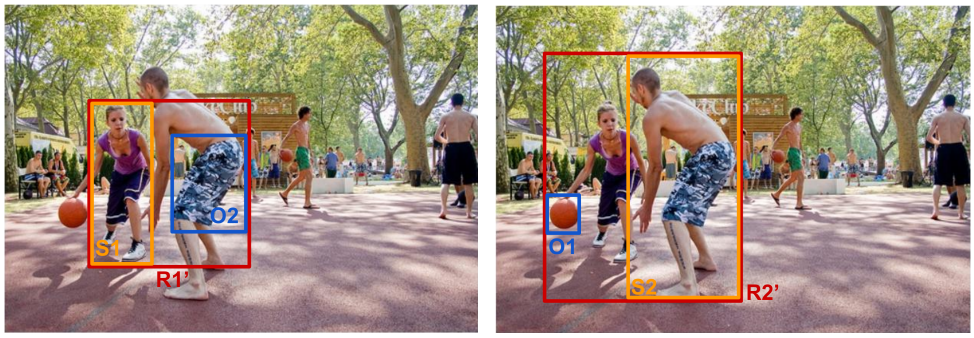
3. **Select qualified pairs:** For each of the $K_{sbj} \times K_{obj}$ subject-object pairs, we check whether its union box overlaps with the current relationship box by a threshold (0.5 in our model), and keep it only if this condition is satisfied; we also consider an additional set of K_{sbj} pairs 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 ⟨kite, in, sky⟩ and ⟨window, of, building⟩. 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 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

(a) Two positive samples



(b) Two negative samples

Figure 3.2: **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.

pairs against mismatched ones and learn the visual connection between subjects and objects in positive pairs. The sampling strategy is illustrated in Figure 3.2.

3.2.3 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 3.1). 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 [25] 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

$$\begin{aligned} t_x^{SO} &= (x^S - x^O)/w^S, \quad t_y^{SO} = (y^S - y^O)/h^S, \\ t_w^{SO} &= log(w^S/w^O), \quad t_h^{SO} = log(h^S/h^O), \\ t_x^{OS} &= (x^O - x^S)/w^O, \quad t_y^{OS} = (y^O - y^S)/h^O, \end{aligned} \tag{3.1}$$

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

$$p = \alpha p_v + (1 - \alpha) p_s \tag{3.2}$$

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 3.4.2.

3.3 Implementation Details

We utilize Region Proposals Network (RPN) [87] framework to build our 3-branch RPN module, and implemented proposal selection and compatibility evaluation module by our own using python layer from Caffe library [37]. Our model uses pre-trained VGG16 [95] weights for convolutional layers, and initializes other layers randomly by "xavier" algorithm [27]. For proposal selection, we consider the top $5,000$ union boxes by confidence scores generated from the relationship branch, then for each of them, we enlarge it by 1.1 to get a search region, and pick the top 9 subject boxes and top 9 object boxes that are within this search region, resulting in 81 subject-object pairs. Then we eliminate those pairs that do not overlap with the current union box by at least 0.5. As mentioned in section 3.2 in the paper, we also pair each of the 9 subject boxes with the current union box. Therefore, the resulting set contains both subject-object pairs and subject-union pairs. We pick the top $15,000$ from this set while ensuring that 30% of them are subject-union pairs, since this is the ratio of such pairs in the training data. Note that these $15,000$ pairs are to be evaluated on compatibility and are not the final relationship proposals.

At train-time, we merge the $15,000$ boxes with ground-truth boxes and sample 256 out of them with 64 positive and 192 negative pairs, where half of the negative samples have overlaps of less than 0.5 for either subject or object (condition 1 and 2 in section 3.2 in the paper), and the other half overlap with ground-truth by at least 0.7 for both subjects and objects but are simply mismatched (condition 3 in section 3.2 in the paper). In this way, we enforce the model to learn all potential types of negatives.

At test-time, we feed all the $15,000$ proposals to ROI pooling layer by 2 mini-batches

implemented as 2 forward passes of $7,500$ proposals since GPU memory is insufficient for one single pass. Once we get compatibility scores for each of these proposals, we select the top N_{rel} as the final relationship proposals where N_{rel} varies in our experiments.

It is worth mentioning that at train-time, we do not back-propagate either visual or spatial compatibility loss to the preceding convolution layers (i.e., the 3-branch RPN module), because those convolution layers are supposed to learn to generate subject, object and relationship boxes independently without considering connections between boxes, so we should not blame them for the loss from some incompatible box pairs. For example, if a subject box and an object box are perfectly located, but they do not form a meaningful pair (i.e., there is no relationship between them), then each of them should still be a positive sample for the subject and object branch, but this pair of boxes is a negative sample for the compatibility module. Therefore, we disable back-propagation of compatibility loss to the 3-branch RPN module in order to eliminate such confusion.

3.4 Experiments

We evaluate our model by localizing relationships in images. To our best knowledge we are the first to study relationship 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 [48] and Visual Relationship Detection (VRD) dataset [62].

3.4.1 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

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 3.1: **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.

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 3.2: **Recall rates on VG with IoU≥0.5**. Abbreviations are the same with Table 3.1.

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) [101], EdgeBoxes (EB) [147] and Region Proposal Network (RPN) [87]. 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.

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

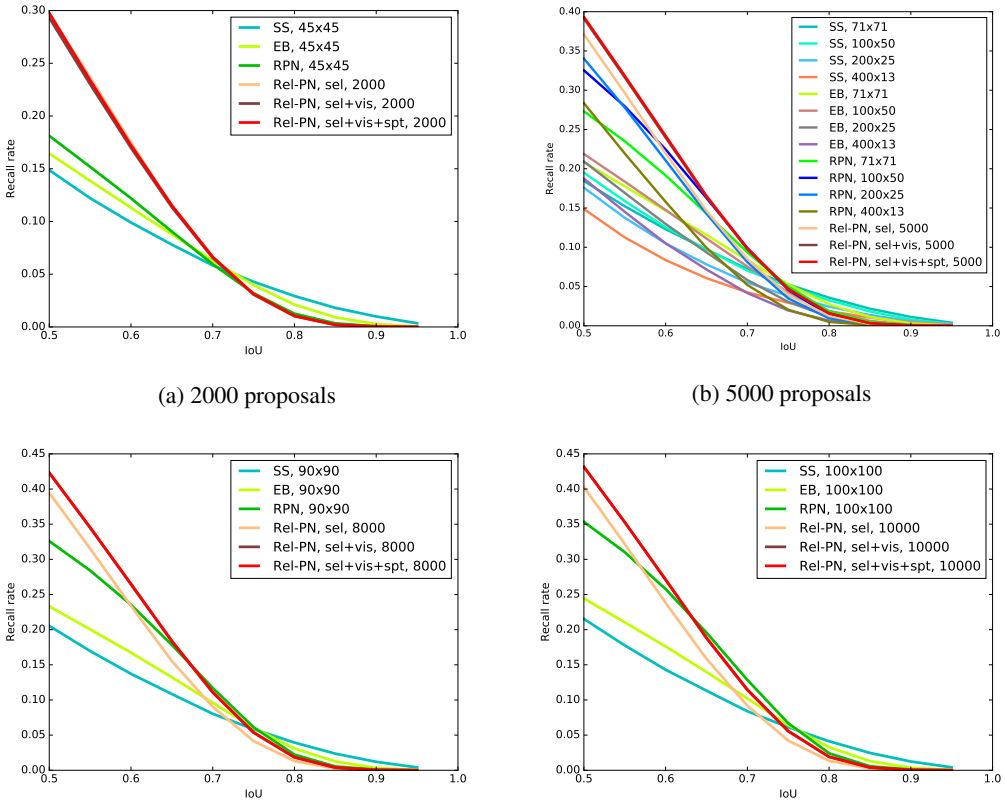
- **Proposal Selection.** We select top N proposals by the average of subjectness and objectness scores from the proposal selection module without feeding it to the compatibility module.
- **Proposal Selection + Spatial Compatibility.** We use only spatial confidence scores for the final proposals.
- **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 3.2.3.

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$\geq$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.

3.4.2 Visual Genome

The Visual Genome dataset (VG) contains $108,077$ images with 21 relationships on average per image. Each relationship is of the form $\langle$subject, predicate, object$\rangle$ with annotated subject and object bounding boxes. We follow [39] and split the data into $103,077$ training images and



(a) 2000 proposals

(b) 5000 proposals

(c) 8000 proposals

(d) 10000 proposals

Figure 3.3: **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.

$5,000$ 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 3.1, while the second experiment is reported in Table 3.2. We also show Recall vs IoU curves with 2000, 5000, 8,000 and 10,000 proposals in Figure 3.3. We make the following observations:

- Table 3.1 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 3.1) 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.

(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)

Figure 3.4: **Example relationship proposals on VG.** Red and blue boxes are ground-truth subject and object, yellow and green boxes are outputs from our model.

- The visual compatibility is clearly more important than spatial. Using only visual compatibility can lead to a sub-optimal 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

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

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 3.3, 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 [87]), 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≥0.7).

Qualitative Results. In Figure 3.4, 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.

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 3.4: **Recall rates on VRD with IoU≥0.5.**

Visual Compatibility Weight. In Table 3.3, 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. 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").

3.4.3 Visual Relationship Detection dataset

In this section we conduct experiments on the Visual Relationship dataset (VRD) from [62]. We use the same settings with the Visual Genome experiments. In Table 3.4 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 than Visual Genome (e.g., 44% for $8,000$ proposals and 46% 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.

We also report results of various methods on Visual relationship Detection (VRD) dataset in Table 3.5. We notice that our model achieves better recall rates on VRD compared to Visual Genome (VG) with all the three IoU thresholds (see Table 1 in the paper). We attribute this phenomenon to two factors: 1) The VRD dataset is cleaner than VG in that subjects and objects are located more accurately and annotated with less ambiguity. For example, for some relationships in VG where the objects are large areas, such as ⟨sheep, eat, grass⟩, the bounding box for "grass" might be a small rectangle containing grass near the feet of the sheep, instead

5000 proposals	IoU≥0.5	IoU≥0.6	IoU≥0.7
SS, pairwise, 71×71	28	18.8	10
SS, nns, 100×50	29.1	18.7	9.3
SS, nns, 200×25	21	12	5.2
SS, nns, 400×13	14.3	7.9	3.3
EB, pairwise, 71×71	20.6	13.8	7.9
EB, nns, 100×50	21.6	14.7	7.4
EB, nns, 200×25	20.2	11.6	5.4
EB, nns, 400×13	18	9.5	3.4
RPN, pairwise, 71×71	28.4	20.8	10.3
RPN, nns, 100×50	33.5	23.3	10.1
RPN, nns, 200×25	31.4	18.9	7
RPN, nns, 400×13	24.9	13.8	4.1
Rel-PN, pro_sel	41.9	24.3	8.9
Rel-PN, pro_sel + spt	38.6	21.2	7.7
Rel-PN, pro_sel + vis	42.9	25.2	9.6
Rel-PN, pro_sel + vis + spt	**44.3**	**26.6**	**10.6**

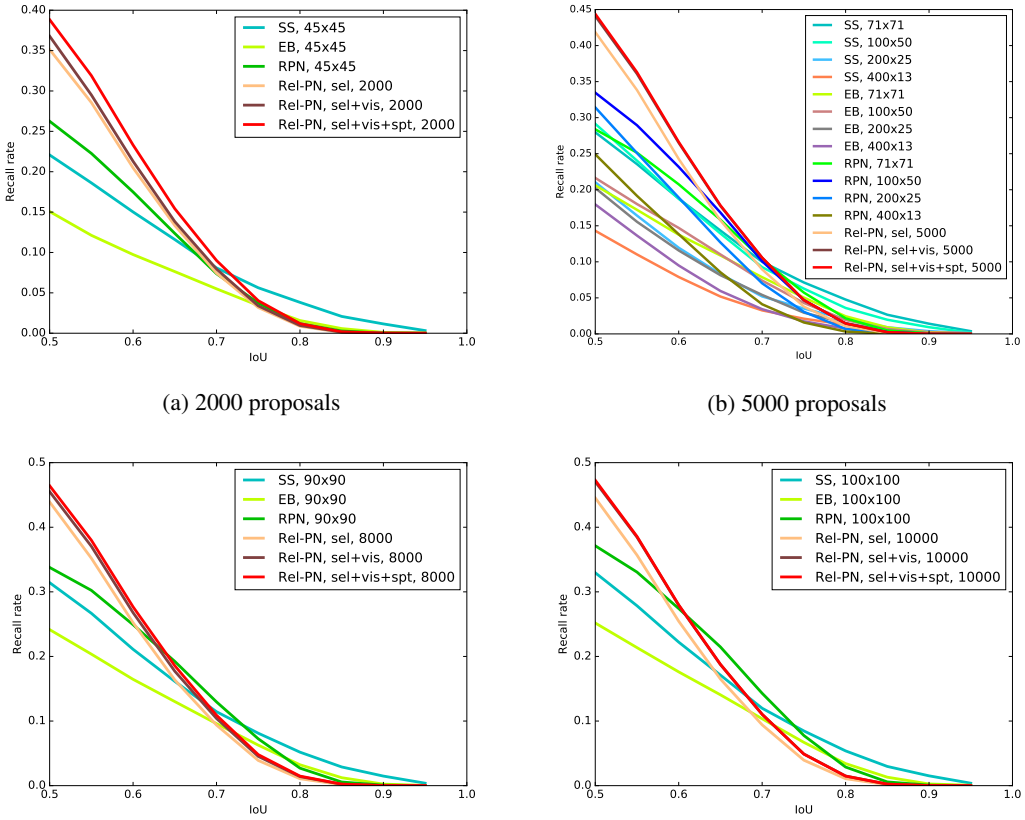
Table 3.5: **Recall rates on VRD by 5000 proposals.** Abbreviations are the same with Table 1 in the paper.

of the whole meadow where the sheep stands in. Such inaccuracy of annotation could confuse the relationship proposer at train-time when it occurs in training data, and cause misjudgment during test-time when it is in testing data. In contrast, this is barely the case in the VRD dataset; 2) VRD has only 100 object categories and 70 predicate types compared to 33,877 object categories and 40,480 unique relationships in VG, which also leads to a more limited and biased space of relationship types than VG. Hence, not only our 3-branch RPN module has less burden on individually generating initial subject and object candidates, but also our compatibility module can learn to discriminate relationships from non-relationships more easily.

We also illustrate Recall vs IoU curves in Figure 3.5. Similar with VG, the gap between our model and baselines is larger when the number of proposals is smaller and the IoU threshold is smaller. Again, it is mainly due to the small size (7×7) of ROI-pooled features. Potential solutions include enlarging this feature size by using dilation convolutions [142, 12, 125] or by using deconvolutions [131].

3.5 More Qualitative Results on VG

We show more example relationship proposals from VG dataset in Figure 3.6. We fix the number of proposals as 5,000 for all examples. For each image, the 5,000 proposals are ranked



(a) 2000 proposals

(b) 5000 proposals

(c) 8000 proposals

(d) 10000 proposals

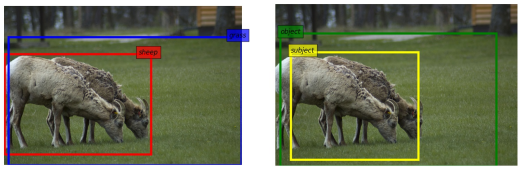
Figure 3.5: **Recall vs IoU on VRD with various numbers of proposals.**

by IoUs with ground-truth, and the shown proposal is selected from the top 17 with the best illustrative effect.

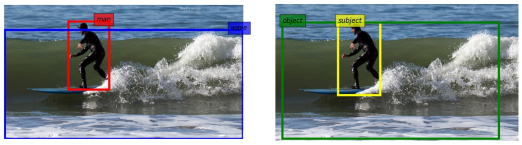
3.6 Limitations

Although our model is able to localize meaningful relationships, these relationships might not be significantly connected to the underlying scenes. For example, Figure 3.6m is mainly about a man lying beside a dog, so the most important relationship should be ⟨man, beside, dog⟩. Our model did localize this relationship as shown in Figure 3.6m, but it localized other trivial relationships with higher accuracy. In fact, if we rank all the 5,000 proposals by IoUs with ground-truth, the proposal for ⟨man, beside, dog⟩ is ranked only at 14th, while other minor proposals with higher ranks include ⟨head, on, pillow⟩ at 1st (Figure 3.7a), ⟨man, has, head⟩ at 2nd (Figure 3.7b), ⟨teeth, in, mouth⟩ at 9th (Figure 3.7c), and ⟨dog, in, bed⟩ at 10th (Figure

3.7d). Such failure of attending to the most interesting region in the scene is due to the lack of global context of the scene (i.e., the whole image), which provides information about how salient a relationship is given the image. A possible solution is to obtain context information by spatial Recurrent Neural Networks (RNNs) [6, 104] and integrate into our 3-branch RPN module so that subjects and objects with tighter connection to the background have higher subjectness and objectness scores.

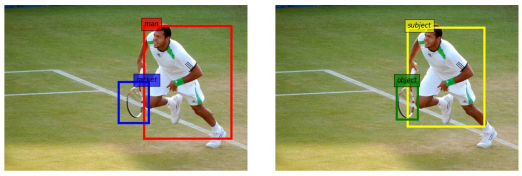


(a) ⟨sheep, eating, grass⟩



(b) ⟨man, surfing, wave⟩

(c) ⟨boy, with, frisbee⟩

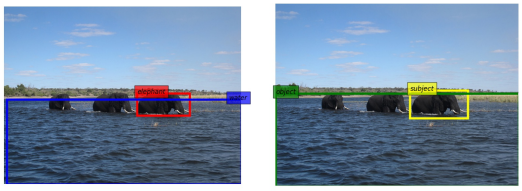


(d) ⟨man, with, racket⟩

(e) ⟨bride, with, groom⟩

(f) ⟨people, riding, elephant⟩

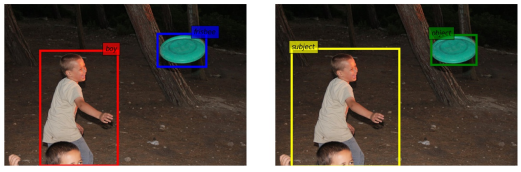
(g) ⟨surfer, on, board⟩



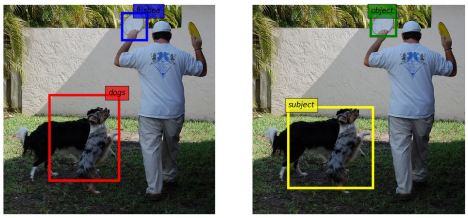
(h) ⟨elephant, in, water⟩

(i) ⟨man, on, a skateboard⟩

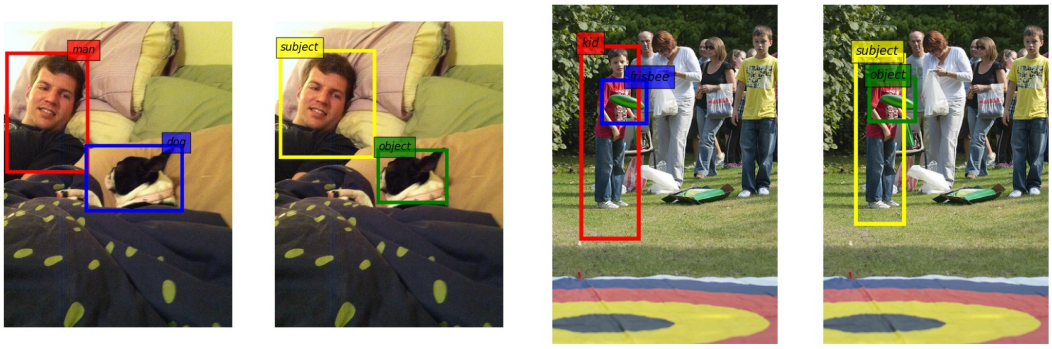
(j) ⟨person, on, horse⟩



(k) ⟨boy, playing, frisbee⟩

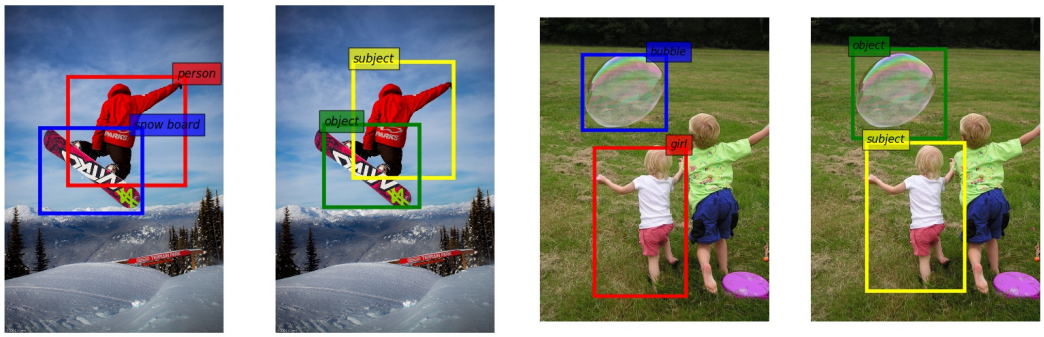


(l) ⟨dogs, want, frisbee⟩



(m) ⟨man, beside, dog⟩

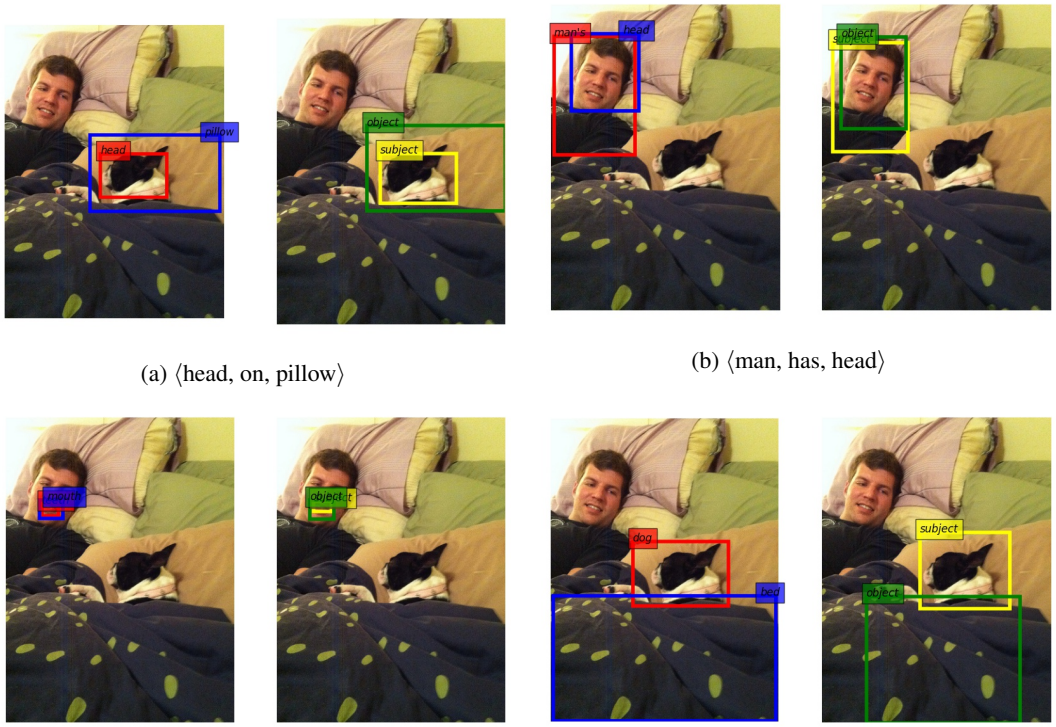
(n) ⟨kid, has, frisbee⟩



(o) ⟨person, on, snow board⟩

(p) ⟨girl, chasing, bubble⟩

Figure 3.6: **More example relationship proposals on VG.** Red and blue boxes are ground-truth subject and object, yellow and green boxes are outputs from our model.



(a) ⟨head, on, pillow⟩

(b) ⟨man, has, head⟩

(c) ⟨teeth, in, mouth⟩

(d) ⟨dog, in, bed⟩

Figure 3.7: **Illustration of our model's limitation.** These four output proposals are ranked prior to the one shown in Figure 3.6m, which is supposed to be ranked at the top according to human's perceptual intuition about this image, since it is mostly about a man lies beside a dog.

Chapter 4

Large-Scale Visual Relationship Understanding

Large scale visual understanding is challenging, as it requires a model to handle the widely-spread and imbalanced distribution of ⟨subject, relation, object⟩ triples. In real-world scenarios with large numbers of objects and relations, some are seen very commonly while others are barely seen. We develop a new relationship detection model that embeds objects and relations into two vector spaces where both discriminative capability and semantic affinity are preserved. We learn a visual and a semantic module that map features from the two modalities into a shared space, where matched pairs of features have to discriminate against those unmatched, but also maintain close distances to semantically similar ones. Benefiting from that, our model can achieve superior performance even when the visual entity categories scale up to more than $80,000$, with extremely skewed class distribution. We demonstrate the efficacy of our model on a large and imbalanced benchmark based of Visual Genome that comprises $53,000+$ objects and $29,000+$ relations, a scale at which no previous work has been evaluated at. We show superiority of our model over competitive baselines on the original Visual Genome dataset with $80,000+$ categories. We also show state-of-the-art performance on the VRD dataset and the scene graph dataset which is a subset of Visual Genome with 200 categories.

4.1 Introduction

Scale matters. In the real world, people tend to describe visual entities with open vocabulary, eg., the raw ImageNet [15] dataset has 21,841 synsets that cover a vast range of objects. The number of entities is significantly larger for relationships since the combinations of ⟨subject, relation, object⟩ are orders of magnitude more than objects [62, 84, 136]. Moreover, the long-tailed distribution of objects can be an obstacle for a model to learn all classes sufficiently well,

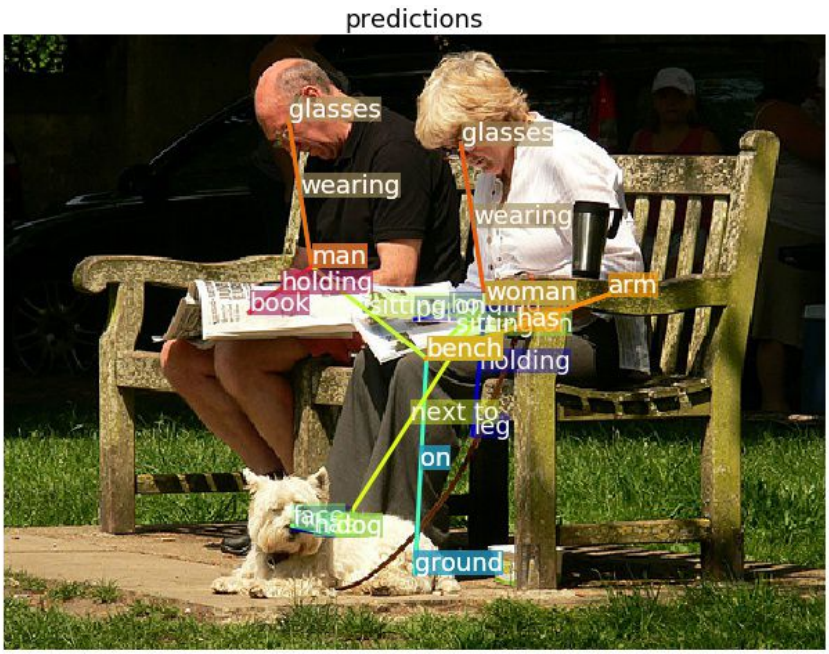


Figure 4.1: Relationships predicted by our approach on an image. Different relationships are colored differently with a relation line connecting each subject and object. Our model is able to recognize relationships composed of over $53,000$ object categories and over $29,000$ relation categories.

and such challenge is exacerbated in relationship detection because either the subject, the object, or the relation could be infrequent, or their triple might be jointly infrequent. Figure 5.1 shows an example from the Visual Genome dataset, which contains commonly seen relationship (e.g., ⟨man,wearing,glasses⟩) along with uncommon ones (e.g., ⟨dog,next to,woman⟩).

Another challenge is that object categories are often semantically associated [15, 48, 14], and such connections could be more subtle for relationships since they are conditioned on the contexts. For example, an image of ⟨person,ride,horse⟩ could look like one of ⟨person,ride,ele--phant⟩ since they both belong to the kind of relationships where a person is riding an animal, but ⟨person,ride,horse⟩ would look very different from ⟨person,walk with,horse⟩ even though they have the same subject and object. It is critical for a model to be able to leverage such conditional connections.

In this work, we study relationship recognition at an unprecedented scale where the total number of visual entities is more than 80,000. To achieve that we use a continuous output space for objects and relations instead of discrete labels. We demonstrate our superiority over

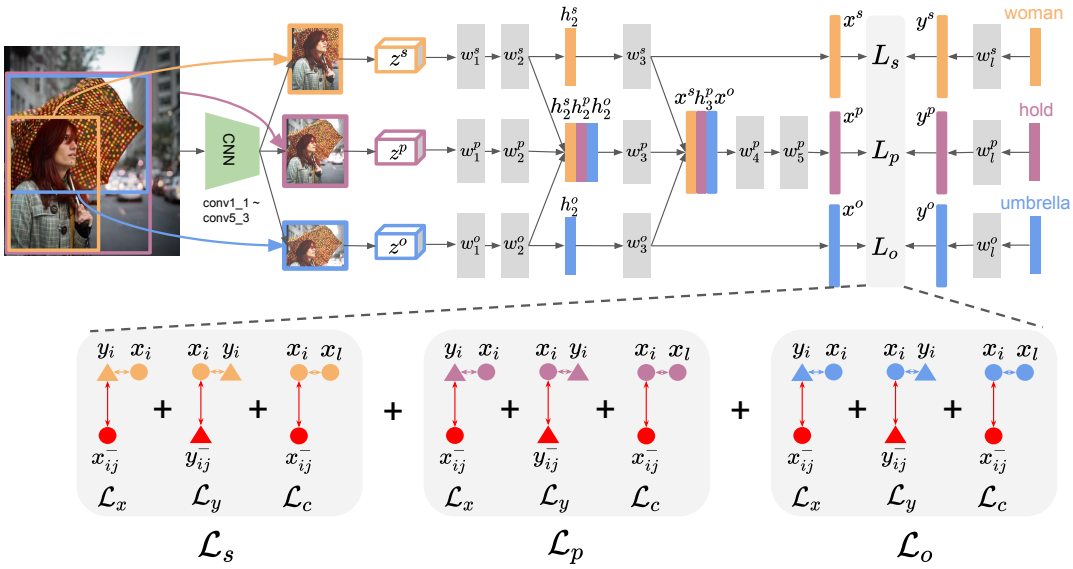


Figure 4.2: (a) Overview of the proposed approach. L_s, L_p, L_o are the losses of subject, relation and object. Orange, purple and blue colors represent subject, relation, object, respectively. Grey rectangles are fully connected layers, which are followed by ReLU activations except the last ones, i.e. w_3^s, w_5^p, w_3^o. We share layer weights of the subject and object branches, i.e. w_i^s and w_i^o, $i = 1, 2...5$.

competitive baselines on a large and imbalanced benchmark based of Visual Genome that comprises $53,000+$ objects and $29,000+$ relations. We also achieve state-of-the-art performance on the Visual Relationship Detection (VRD) dataset [62], and the scene graph dataset [114].

4.2 Method

Figure 4.2 shows the work flow of our model. We take an image as input to the visual module and output three visual embeddings x^s, x^p, and x^o for subject, relation, and object. During training we take word vectors of subject, relation, object as input to the semantic module and output three semantic embeddings y^s, y^p, y^o. We minimize the loss by matching the visual and semantic embeddings using our designed losses. During testing we feed word vectors of all objects and relations and use nearest neighbor searching to predict relationship labels. The following sections describe our model in details.

4.2.1 Visual Module

The design logic of our visual module is that a relation exists when its subject and object exist, but not vice versa. Namely, relation recognition is conditioned on subject and object, but object

recognition is independent from relations. The main reason is that we want to learn embeddings for subject and object in a separate semantic space from the relation space. That is, we want to learn a mapping from visual feature space (which is shared among subject/object and relation) to the two separate semantic embedding spaces (for objects and relations). Therefore, involving relation features for subject/object embeddings would have the risk of entangling the two spaces. Following this logic, as shown in Figure 4.2 an image is fed into a CNN ($conv1\_1$ to $conv5\_3$ of VGG16) to get a global feature map of the image, then the subject, relation and object features z^s, z^p, z^o are ROI-pooled with the corresponding regions $\mathcal{R}_\mathcal{S}$, $\mathcal{R}_\mathcal{P}$, $\mathcal{R}_\mathcal{O}$, each branch followed by two fully connected layers which output three intermediate hidden features h_2^s, h_2^p, h_2^o. For the subject/object branch, we add another fully connected layer w_3^s to get the visual embedding x^s, and similarly for the object branch to get x^o. For the relation branch, we apply a two-level feature fusion: we first concatenate the three hidden features h_2^s, h_2^p, h_2^o and feed it to a fully connected layer w_3^p to get a higher-level hidden feature h_3^p, then we concatenate the subject and object embeddings x^s and x^o with h_3^p and feed it to two fully connected layers w_4^p w_5^p to get the relation embedding x^p.

4.2.2 Semantic Module

On the semantic side, we feed word vectors of subject, relation and object labels into a small MLP of one or two fc layers which outputs the embeddings. As in the visual module, the subject and object branches share weights while the relation branch is independent. The purpose of this module is to map word vectors into an embedding space that is more discriminative than the raw word vector space while preserving semantic similarity. During training, we feed the ground-truth labels of each relationship triplet as well as labels of negative classes into the semantic module, as the following subsection describes; during testing, we feed the whole sets of object and relation labels into it for nearest neighbors searching among all the labels to get the top k as our prediction.

A good word vector representation for object/relation labels is critical as it provides proper initialization that is easy to fine-tune on. We consider the following word vectors:

Pre-trained word2vec embeddings (wiki). We rely on the pre-trained word embeddings provided by [71] which are widely used in prior work. We use this embedding as a baseline, and

show later that by combining with other embeddings we achieve better discriminative ability.

Relationship-level co-occurrence embeddings (relco). We train a skip-gram word2vec model that tries to maximize classification of a word based on another word in the same context. As is in our case we define context via our training set's relationships, we effectively learn to maximize the likelihoods of $P(P|S,O)$ as well as $P(S|P,O)$ and $P(O|S,P)$. Although maximizing $P(P|S,O)$ is directly optimized in [128], we achieve similar results by reducing it to a skip-gram model and enjoy the scalability of a word2vec approach.

Node2vec embeddings (node2vec). As the Visual Genome dataset further provides image-level relation graphs, we also experimented with training *node2vec* embeddings as in [31]. These are effectively also word2vec embeddings, but the context is determined by random walks on a graph. In this setting, nodes correspond to subjects, objects and relations from the training set and edges are directed from $S \rightarrow P$ and from $P \rightarrow O$ for every image-level graph. This embedding can be seen as an intermediate between image-level and relationship level co-occurrences, with proximity to the one or the other controlled via the length of the random walks.

4.2.3 Training Loss

To learn the joint visual and semantic embedding we employ a modified triplet loss. Traditional triplet loss [46] encourages matched embeddings from the two modalities to be closer than the mismatched ones by a fixed margin, while our version tries to maximize this margin in a softmax form. In this subsection we review the traditional triplet loss and then introduce our triplet-softmax loss in a comparable fashion. To this end, we denote the two sets of triplets for each positive visual-semantic pair by $(\mathbf{x}^l, \mathbf{y}^l)$:

$$tri^l_{\mathbf{x}} = \{\mathbf{x}^l, \mathbf{y}^l, \mathbf{x}^{l-}\} \tag{4.1}$$

$$tri^l_{\mathbf{y}} = \{\mathbf{x}^l, \mathbf{y}^l, \mathbf{y}^{l-}\} \tag{4.2}$$

where $l \in \{s, p, o\}$, and the two sets $tri_{\mathbf{x}}, tri_{\mathbf{y}}$ correspond to triplets with negatives from the visual and semantic space, respectively.

Triplet loss. If we omit the superscripts $\{s, p, o\}$ for clarity, the triplet loss $\mathcal{L}^{Tr}$ for each branch is summation of two losses $\mathcal{L}^{Tr}_{\mathbf{x}}$ and $\mathcal{L}^{Tr}_{\mathbf{y}}$:

$$\mathcal{L}^{Tr}_{\mathbf{x}} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\max[0, m + s(\mathbf{y}_i, \mathbf{x}^-_{ij}) - s(\mathbf{y}_i, \mathbf{x}_i)] \tag{4.3}$$

$$\mathcal{L}^{Tr}_{\mathbf{y}} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\max[0, m + s(\mathbf{x}_i, \mathbf{y}^-_{ij}) - s(\mathbf{x}_i, \mathbf{y}_i)] \tag{4.4}$$

$$\mathcal{L}^{Tr} = \mathcal{L}^{Tr}_{\mathbf{x}} + \mathcal{L}^{Tr}_{\mathbf{y}} \tag{4.5}$$

where N is the number of positive ROIs, K is the number of negative samples *per positive* ROI, m is the margin between the distances of positive and negative pairs, and $s(\cdot, \cdot)$ is a similarity function.

We can observe from Equation (3) that as long as the similarity between positive pairs is larger than that between negative ones by margin m, $[m + s(\mathbf{x}_i, \mathbf{x}^-_{ij}) - s(\mathbf{x}_i, \mathbf{y}_i)] \leq 0$, and thus $max(0, \cdot)$ will return zero for that part. That means, during training once the margin is pushed to be larger than m, the model will stop learning anything from that triplet. Therefore, it is highly likely to end up with an embedding space where points are not discriminative enough for a classification-oriented task.

It is worth noting that although theoretically traditional triplet loss can pushes the margin as much as possible when $m = 1$, most previous works (eg., [46, 19, 29]) adopted a small m to allow slackness during training. It is also unclear how to determine the exact value of m given a specific task. We follow previous works and set $m = 0.2$ in all of our experiments.

Triplet-Softmax loss. The issue of triplet loss mentioned above can be alleviated by applying softmax on top of each triplet, i.e.:

$$\mathcal{L}^{TrSm}_{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{e^{s(\mathbf{y}_i, \mathbf{x}_i)}}{e^{s(\mathbf{y}_i, \mathbf{x}_i)} + \sum_{j=1}^{K} e^{s(\mathbf{y}_i, \mathbf{x}^-_{ij})}} \tag{4.6}$$

$$\mathcal{L}^{TrSm}_{\mathbf{y}} = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{e^{s(\mathbf{x}_i, \mathbf{y}_i)}}{e^{s(\mathbf{x}_i, \mathbf{y}_i)} + \sum_{j=1}^{K} e^{s(\mathbf{x}_i, \mathbf{y}^-_{ij})}} \tag{4.7}$$

$$\mathcal{L}^{TrSm} = \mathcal{L}^{TrSm}_{\mathbf{x}} + \mathcal{L}^{TrSm}_{\mathbf{y}} \tag{4.8}$$

where $s(\cdot,\cdot)$ is the same similarity function (we use cosine similarity in this paper). All the other notations are the same as above. For each positive pair $(\mathbf{x}_i, \mathbf{y}_i)$ and its corresponding set of negative pairs $(\mathbf{x}_i, \mathbf{y}_{ij}^-)$, we calculate similarities between each of them and put them into a softmax layer followed by multi-class logistic loss so that the similarity of positive pairs would be pushed to be 1, and 0 otherwise. Compared to triplet loss, this loss always tries to enlarge the margin to its largest possible value (i.e., 1), thus has more discriminative power than the traditional triplet loss.

Visual Consistency loss. To further force the embeddings to be more discriminative, we add a loss that pulls closer the samples from the same category while pushes away those from different categories, i.e.:

$$\mathcal{L}_c = \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\max[0, m + s(\mathbf{x}_i, \mathbf{x}_{ij}^-) - \min_{l\in\mathcal{C}(i)} s(\mathbf{x}_i, \mathbf{x}_l)] \tag{4.9}$$

where N is the number of positive ROIs, $\mathcal{C}(l)$ is the set of positive ROIs in the same class of $\mathbf{x}_i$, K is the number of negative samples *per positive* ROI and m is the margin between the distances of positive and negative pairs. The interpretation of this loss is: the minimum similarity between samples from the same class should be larger than any similarity between samples from different classes by a margin. Here we utilize the traditional triplet loss format since we want to introduce slackness between visual embeddings to prevent embeddings from collapsing to the class centers.

Empirically we found it the best to use triplet-softmax loss for $\mathcal{L}_\mathbf{y}$ while using triplet loss for $\mathcal{L}_\mathbf{x}$. The reason is similar with that of the visual consistency loss: mode collapse should be prevented by introducing slackness. On the other hand, there is no such issue for y since each label y is a mode by itself, and we encourage all modes of y to be separated from each other. In conclusion, our final loss is:

$$\mathcal{L} = \mathcal{L}_\mathbf{y}^{TrSm} + \alpha\mathcal{L}_\mathbf{x}^{Tr} + \beta\mathcal{L}_c \tag{4.10}$$

where we found that $\alpha = \beta = 1$ works reasonably well for all scenarios.

Implementation details. For all the three datasets, we train our model for 7 epochs using 8

GPUs. We set learning rate as 0.001 for the first 5 epochs and 0.0001 for the rest 2 epochs. We initialize each branch with weights pre-trained on COCO [59]. For the word vectors, we used the `gensim` library [86] for both word2vec and node2vec [31]. For the triplet loss, we set $m = 0.2$ as the default value.

For the VRD and VG200 datasets, we need to predict whether a box pair has relationship, since unlike VG80k where we use ground-truth boxes, here we want to use general proposals that might contain non-relationships. In order for that, we add an additional "unknown" category to the relation categories. The word "unknown" is semantically dissimilar with any of the relations in these datasets, hence its word vector is far away from those relations' vectors.

There is a critical factor that significantly affects our triplet-softmax loss. Since we use cosine similarity, $s(\cdot, \cdot)$ is equivalent to dot product of two normalized vectors. We empirically found that simply feeding normalized vector could cause gradient vanishing problem, since gradients are divided by the norm of input vector when back-propagated. This is also observed in [6] where it is necessary to scale up normalized vectors for successful learning. Similar with [6], we set the scalar to a value that is close to the mean norm of the input vectors and multiply $s(\cdot, \cdot)$ before feeding to the softmax layer. We set the scalar to 3.2 for VG80k and 3.0 for VRD in all experiments.

ROI Sampling. One of the critical things that powers Fast-RCNN is the well-designed ROI sampling during training. It ensures that for most ground-truth boxes, each has 32 positive ROIs and $128 - 32 = 96$ negative ROIs, where positivity is defined as overlap IoU $>= 0.5$. In our setting, ROI sampling is similar for the subject/object branch, while for the relation branch, positivity is defined as both subject and object IoUs $>= 0.5$. Accordingly, we sample 64 subject ROIs with 32 unique positives and 32 unique negatives, and do the same thing for object ROIs. Then we pair all the 64 subject ROIs with 64 object ROIs to get 4096 ROI pairs as relationship candidates. For each candidate, if both ROIs' IoU $>= 0.5$ we mark it as positive, otherwise negative. We finally sample 32 positive and 96 negative relation candidates and use the union of each ROI pair as a relation ROI. In this way we end up with a consistent number of positive and negative ROIs for the relation branch.

4.3 Experiments

Datasets. We present experiments on three datasets, the original *Visual Genome* (VG80k) [48], the version of *Visual Genome* with 200 categories (VG200) [114], and *Visual Relationship Detection* (VRD) dataset [62].

- **VRD.** The VRD dataset [62] contains 5,000 images with 100 object categories and 70 relations. In total, VRD contains 37,993 relation annotations with 6,672 unique relations and 24.25 relationships per object category. We follow the same train/test split as in [62] to get 4,000 training images and 1,000 test images. We use this dataset to demonstrate that our model can work reasonably well on small dataset with small category space, even though it is designed for large-scale settings.
- **VG200.** We also train and evaluate our model on a subset of VG80k which is widely used in previous methods [114, 76, 133, 116]. There are totally 150 object categories and 50 predicate categories in this dataset. We use the same train/test splits as in [114]. Similarly with VRD, the purpose here is to show our model is also state-of-the-art in large-scale sample but small-scale category settings.
- **VG80k.** We use the latest version of Visual Genome (VG v1.4) [48] that contains $108,077$ images with 21 relationships on average per image. We follow [39] and split the data into $103,077$ training images and $5,000$ testing images. Since text annotations of VG are noisy, we first clean it by removing non-alphabet characters and stop words, and use the `autocorrect` library to correct spelling. Following that, we check if all words in an annotation exist in the word2vec dictionary [71] and remove those that do not. We run this cleaning process on both training and testing set and get $99,961$ training images and $4,871$ testing images, with $53,304$ object categories and $29,086$ relation categories. We further split the training set into $97,961$ training and $2,000$ validation images.[1]

Evaluation protocol. For VRD, we use the same evaluation metrics used in [128], which runs relationship detection using non-ground-truth proposals and reports recall rates using the top 50

[1] We will release the cleaned annotations along with our code.

	Relationship		Phrase		Relationship Detection						Phrase Detection					
	free k				k = 1		k = 10		k = 70		k = 1		k = 10		k = 70	
Recall at	50	100	50	100	50	100	50	100	50	100	50	100	50	100	50	100
w/ proposals from [62]																
CAI*[146]	15.63	17.39	17.60	19.24	-	-	-	-	-	-	-	-	-	-	-	-
Language cues[84]	16.89	20.70	15.08	18.37	-	-	16.89	20.70	-	-	-	-	15.08	18.37	-	-
VRD[62]	17.43	22.03	20.42	25.52	13.80	14.70	17.43	22.03	17.35	21.51	16.17	17.03	20.42	25.52	20.04	24.90
Ours	**19.18**	**22.64**	**21.69**	**25.92**	**16.08**	**17.07**	**19.18**	**22.64**	**18.89**	**22.35**	**18.32**	**19.78**	**21.69**	**25.92**	**21.39**	**25.65**
w/ better proposals																
DR-Net*[13]	17.73	20.88	19.93	23.45	-	-	-	-	-	-	-	-	-	-	-	-
ViP-CNN[55]	17.32	20.01	22.78	27.91	17.32	20.01	-	-	-	-	22.78	27.91	-	-	-	-
VRL[56]	18.19	20.79	21.37	22.60	18.19	20.79	-	-	-	-	21.37	22.60	-	-	-	-
PPRFCN*[135]	14.41	15.72	19.62	23.75	-	-	-	-	-	-	-	-	-	-	-	-
VTransE*	14.07	15.20	19.42	22.42	-	-	-	-	-	-	-	-	-	-	-	-
SA-Full*[82]	15.80	17.10	17.90	19.50	-	-	-	-	-	-	-	-	-	-	-	-
CAI*[146]	20.14	23.39	23.88	25.26	-	-	-	-	-	-	-	-	-	-	-	-
KL distilation[128]	22.68	31.89	26.47	29.76	19.17	21.34	22.56	29.89	22.68	31.89	23.14	24.03	26.47	29.76	26.32	29.43
Zoom-Net[122]	21.37	27.30	29.05	37.34	18.92	21.41	-	-	21.37	27.30	24.82	28.09	-	-	29.05	37.34
CAI + SCA-M[122]	22.34	28.52	29.64	38.39	19.54	22.39	-	-	22.34	28.52	25.21	28.89	-	-	29.64	38.39
Ours	**26.98**	**32.63**	**32.90**	**39.66**	**23.68**	**26.67**	**26.98**	**32.63**	**26.98**	**32.59**	**28.93**	**32.85**	**32.90**	**39.66**	**32.90**	**39.64**

Table 4.1: Comparison with state-of-the-art on the VRD dataset.

	Scene Graph Detection			Scene Graph Classification			Predicate Classification		
Recall at	20	50	100	20	50	100	20	50	100
VRD[62]	-	0.3	0.5	-	11.8	14.1	-	27.9	35.0
Message Passing[114]	-	3.4	4.2	-	21.7	24.4	-	44.8	53.0
Message Passing+	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3
Associative Embedding[76]	6.5	8.1	8.2	18.2	21.8	22.6	47.9	54.1	55.4
Frequency	17.7	23.5	27.6	27.7	32.4	34.0	49.4	59.9	64.1
Frequency+Overlap	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2
MotifNet-LeftRight [133]	**21.4**	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1
Ours	20.7	**27.9**	**32.5**	**36.0**	**36.7**	**36.7**	**66.8**	**68.4**	**68.4**

Table 4.2: Comparison with state-of-the-art on the VG200 dataset.

and 100 relationship predictions, with $k = 1, 10, 70$ relations per relationship proposal before taking the top 50 and 100 predictions.

For VG200, we use the same evaluation metrics used in [133], which uses three modes: 1) **predicate classification**: predict predicate labels given ground truth subject and object boxes and labels; 2) **scene graph classification**: predict subject, object and predicate labels given ground truth subject and object boxes; 3) **scene graph detection**: predict all the three labels and two boxes. Recalls under the top 20, 50, 100 predictions are used as metrics. The mean is computed over the 3 evaluation modes over R@50 and R@100 as in [133].

For VG80k, we evaluate all methods on the whole $53,304$ object and $29,086$ relation categories. We use ground-truth boxes as relationship proposals, meaning there is no localization errors and the results directly reflect recognition ability of a model. We use the following metrics to measure performance: (1) top1, top5, and top10 accuracy, (2) mean reciprocal ranking (rr), defined as $\frac{1}{M}\sum_{i=1}^{M}\frac{1}{rank_i}$, (3) mean ranking (mr), defined as $\frac{1}{M}\sum_{i=1}^{M} rank_i$, smaller is better.

4.3.1 Evaluation of Relationship Detection on VRD

We first validate our model on VRD dataset with comparison to state-of-the-art methods using the metrics presented in [128] in Table 5.8. Note that there is a variable k in this metric which is the number of relation candidates when selecting top50/100. Since not all previous methods specified k in their evaluation, we first report performance in the "free k" column when considering k as a hyper-parameter that can be cross-validated. For methods where the k is reported for 1 or more values, the column reports the performance using the best k. We then list all available results with specific k in the right two columns.

For fairness, we split the table in two parts. The top part lists methods that use the same proposals from [62], while the bottom part lists methods that are based on a different set of proposals, and ours uses better proposals obtained from Faster-RCNN as previous works. We can see that we outperform all other methods with proposals from [62] even without using message-passing-like post processing as in [55, 13], and also very competitive to the overall best performing method from [128]. Note that although spatial features could be advantageous for VRD according to previous methods, we do not use them in our model in concern of large-scale settings. We expect better performance if integrating spatial features for VRD, but for model consistency we do experiments without it everywhere.

4.3.2 Scene Graph Classification & Detection on VG200

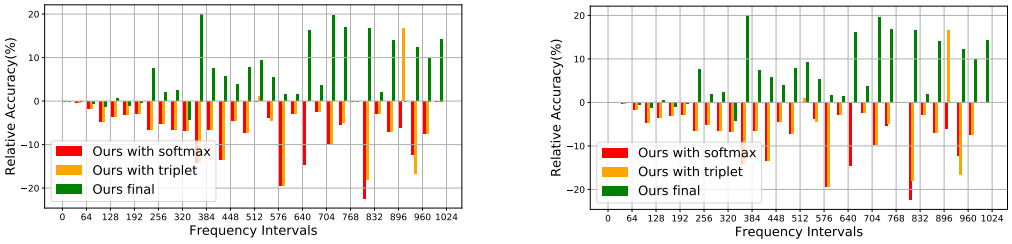
We present our results in Table 4.2. Note that scene graph classification isolates the factor of subject/object localization accuracy by using ground truth subject/object boxes, meaning that it focuses more on the relationship recognition ability of a model, and predicate classification focuses even more on it by using ground truth subject/object boxes and labels. It is clear that the gaps between our model and others are higher on scene graph/predicate classification, meaning our model displays superior relation recognition ability.

4.3.3 Relationship Recognition on VG80k

Baselines. Since there is no previous method that has been evaluated in our large-scale setting, we carefully design 3 baselines to compare with. 1) 3-branch Fast-RCNN: an intuitively

	Relationship Triplet					Relation				
	top1	top5	top10	rr	mr	top1	top5	top10	rr	mr
All classes										
3-branch Fast-RCNN	9.73	41.95	55.19	52.10	16.36	36.00	69.59	79.83	50.77	7.81
ours w/ triplet	8.01	27.06	35.27	40.33	32.10	37.98	61.34	69.60	48.28	14.12
ours w/ softmax	14.53	46.33	57.30	55.61	16.94	49.83	76.06	82.20	61.60	8.21
ours final	**15.72**	**48.83**	**59.87**	**57.53**	**15.08**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**
Tail classes										
3-branch Fast-RCNN	0.32	3.24	7.69	24.56	49.12	0.91	4.36	9.77	4.09	52.19
ours w/ triplet	0.02	0.29	0.58	7.73	83.75	0.12	0.61	1.10	0.68	86.60
ours w/ softmax	0.00	0.07	0.47	20.36	58.50	0.00	0.08	0.55	1.11	65.02
ours final	**0.48**	**13.33**	**28.12**	**43.26**	**45.48**	**0.96**	**7.61**	**16.36**	**5.56**	**45.70**

Table 4.3: Results on all relation classes and tail classes ($\#occurrence \leq 1024$) in VG80k. Note that since VG80k is extremely imbalanced, classes with no greater than 1024 occurrences are still in the tail. In fact, there are more than 99% of relation classes but only 10.04% instances of these classes that occur for no more than 1024 times.



(a) Top 5 rel triplet

(b) Top 5 relation

Figure 4.3: Top-5 relative accuracies against the 3-branch Fast-RCNN baseline in the tail intervals. The intervals are defined as bins of 32 from 1 to 1024 occurrences of the relation classes.

straightforward model is a Fast-RCNN with a shared $conv1$ to $conv5$ backbone and 3 fc branches for subject, relation and object respectively, where the subject and object branches share weights since they are essentially an object detector; 2) our model with softmax loss: we replace our loss with softmax loss; 3) our model with triplet loss: we replace our loss with triplet loss.

Results. As shown in Table 4.3, we can see that our loss is the best for the general case where all instances from all classes are considered. The baseline has reasonable performance but is clearly worse than ours with softmax, demonstrating that our visual module is critical for efficient learning. Ours with triplet is worse than ours with softmax in the general case since triplet loss is not discriminative enough among the massive data. However it is the opposite for tail classes (i.e., $\#occurrence \leq 1024$), since recognition of infrequent classes can benefit from the transferred knowledge learned from frequent classes, which the softmax-based model is not capable of. Another observation is that although the 3-branch Fast-RCNN baseline works

	Relationship Triplet					Relation				
Methods	top1	top5	top10	rr	mr	top1	top5	top10	rr	mr
wiki	15.59	46.03	54.78	52.45	25.31	51.96	78.56	84.38	63.61	8.61
relco	15.58	46.63	55.91	54.03	22.23	52.00	79.06	84.75	63.90	7.74
wiki + relco	**15.72**	**48.83**	**59.87**	**57.53**	**15.08**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**
wiki + node2vec	15.62	47.58	57.48	54.75	20.93	51.92	78.83	85.01	63.86	7.64
0 sem layer	11.21	28.78	34.84	38.64	43.49	44.66	60.06	64.74	51.60	24.74
1 sem layer	**15.75**	48.23	58.28	55.70	19.15	51.82	78.94	85.00	63.79	7.63
2 sem layer	15.72	**48.83**	**59.87**	**57.53**	**15.08**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**
3 sem layer	15.49	48.42	58.75	56.98	15.83	**52.00**	79.19	85.08	63.99	6.40
no concat	10.47	42.51	54.51	51.51	20.16	36.96	70.44	80.01	51.62	9.26
early concat	15.09	45.88	55.72	54.72	19.69	49.54	75.56	81.49	61.25	8.82
late concat	15.57	47.72	58.05	55.34	19.27	51.06	78.15	84.47	63.03	7.90
both concat	**15.72**	**48.83**	**59.87**	**57.53**	**20.62**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**
$\mathcal{L}_y$	15.21	47.28	57.77	55.06	19.12	50.67	78.21	84.70	62.82	7.31
$\mathcal{L}_y + \mathcal{L}_x$	15.07	47.37	57.85	54.92	19.59	50.60	78.06	84.40	62.71	7.60
$\mathcal{L}_y + \mathcal{L}_c$	15.53	47.97	58.49	55.78	18.55	51.48	78.99	84.90	63.59	7.32
$\mathcal{L}_y + \mathcal{L}_x + \mathcal{L}_c$	**15.72**	**48.83**	**59.87**	**57.53**	**15.08**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**

Table 4.4: Ablation study of our model on VG80k.

poorly in the general case, it is better than our model with softmax. Since the main difference of them is with and without visual feature concatenation, it means that integrating subject and object features does not necessarily helps infrequent relation classes. This is because subject and object features could lead to strong prior on the relation, resulting in lower chance of predicting infrequent relation when using softmax. For example, when seeing a rare image where the relationship is "dog ride horse", subject being "dog" and object being "horse" would give very little probability to the relation "ride", even though it is the correct answer. Our model alleviates this problem by not mapping visual features directly to the discrete categorical space, but to a continuous embedding space where visual similarity is preserved. Therefore, when seeing the visual features of "dog", "horse" and the whole "dog ride horse" context, our model is able to associate them with a visually similar relationship "person ride horse" and correctly output the relation "ride".

4.3.4 Ablation Study

Variants of our model. We explore variants of our model in 4 dimensions: 1) the semantic embeddings fed to the semantic module; 2) structure of the semantic module; 3) structure of the visual module; 4) the losses. The default settings of them are 1) using *wiki + relco*; 2) 2 semantic layer; 3) with both visual concatenation; 4) with all the 3 loss terms. We fix the other 3 dimensions as the default settings when exploring one of them.

	Relationship Triplet					Relation				
λ =	top1	top5	top10	rr	mr	top1	top5	top10	rr	mr
1.0	0.00	0.61	3.77	22.43	48.24	0.04	1.12	5.97	4.11	21.39
2.0	8.48	27.63	34.26	35.25	46.28	44.94	70.60	76.63	56.69	13.20
3.0	14.19	39.22	46.71	48.80	29.65	51.07	74.61	78.74	61.74	10.88
4.0	**15.72**	47.19	56.94	54.80	20.85	51.67	78.66	84.23	63.53	8.68
5.0	**15.72**	**48.83**	**59.87**	**57.53**	**15.08**	**52.00**	**79.37**	**85.60**	**64.12**	**6.21**
6.0	15.32	47.99	58.10	55.57	18.67	51.60	78.95	85.05	63.62	7.23
7.0	15.11	44.72	54.68	54.04	20.82	51.23	77.37	83.37	62.95	7.86
8.0	14.84	45.12	54.95	54.07	20.56	51.25	77.67	83.36	62.97	7.81
9.0	14.81	45.72	55.81	54.29	20.10	50.88	78.59	84.70	63.08	7.21
10.0	14.71	45.62	55.71	54.19	20.19	51.07	78.64	84.78	63.21	7.26

Table 4.5: Performances of our model on VG80k validation set with different values of the scaling factor. We use scaling factor $\lambda = 5.0$ for all our experiments on VG80k.

The scaling factor before softmax. As mentioned in the implementation details, this value scales up the output by a value that is close to the average norm of the input and prevents gradient vanishing caused by the normalization. Specifically, for Eq(7) in the paper we use $s(\mathbf{x}, \mathbf{y}) = \lambda \frac{\mathbf{x}^T \mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$ where λ is the scaling factor. In Table 4.5 we show results of our model when changing the value of the scaling factor applied before the softmax layer. We observe that when the value is close to the average norm of all input vectors (i.e., 5.0), we achieve optimal performance, although slight difference of this value does not change results too much (i.e., when it is 4.0 or 6.0). It is clear that when the scaling factor is 1.0, which is equivalent to training without scaling, the model is not sufficiently trained. We therefore pick 5.0 for this scaling factor for all the other experiments on VG80k.

Which semantic embedding to use? We explore 4 settings: 1) *wiki* and 2) *relco* use wikipedia and relationship-level co-occurrence embedding alone, while 3) *wiki* + *relco* and 4) *wiki* + *node2vec* use concatenation of two embeddings. The intuition of concatenating *wiki* with *relco* and *node2vec* is that *wiki* contains common knowledge acquired outside of the dataset, while *relco* and *node2vec* are trained specifically on VG80k, and their combination provides abundant information for the semantic module. As shown in Table 4.4, fusion of *wiki* and *relco* outperforms each one alone with clear margins. We found that using *node2vec* alone does not perform reasonably, but *wiki* + *node2vec* is competitive to others, demonstrating the efficacy of concatenation.

Number of semantic layers. We also study how many, if any, layers are necessary to embed the word vectors. As it is shown in Table 4.4, directly using the word vectors (0 semantic layers)

	Relationship Triplet					Relation				
m =	top1	top5	top10	rr	mr	top1	top5	top10	rr	mr
0.1	7.77	29.84	**38.53**	**42.29**	**28.13**	36.50	**63.50**	**70.20**	47.48	14.20
0.2	**8.01**	**27.06**	35.27	40.33	32.10	**37.98**	61.34	69.60	**48.28**	**14.12**
0.3	5.78	24.39	33.26	37.03	34.55	36.75	58.65	64.86	46.62	20.62
0.4	3.82	22.55	31.70	34.10	36.26	34.89	57.25	63.74	45.04	21.89
0.5	3.14	19.69	30.01	31.63	38.25	33.65	56.16	62.77	43.88	23.19
0.6	2.64	15.68	27.65	29.74	39.70	32.15	55.08	61.68	42.52	24.25
0.7	2.17	11.35	24.55	28.06	41.47	30.36	54.20	60.60	41.02	25.23
0.8	1.87	8.71	16.30	26.43	43.18	29.78	53.43	60.01	40.29	26.19
0.9	1.43	7.44	11.50	24.76	44.83	28.35	51.73	58.74	38.89	27.27
1.0	1.10	6.97	10.51	23.57	46.60	27.49	50.72	58.10	37.97	28.13

Table 4.6: Performances of triplet loss on VG80k validation set with different values of margin m. We use margin $m = 0.2$ for all our experiments in the main paper.

is not a good substitute of our learned embedding; raw word vectors are learned to represent as much associations between words as possible, but not to distinguish them. We find that either 1 or 2 layers give similarly good results and 2 layers are slightly better, though performance starts to degrade when adding more layers.

Are both visual feature concatenations necessary? In Table 4.4, "early concat" means using only the first concatenation of the three branches, and "late concat" means the second. Both early and late concatenation boost performance significantly compared to no concatenation, and it is the best with both. Another observation is that late concatenation is better than early alone. We believe the reason is, as mentioned above, relations are naturally conditioned on and constrained by subjects and objects, e.g., given "man" as subject and "chair" as object, it is highly likely that the relation is "sit on". Since late concatenation is at a higher level, it integrates features that are more semantically close to the subject and object labels, which gives stronger prior to the relation branch and affects relation prediction more than the early concatenation.

Do all the losses help? In order to understand how each loss helps training, we trained 3 models of which each excludes one or two loss terms. We can see that using $\mathcal{L}_y + \mathcal{L}_x$ is similar with $\mathcal{L}_y$, and it is the best with all the three losses. This is because $\mathcal{L}_x$ pulls positive x pairs close while pushes negative x away. However, since (x, y) is a many-to-one mapping (i.e., multiple visual features could have the same label), there is no guarantee that all x with the same y would be embedded closely, if not using $\mathcal{L}_c$. By introducing $\mathcal{L}_c$, x with the same y are forced to be close to each other, and thus the structural consistency of visual features is preserved.

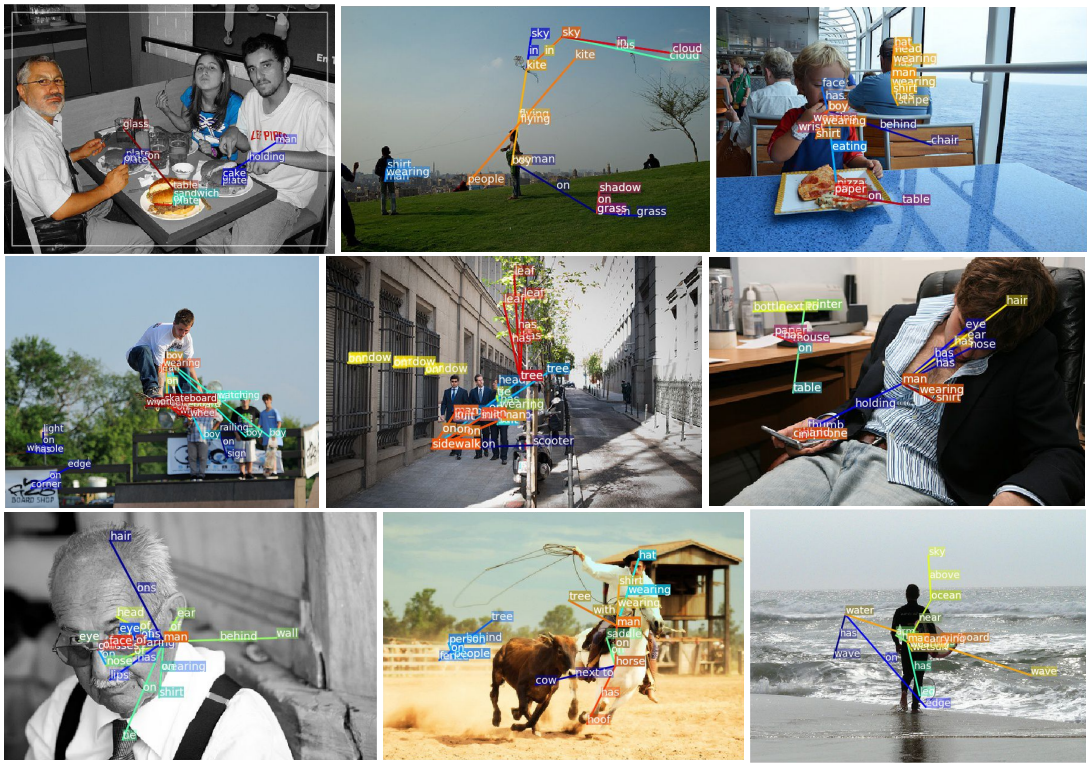


Figure 4.4: Qualitative results. Our model recognizes a wide range of relation ship triples. Even if they are not always matching the ground truth they are frequently correct or at least reasonable as the ground truth is not complete.

The margin m in triplet loss We show results of triplet loss with various values for the margin m in Table 4.6. As described earlier, this value allows slackness in pushing negative pairs away from positive ones. We observe similar results with previous works [46, 19] that it is the best to set $m = 0.1$ or $m = 0.2$ in order to achieve optimal performance. It is clear that triplet loss is not able to learn discriminative embeddings that are suitable for classification tasks, even with larger m that can theoretically enforce more contrast against negative labels. We believe that the main reason is that in a hinge loss form, triplet loss treats all negative pairs equally "hard" as long as they are within the margin m. However, as shown by the successful softmax models, "easy" negatives (e.g., those that are close to positives) should be penalized less than those "hard" ones, which is a property our model has since we utilize softmax for contrastive training.

4.3.5 Qualitative results

The VG80k has densely annotated relationships for most images with a wide range of types. In Figure 4.4 there are interactive relationships such as “boy flying kite”, “batter holding bat”, positional relationships such as “glass on table”, “man next to man”, attributive relationships such as “man in suit” and “boy has face”. Our model is able to cover all these kinds, no matter frequent or infrequent, and even for those incorrect predictions, our answers are still semantic meaningful and similar to the ground-truth, e.g., the ground-truth “lamp on pole” v.s. the predicted “light on pole”, and the ground-truth “motorcycle on sidewalk” v.s. the predicted “scooter on sidewalk”.

4.4 Summary

In this work we study visual relationship detection at an unprecedented scale and propose a novel model that can generalize better on long tail class distributions. We find it is crucial to integrate subject and object features at multiple levels for good relation embeddings and further design a loss that learns to embed visual and semantic features into a shared space, where semantic correlations between categories are kept without hurting discriminative ability. We validate the effectiveness of our model on multiple datasets, both on the classification and detection task, and demonstrate the superiority of our approach over strong baselines and the state-of-the-art. Future work includes integrating a relationship proposal into our model that would enable end-to-end training.

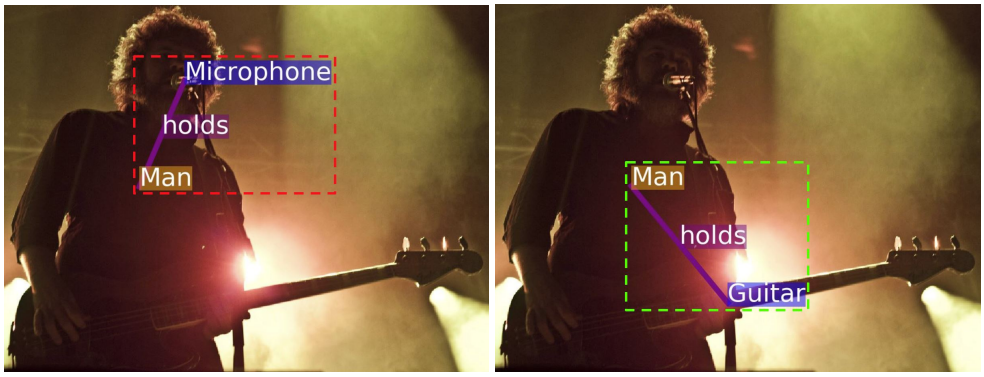
Chapter 5

Graphical Contrastive Losses for Scene Graph Parsing

Most scene graph parsers use a two-stage pipeline to detect visual relationships: the first stage detects entities, and the second predicts the predicate for each entity pair using a softmax distribution. We find that such pipelines, trained with only a cross entropy loss over predicate classes, suffer from two common errors. The first, Entity Instance Confusion, occurs when the model confuses multiple instances of the same type of entity (e.g. multiple cups). The second, Proximal Relationship Ambiguity, arises when multiple subject-predicate-object triplets appear in close proximity with the same predicate, and the model struggles to infer the correct subject-object pairings (e.g. mis-pairing musicians and their instruments). We propose a set of contrastive loss formulations that specifically target these types of errors within the scene graph parsing problem, collectively termed the Graphical Contrastive Losses. These losses explicitly force the model to disambiguate related and unrelated instances through margin constraints specific to each type of confusion. We further construct a relationship detector, called RelDN, using the aforementioned pipeline to demonstrate the efficacy of our proposed losses. Our model outperforms the winning method of the OpenImages Relationship Detection Challenge by 4.7% (16.5% relative) on the test set. We also show improved results over the best previous methods on the Visual Genome and Visual Relationship Detection datasets.

5.1 Introduction

Given an image, the aim of scene graph parsing is to infer a visually grounded graph comprising localized entity categories, along with edges denoting their pairwise relationships. This is often formulated as the detection of $\langle subject, predicate, object \rangle$ triplets, e.g. $\langle man, holds, guitar \rangle$ in Figure 5.1b. Current state-of-the-art methods achieve this goal by a two-stage mechanism: first detecting entities, then predicting a predicate for each pair of entities.

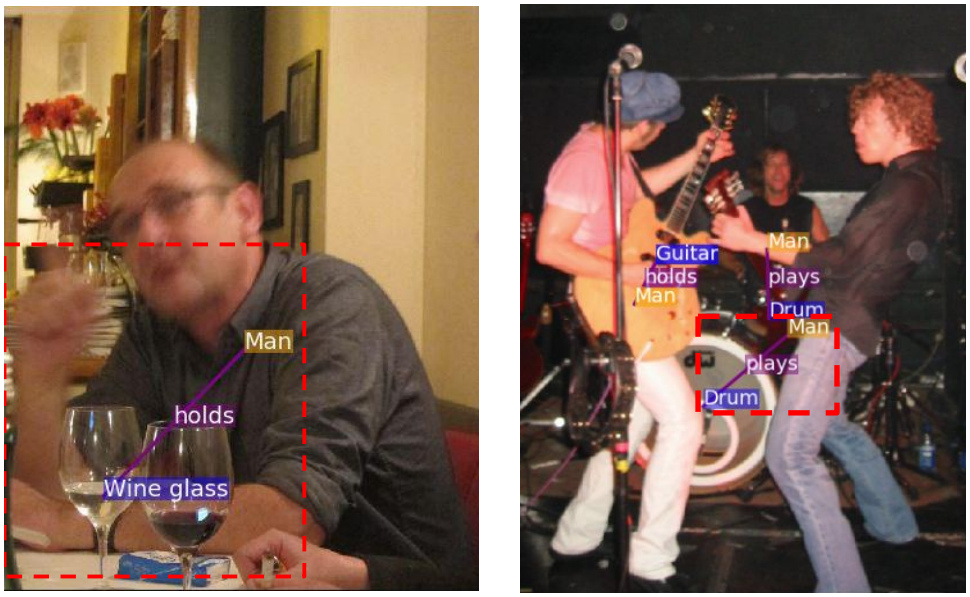


(a) without our losses (b) with our losses

Figure 5.1: Example of failure of models without our losses and success of our losses. (a) RelDN learned with only multi-class cross-entropy loss incorrectly relates the man with the microphone, while (b) RelDN learned with our *Graphical Contrastive Losses* detects the correct relationship $\langle man, holds, guitar \rangle$.

We find that scene graph parsing models using such pipelines tend to struggle with two types of errors. The first is **Entity Instance Confusion**, in which the subject or object is related to one of many instances of the same class, and the model fails to distinguish between the target instance and the others. We show an example in Figure 5.2a, in which the model identifies the man is holding a wine glass, but struggles to determine exactly which of the 3 visually similar wine glasses is being held. The incorrectly predicted wine glass is transparent and intersecting with the left arm, which makes it look like being held. The second type of error, **Proximal Relationship Ambiguity**, occurs when the image contains multiple subject-object pairs interacting in the same way, and the model fails to identify the correct pairing. An example can be seen in the multiple musicians ”playing” their respective instruments in Figure 5.2b. Due to their close proximity, visual features for each musician-instrument pair overlap significantly, making it difficult for the scene graph models to identify the correct pairings.

The primary cause of these two failures lies in the inherent difficulty of inferring relationships like “hold” and “play” from visual cues. Concretely, which glass is being held is determined by the small part of the hand that covers the glass. Whether a player is playing the drum can only be inferred by very subtle visual cues such as his standing pose or where his fingers are placed. It is challenging for any model to learn to attend to these details precisely, and it would be impractical to specify which details to focus on for all kinds of relationships, let alone to learn all these details. These challenges motivate the need for a mechanism that can automatically learn fine details that determine visual relationships, and explicitly discriminate



(a) Entity Instance Confusion (b) Proximal Rel Ambiguity

Figure 5.2: Examples of Entity Instance Confusion and Proximal Relationship Ambiguity. Red boxes highlight relationships our baseline model predicts incorrectly. (a) the man is not holding the predicted wine glass. (b) the guitar player on the right is not playing drum.

related entities from unrelated ones, for all types of relationships. This is the goal of our work.

In this paper we propose a set of *Graphical Contrastive Losses* to tackle these issues. The losses use the form of the margin-based triplet loss, but are specifically designed to address the two aforementioned errors. It adds additional supervision in the form of hard negatives specific to Entity Instance Confusion and Proximal Relationship Ambiguity. To demonstrate the effectiveness of our proposed losses, we design a relationship detection network named *RelDN* using the aforementioned pipeline with our losses. Figure 5.1 shows a result of RelDN with N-way cross-entropy loss only vs. with our additional contrastive losses. Our best model achieves 0.328 on the Private set of the OpenImages Relationship Detection Challenge, outperforming the winning model by a significant 4.7% (16.5% relative) margin. It also attains state-of-the-art performance on the Visual Genome[48] and VRD[62] datasets.

In this paper, we denote subject, predicate, object and attribute with $s, pred, o, a$. We use "entity" to describe individual detected objects to distinguish from "object" in the semantic sense, and use "relationships" to describe the entire $\langle s, pred, o \rangle$ tuple, not to be confused with "predicate," which is an element of said tuple.

5.2 Graphical Contrastive Losses

Our Graphical Contrastive Losses encompass three types of loss, each addressing the two aforementioned issues in their own way: 1) **Class Agnostic**: contrasts positive/negative entity pairs regardless of their relation and adds contrastive supervision for generic cases; 2) **Entity Class Aware**: addresses the issue in Figure 5.2a by focusing on entities with the same class; 3) **Predicate Class Aware**: addresses the issue in Figure 5.2b by focusing on entity pairs with the same potential predicate. We define our contrastive losses over an affinity term $\Phi(s, o)$, which can be interpreted as the probability that subject s and object o have some relationship or interaction. Given a model that outputs the distribution over predicate classes conditioned on a subject and object pair $p(pred|s, o)$, we define $\Phi(s, o)$ as:

$$\Phi(s, o) = 1 - p(pred = \varnothing|s, o) \tag{5.1}$$

where $\varnothing$ is the class symbol representing `no_relationship`. This is equivalent to summing over all predicate classes except $\varnothing$.

5.2.1 Class Agnostic Loss

Our first contrastive loss term aims to maximize the affinity of the lowest scoring positive pairing and minimize the affinity of the highest scoring negative pairing. For a subject indexed by i and an object indexed by j, the margins we wish to maximize can be written as:

$$\begin{aligned} m_1^s(i) &= \min_{j \in \mathcal{V}_i^+} \Phi(s_i, o_j^+) - \max_{k \in \mathcal{V}_i^-} \Phi(s_i, o_k^-) \\ m_1^o(j) &= \min_{i \in \mathcal{V}_j^+} \Phi(s_i^+, o_j) - \max_{k \in \mathcal{V}_j^-} \Phi(s_k^-, o_j) \end{aligned} \tag{5.2}$$

where $\mathcal{V}_i^+$ and $\mathcal{V}_i^-$ represent sets of objects related to and not related to subject s_i; $\mathcal{V}_j^+$ and $\mathcal{V}_j^-$ are defined similarly for object j as the sets of subjects related to and not related to o_j.

The class agnostic loss for all sampled positive subjects and objects is written as:

$$\begin{aligned} L_1 =& \frac{1}{N}\sum_{i=1}^{N}\max(0, \alpha_1 - m_1^s(i)) \\ &+ \frac{1}{N}\sum_{j=1}^{N}\max(0, \alpha_1 - m_1^o(j)) \end{aligned} \tag{5.3}$$

where N is the number of annotated entities and α_1 is the margin threshold.

This loss tries to contrast positive and negative (s, o) pairs, ignoring any class information, and is similar to the triplet losses used referring expression and phrase-grounding literature. We found it works as well in our scenario and even better with the following class-aware losses, as shown in Table 5.3.

5.2.2 Entity Class Aware Loss

The Entity Class Aware loss deals with entity instance confusion, in which the model struggles to determine interactions between a subject (object) and multiple instances of a same-class object (subject). It can be viewed as an extension of the Class Agnostic loss where we further specify a class c when populating the positive and negative sets $\mathcal{V}^+$ and $\mathcal{V}^-$. We extend the formulation in equation (5.3) as:

$$\begin{aligned} m_2^s(i, c) &= \min_{j \in \mathcal{V}_i^{c+}} \Phi(s_i, o_j^+) - \max_{k \in \mathcal{V}_i^{c-}} \Phi(s_i, o_k^-) \\ m_2^o(j, c) &= \min_{i \in \mathcal{V}_j^{c+}} \Phi(s_i^+, o_j) - \max_{k \in \mathcal{V}_j^{c-}} \Phi(s_k^-, o_j) \end{aligned} \tag{5.4}$$

where $\mathcal{V}_i^{c+}$, $\mathcal{V}_i^{c-}$, $\mathcal{V}_j^{c+}$ and $\mathcal{V}_j^{c-}$ are now constrained to instances of class c.

The entity class aware loss for all sampled positive subjects and objects is defined as

$$\begin{aligned} L_2 =& \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{C}(\mathcal{V}_i^+)|}\sum_{c \in \mathcal{C}(\mathcal{V}_i^+)}\max(0, \alpha_2 - m_2^s(i, c)) \\ &+ \frac{1}{N}\sum_{j=1}^{N}\frac{1}{|\mathcal{C}(\mathcal{V}_j^+)|}\sum_{c \in \mathcal{C}(\mathcal{V}_j^+)}\max(0, \alpha_2 - m_2^o(j, c)) \end{aligned} \tag{5.5}$$

where $\mathcal{C}()$ returns the set of unique classes of the sets $\mathcal{V}_i^+$ and $\mathcal{V}_j^+$ as defined in the class

agnostic loss. Compared to the class agnostic loss which maximizes the margins across all instances, this loss maximizes the margins between instances of the same class. It forces a model to disentangle confusing entities illustrated in Figure 5.2a, where the subject has several potentially related objects with the same class.

5.2.3 Predicate Class Aware Loss

Similar to the entity class aware loss, this loss maximizes the margins within groups of instances determined by their associated predicates. It is designed to deal with the proximal relationship ambiguity as exemplified in Figure 5.2b, where instances joined by the same predicate class are within close proximity of each other. In the context of Figure 5.2b, this loss would encourage the correct pairing of who is playing which instrument by penalizing wrong pairing, *i.e.*, "man plays drum" in the red box. Replacing the class groupings in equation (5.4) with predicate groupings restricted to predicate class e, we define our margins to maximize as:

$$\begin{aligned} m_3^s(i,e) &= \min_{j \in \mathcal{V}_i^{e+}} \Phi(s_i, o_j^+) - \max_{k \in \mathcal{V}_i^{e-}} \Phi(s_i, o_k^-) \\ m_3^o(j,e) &= \min_{i \in \mathcal{V}_j^{e+}} \Phi(s_i^+, o_j) - \max_{k \in \mathcal{V}_j^{e-}} \Phi(s_k^-, o_j) \end{aligned} \tag{5.6}$$

Here, we define the sets $\mathcal{V}_i^{e+}$ and $\mathcal{V}_j^{e+}$ as the sets of subject-object pairs where the ground truth predicate between s_i and o_j is e, anchored with respect to subject i and object j respectively. We define the sets $\mathcal{V}_i^{e-}$ and $\mathcal{V}_j^{e-}$ as is the set of instances where the model *incorrectly predicts* (via argmax) the predicate to be e, anchored with respect to subject i and object j respectively.

The predicate class aware loss for all sampled positive subjects and objects is defined as

$$\begin{aligned} L_3 =& \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{E}(\mathcal{V}_i^+)|} \sum_{e \in \mathcal{E}(\mathcal{V}_i^+)} \max(0, \alpha_3 - m_3^s(i,e)) \\ &+ \frac{1}{N} \sum_{j=1}^{N} \frac{1}{|\mathcal{E}(\mathcal{V}_j^+)|} \sum_{e \in \mathcal{E}(\mathcal{V}_j^+)} \max(0, \alpha_3 - m_3^o(j,e)) \end{aligned} \tag{5.7}$$

where $\mathcal{E}()$ returns the set of unique predicates associated with the input (excluding $\varnothing$). The

final loss is expressed as:

$$L = L_0 + \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \tag{5.8}$$

where L_0 is the cross-entropy loss over predicate classes.

5.2.4 Complexity Analysis

We look at the case where the subject s_i is fixed and we vary object for positive/negative pairings. The reverse case (object fixed, subject varies) has the same complexity. All sampling is conducted on the entities of a single image per batch. The set of entities include ground truth bounding boxes, as well as any detector output with $>= 0.5$ IOU to ground truth entities.

For the Class Agnostic Loss L_1, the computational complexity of the sampling procedure is $O(N^2)$, where N is the upper bounded on number of sampled entities per image. In practice, for each subject, we randomly sample at most K non-related objects (negative pairings), which makes the actual complexity $O(NK)$.

For the Entity Class Aware Loss L_2, the sampling procedure is the same as with L_1, except that we need to keep only those non-related objects that are of class c, *i.e.*, the object class of the current o in the sampled (s, o) pair. This involves a filtering operation on the K objects which takes $O(K)$ time, therefore the overall complexity is still $O(NK)$.

The analysis for the Predicate Class Aware Loss L_3 is similar to that of L_2, except that the filtering operation looks at the predicate class e instead of the object class c. The overall complexity is also $O(NK)$.

We set $N = 512$ and $K = 64$ per batch in practice.

5.3 RelDN

We demonstrate the efficacy of our proposed losses with our Relationship Detection Network (RelDN). The RelDN follows a two stage pipeline: it first identifies a proposal set of likely subject-object relationship pairs, then extracts features from these candidate regions to perform a fine-grained classification into a predicate class. We build a separate CNN branch for predicates (conv_body_rel) with the same structure as that of entity detector CNN (conv_body_det)

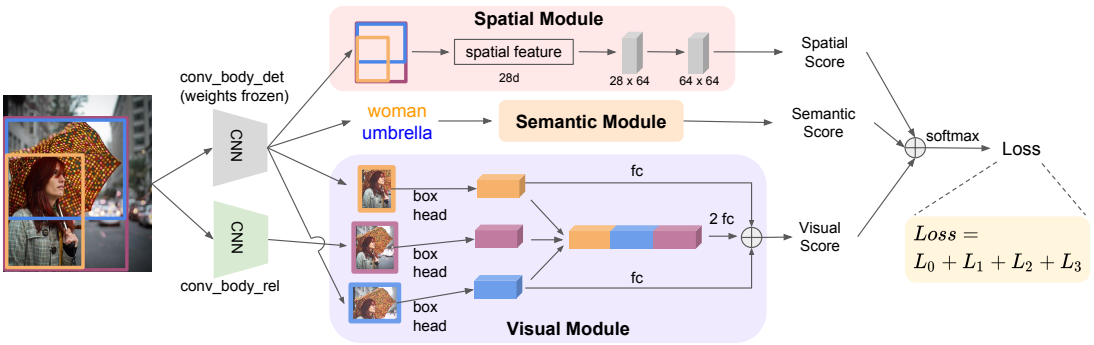


Figure 5.3: The RelDN model architecture. The structures of conv_body_det and conv_body_rel are identical. We freeze the weights of the former and only train the latter.

to extract predicate features. The intuition for having a separate branch is that we want visual features for predicates to focus on the interactive areas of subjects and objects as opposed to individual entities. As Figure 5.4 illustrates, the predicate CNN clearly learns better features which concentrate on regions that strongly imply relationships.

The first stage of the RelDN exhaustively returns bounding box regions containing every pair. In the second stage, it computes three types of features for each relationship proposal: semantic, visual, and spatial. Each feature is used to output a set of class logits, which we combine via element-wise addition, and apply softmax normalization to attain a probability distribution over predicate classes. See Figure 5.3 for our model pipeline.

Semantic Module: The semantic module conditions the predicate class prediction on subject-object class co-occurrence frequencies. It is inspired by Zeller, et al. [132] which introduced a frequency baseline that performs reasonably well on Visual Genome by counting frequencies of predicates given subject and object. Its motivation is that in general, the combination of relationships between two entities is usually very limited, e.g., the relationship between a person-horse subject-object pairing is most likely to be "ride", "walk", or "feed", and unlikely to be "stand on" or "wear". For each training image, we count the occurrences of predicate class $pred$ given subject and object classes s and o in the ground truth annotations. This gives us an empirical distribution $p(pred|s, o)$. We assume that the test set is also drawn from the same distribution.

Spatial Module: The spatial module conditions the predicate class predictions on the relative positions of the subject and object. One of the major predicate types are about positions, for example, "on", "under", or "inside_of." These predicate types can often be inferred using only

relative spatial information. We capture spatial information by encoding the box coordinates of subjects and objects using the box delta [87] and normalized coordinates.

We define the delta feature between two sets of bounding box coordinates as follows:

$$\Delta(b_1, b_2) = \langle \frac{x_1 - x_2}{w_2}, \frac{y_1 - y_2}{h_2}, \log \frac{w_1}{w_2}, \log \frac{h_1}{h_2} \rangle \tag{5.9}$$

where b_1 and b_2 are two coordinate tuples in the form of (x, y, w, h).

We then compute the normalized coordinate features for a bounding box b as follows:

$$c(b) = \langle \frac{x}{w_{img}}, \frac{y}{h_{img}}, \frac{x+w}{w_{img}}, \frac{y+h}{h_{img}}, \frac{wh}{w_{img}h_{img}} \rangle \tag{5.10}$$

where w_{img} and h_{img} are the width and height dimensions of the image. Our spatial feature vector for the subject, object, and predicate bounding boxes b_s, b_o, b_{pred} is represented as:

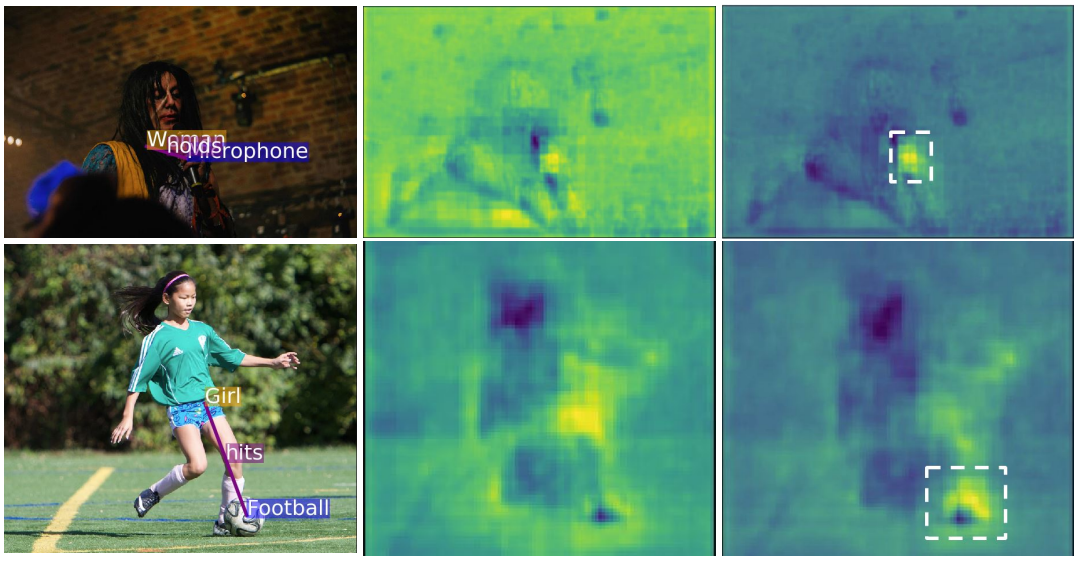
$$\langle \Delta(b_s, b_o), \Delta(b_s, b_{pred}), \Delta(b_{pred}, b_o), c(b_s), c(b_o) \rangle \tag{5.11}$$

Note that b_{pred} is the tightest bounding box around b_s and b_o. This feature vector is fed through an MLP to attain predicate class logit scores.

Visual Module: The visual module produces a set of class logits conditioned ROI feature maps, as in the fast-RCNN pipeline. We extract subject and object ROI features from the entity detector's convolution layers (conv_body_det in Figure 5.3) and extract predicate ROI features from the relationship convolution layers (conv_body_rel in Figure 5.3). The subject, object, and predicate feature vectors are concatenated and passed through an MLP to attain the predicate class logits.

We also include two skip-connections projecting subject-only and object-only ROI features to the predicate class logits. These skip connections are inspired by the observation that many relationships, such as human interactions [26], can be accurately inferred by the appearance of only the subjects or objects. We show an improvement from adding these skip connections in 5.5.4.

Module Fusion: As illustrated in Figure 5.3, we obtain the final probability distribution over



(a) ground truth (b) conv_body_det (c) conv_body_rel

Figure 5.4: Visualization of CNN features by averaging over the channel dimension of convolution feature maps [130]. (a) shows the image ground truth relationships, (b) shows the convolution feature from the entity detector backbone, and (c) shows the feature from the predicate backbone. In all the three examples there are clear shifts of salience from large entities to small areas that strongly indicate the predicates (highlighted in white boxes).

predicate classes by adding the three scores followed by softmax normalization:

$$\mathbf{p}^{pred} = softmax(\mathbf{f}_{vis} + \mathbf{f}_{spt} + \mathbf{f}_{sem}) \tag{5.12}$$

where $\mathbf{f}_{vis}, \mathbf{f}_{spt}, \mathbf{f}_{sem}$ are unnormalized class logits from the visual, spatial, semantic modules.

5.4 Implementation Details

We train the entity detector CNN (conv_body_det) independently using entity annotations, then fix it when training our model. While previous works [55, 13, 122] claim it is beneficial to fine-tune the entity detector end-to-end with the second stage of the pipeline, we opt to freeze our entity detector weights for simplicity. We initialize the predicate CNN (conv_body_rel) with the entity detector's weights and fine-tune it end-to-end with the second stage.

During training, we independently sample positive and negative pairs for each loss, subject to their respective constraints. For L_0, we sample 512 pairs in total where 128 of them are positive. For our class-agnostic loss, we sample 128 positive subjects, then for each of them sample the two closet contrastive pairs according to Eq.5.2; we do the sampling symmetrically

for objects. For our entity and predicate aware losses, we sample in the same way with class-agnostic except that negative pairs are grouped by entity and predicate classes, as described in Eq.5.4,5.6. We set $\lambda_1 = 1.0, \lambda_2 = 0.5, \lambda_3 = 0.1$, determined by cross-validations, for all experiments.

During testing, we take up to 100 outputs from the entity detector and exhaustively group all pairs as relationship proposals/entity pairs. We rank relationship proposals by multiplying the predicted subject, object, predicate probabilities as $\mathbf{p}^{det}(s) \cdot \mathbf{p}^{pred}(pred) \cdot \mathbf{p}^{det}(o)$ where $\mathbf{p}^{det}(s), \mathbf{p}^{det}(o)$ are the probabilities of the predicted subject and object classes from the entity detector, and $\mathbf{p}^{pred}(pred)$ is the probability of the predicted predicate class from the result of Eq.5.12.

To match the architectures of previous state-of-the-art methods, We use ResNeXt-101-FPN [113, 58] as our OpenImages backbone and VGG-16 on Visual Genome (VG) and Visual Relationship Detection (VRD).

5.5 Experiments

We present experimental results on three datasets: OpenImages (OI) [1], Visual Genome (VG) [48] and Visual Relationship Detection (VRD) [62]. We first report evaluation settings, followed by ablation studies and finally external comparisons.

5.5.1 Evaluation Settings

OpenImages: The full train and val sets contains 53,953 and 3,234 images, which takes our model 2 days to train. For quick comparisons, we sample a "mini" subset of 4,500 train and 1,000 validation images where predicate classes are sampled proportionally with a minimum of one instance per class in train and val. We first conduct parameter searches on the mini set, then train and compare with the top model of the OpenImages VRD Challenge [1] on the full set. We show two types of results, one using the same entity detector from the top model, and the other using a detector trained by our own initialized by COCO pre-trained weights.

In the OpenImages Challenge, results are evaluated by calculating Recall@50 (R@50), mean AP of relationships (mAP$_{rel}$), and mean AP of phrases (mAP$_{phr}$). The final score is

L_0	L_1	L_2	L_3	R@50	mAP$_{rel}$	mAP$_{phr}$	score	mAP$^*_{rel}$	mAP$^*_{phr}$	score*
✓				74.67	35.28	41.04	45.46	33.87	38.99	44.08
✓	✓			75.06	**44.18**	**50.19**	**52.76**	35.24	40.30	45.23
✓		✓		74.64	36.19	41.71	46.09	34.67	39.61	44.64
✓			✓	74.88	34.80	40.47	45.08	34.92	40.01	44.95
✓	✓	✓		75.03	35.10	41.18	45.52	35.09	40.22	45.13
✓	✓		✓	**75.30**	43.96	49.61	52.49	34.89	39.87	44.96
✓		✓	✓	75.00	35.83	41.32	45.86	34.62	39.70	44.73
✓	✓	✓	✓	74.94	39.09	44.47	48.41	**35.82**	**40.43**	**45.49**

Table 5.1: Ablation Study on our losses with the official mAP$_{rel}$, mAP$_{phr}$ and score metrics. Metric marked with a * means "under" and "hits" are excluded from evaluation. The fluctuating numbers in mAP$_{rel}$, mAP$_{phr}$ and score indicate that the mAP metrics are unstable and unreliable, while when "under" and "hits" are excluded, all the results become consistent with Table 5.3.

	R@50	mAP$_{rel}$	mAP$_{phr}$	score
L_0	61.72	25.20	35.37	36.57
$L_0 + L_1 + L_2 + L_3$	**62.65**	**26.77**	**36.79**	**37.95**

Table 5.2: Comparison of our model with Graphical Contrastive Loss vs. without the loss on 100 images containing the 5 classes that suffer from the two aforementioned confusions, selected via visual inspection on a random set of images. The metrics are the official mAP$_{rel}$, mAP$_{phr}$ and the score. The "under" and "hits" predicates are not in this 100 image subset.

obtained by score $= 0.2 \times R@50 + 0.4 \times mAP_{rel} + 0.4 \times mAP_{phr}$. The mAP$_{rel}$ evaluates AP of $s, pred, o$ triplets where *both* the subject and object boxes have an IOU of at least 0.5 with ground truth. The mAP$_{phr}$ is similar, but applied to the enclosing relationship box[1]. In practice, we find mAP$_{rel}$ and mAP$_{phr}$ to suffer from extreme predicate class imbalance. For example, 64.48% of the relationships in val have the predicate "at", while only 0.03% of them are "under". This means a single "under" relationship is worth much more than the more common "at" relationships. We address this by scaling each predicate category by their relative ratios in the val set, which we refer to as the weighted mAP (wmAP). We use wmAP in all of our ablation studies (Table 5.3-5.6), in addition to reporting score$_{wtd}$ which replaces mAP with wmAP in the score formula.

We compare with other top models on the official evaluation server. The official test set is split into a Public and Private set with a 30%/70% split. The Public set is used as a dev set. We present individual results for both, as well as their weighted average under Overall in Table 5.9.

Visual Genome: We follow the same train/val splits and evaluation metrics as [132]. We

[1]More details of evaluation can be found on the official page: `https://storage.googleapis.com/openimages/web/vrd_detection_metric.html`

L_0 only with our losses L_0 only with our losses

(a) Entity Instance Confusion (b) Proximal Relationship Ambiguity

Figure 5.5: Example results of RelDN with L_0 only and with our losses. The top row shows RelDN outputs and the bottom row visualizes the learned predicate CNN features of the two models. Red and green boxes highlight the wrong and right outputs (the first row) or feature saliency (the second row). As it shows, our losses force the model to attend to the representative regions that discriminate the correct relationships against unrelated entity pairs, thus is able to disentangle entity instance confusion and proximal relationship ambiguity.

train our entity detector initialized by COCO pre-trained weights. Following [132], we conduct three evaluations: scene graph detection(SGDET), scene graph classification (SGCLS), and predicate classification (PRDCLS). We report results for these tasks with and without the Graphical Contrastive Losses.

VRD: We evaluate our model with entity detectors initialized by ImageNet and COCO pre-trained weights. We use the same evaluation metrics as in [128], which reports R@50 and R@100 for relationship predictions at 1, 10, and 70 predicates per entity pair.

5.5.2 Loss Analysis

Loss Combinations: We now look at whether our proposed losses reduce two aforementioned errors without affecting the overall performance, and whether all three losses are necessary. Results in Table 5.3 show that combination of all the three losses with the N-way cross-entropy loss ($L_0 + L_1 + L_2 + L_3$) has consistently superior performance over just L_0. Notably, AP_{rel} on "holds" improves by from 41.84 to 43.09 (+1.3). It improves even more significantly from 36.04 to 41.04 (+5.0) on "plays" and from 40.43 to 44.16 (+3.7) on "interacts_with" respectively. These three classes suffer the most from the two aforementioned problems. Our results also show that any subset of the losses is worse than the entire ensemble. We see that $L_0 + L_1$,

$L_0 + L_2$ and $L_0 + L_3$ are inferior to $L_0 + L_1 + L_2 + L_3$, especially on "holds", "plays", and "interacts_with", where the largest margin is 3.87 ($L_0 + L_2$ vs. $L_0 + L_1 + L_2 + L_3$ on "play").

To better verify the isolated impact of our losses, we carefully sample a subset of 100 images containing five predicates that significantly suffer from the two aforementioned problems, selected via visual inspection on a random set of images. The five predicates are "at", "holds", "plays", "interacts_with", and "wears". We sample them by looking at the raw images and select those with either entity instance confusion or proximal relationship ambiguity. Example images can be found in Figure 5.7. Table 5.4 shows comparison of our losses with L_0 only on this subset. The overall gap is 1.4 and the largest gap is 4.1 at AP_{rel} on "holds".

Figure 5.5 shows two examples from this subset, one containing entity instance confusion and the other containing proximal relationship ambiguity. In Figure 5.5a the model with only L_0 fails to identify the wine glass being held, while by adding our losses, the area surrounding the correct wine glass lights up. In Figure 5.5b $\langle woman, plays, drum \rangle$ is incorrectly predicted since the L_0-only model mistakenly pairs the unplayed drum with the singer – a reasonable error considering the amount of person-play-drum examples as well as the relative proximities between the singer and the drum. Our losses successfully suppress that region and attend to the correct microphone being held, demonstrating the effectiveness of our hard-negative sampling strategies.

Margin Thresholds: We study the effects of various values of the margin thresholds $\alpha_1, \alpha_2, \alpha_3$ used in Eq.5.3,5.5,5.7. For each experiment, we set $\alpha_1 = \alpha_2 = \alpha_3 = m$ while varying m. As shown in Table 5.6, we observe similar results with previous work [46, 103] that $m = 0.1$ or $m = 0.2$ achieves the best performance. Note that $m = 1.0$ is the largest possible margin, as our affinity scores range from 0 to 1.

5.5.3 Loss Analysis with the Official mAP metrics

Here, we show our ablation studies using the official uniform-class-weighting evaluation metrics, mAP_{rel}, mAP_{phr} and *score*. We also include mAP^*_{rel}, mAP^*_{phr} and *score**, which is the standard mAP and score excluding "under" and "hits" in the evaluation. Table 5.1 presents ablation study results on loss components. Table 5.2 shows comparison between the L_0-only model against the model with our losses on the 100 selected images. In Table 5.1 the variation

L_0	L_1	L_2	L_3	R@50	wmAP$_{rel}$	wmAP$_{phr}$	score$_{wtd}$	AP$_{rel}$ per class								
								at	on	holds	plays	interacts_with	wears	inside_of	*under*	*hits*
✓				74.67	34.63	37.89	43.94	32.40	36.51	41.84	36.04	40.43	5.70	44.17	*25.00*	*55.40*
✓	✓			75.06	35.25	38.37	44.46	32.78	36.96	42.93	37.55	43.30	**9.01**	44.15	*100.00*	*50.95*
✓		✓		74.64	35.03	38.18	44.21	32.76	36.82	42.24	37.17	40.47	8.53	44.71	*33.33*	*49.68*
✓			✓	74.88	35.19	38.27	44.36	32.88	36.73	42.38	38.03	43.53	6.71	44.18	*16.67*	*52.06*
✓	✓	✓		75.03	35.38	38.50	44.56	**32.95**	**37.10**	42.82	38.58	43.66	6.79	43.72	*20.00*	*50.24*
✓	✓		✓	**75.30**	35.30	38.27	44.49	32.92	36.73	42.58	38.81	44.13	6.35	42.74	*100.00*	*51.40*
✓		✓	✓	75.00	35.12	38.34	44.39	32.79	36.47	42.31	39.74	41.35	6.11	43.57	*25.00*	*55.12*
✓	✓	✓	✓	74.94	**35.54**	**38.52**	**44.61**	32.92	37.00	**43.09**	**41.04**	**44.16**	7.83	**44.72**	*50.00*	*51.04*

Table 5.3: Ablation Study on our losses. We report a frequency-balanced wmAP instead of mAP, as the test set is extremely imbalanced and would fluctuate wildly otherwise (see fluctuations in columns "under" and "hits"). We also report score$_{wtd}$, which is the official OI scoring formula but with wmAP in place of mAP. "Under" and "hits" are not highlighted due to having too few instances.

	R@50	wmAP$_{rel}$	wmAP$_{phr}$	score$_{wtd}$	AP$_{rel}$ per class					AP$_{phr}$ per class				
					at	holds	plays	interacts_with	wears	at	holds	plays	interacts_with	wears
L_0	61.72	25.80	33.15	35.92	14.77	26.34	42.51	21.33	21.03	21.76	35.88	**48.57**	38.74	31.92
$L_0 + L_1 + L_2 + L_3$	**62.65**	**27.37**	**34.58**	**37.31**	**16.18**	**30.39**	**42.73**	**22.40**	**22.14**	**22.67**	**39.60**	48.09	**40.96**	**32.64**

Table 5.4: Comparison of our model with Graphical Contrastive Loss vs. without the loss on 100 images containing the 5 classes that suffer from the two aforementioned confusions, selected via visual inspection on a random set of images.

of numbers using mAP and score demonstrates the necessity of de-emphasizing the extremely infrequent classes. Note that the mAP*-based columns show a similar trend to our wmAP-based results from the paper. In Table 5.2, the model with our losses is still better than the L_0-only model by a non-trivial margin, mainly because the former outperform the latter on almost every per-class AP metric for those 5 selected classes. Note that since "under" and "hits" are not in the 100 image subset, there is no need to evaluate with mAP^*_{rel}, mAP^*_{phr} and $score^*$.

5.5.4 Model Analysis

We conduct an effectiveness evaluation on the three modules of the RelDN. For the visual module, we also investigate the two skip-connections. As Table 5.5 shows, the semantic module alone cannot solve relationship detection by using language bias only. By adding the basic visual feature, *i.e.*, the $\langle$S,P,O$\rangle$ concatenation, we see a significant 4.7 gain, which is further improved by adding additional separate S,O skip-connections, especially at "plays" (+3.1), "interacts_with" (+1.0), "wears" (+2.0) where subjects' or objects' appearance and poses are highly representative of the interactions. Finally, adding the spatial module gives the best results, and the most obvious gaps are at spatial relationships, *i.e.*, "at" (+0.2), "on" (+0.2), "inside_of" (+2.4).

					AP_{rel} per class								
	R@50	$wmAP_{rel}$	$wmAP_{phr}$	$score_{wtd}$	at	on	holds	plays	interacts_with	wears	inside_of	*under*	*hits*
sem only	72.98	28.73	33.07	39.32	28.62	24.52	37.04	27.33	38.37	3.16	16.34	*25.00*	*38.45*
sem + ⟨S,P,O⟩	74.97	34.70	37.96	44.06	32.26	36.26	42.44	38.47	41.63	6.50	40.97	*20.00*	*54.38*
sem + vis	**75.12**	35.22	38.33	44.44	32.68	36.83	42.09	**41.53**	42.58	**8.49**	42.31	*33.33*	*53.95*
sem + vis + spt	74.94	**35.54**	**38.52**	**44.61**	**32.92**	**37.00**	**43.09**	41.04	**44.16**	7.83	**44.72**	*50.00*	*51.04*

Table 5.5: Ablation Study on RelDN modules. *sem only* means using only the semantic module without training any model; ⟨*S,P,O*⟩ means using only the ⟨S,P,O⟩ concatenation without the separate S,O layers in the visual module; *vis* means our full visual module, and *spt* means spatial module. "Under" and "hits" are not highlighted due to having too few instances.

	R@50	$wmAP_{rel}$	$wmAP_{phr}$	$score_{wtd}$
m = 0.1	**75.09**	35.29	38.43	44.51
m = 0.2	74.94	**35.54**	**38.52**	**44.61**
m = 0.5	74.64	35.14	38.39	44.34
m = 1.0	74.28	34.17	37.75	43.62

Table 5.6: Ablation Study on the margin threshold m. We use $m = 0.2$ everywhere in our experiments.

	Graph Constraint									No Graph Constraint					
	SGDET			SGCLS			PRDCLS			SGDET		SGCLS		PRDCLS	
Recall at	20	50	100	20	50	100	20	50	100	50	100	50	100	50	100
VRD[62]	-	0.3	0.5	-	11.8	14.1	-	27.9	35.0	-	-	-	-	-	-
Associative Embedding[76]	6.5	8.1	8.2	18.2	21.8	22.6	47.9	54.1	55.4	9.7	11.3	26.5	30.0	68.0	75.2
Message Passing[114]	-	3.4	4.2	-	21.7	24.4	-	44.8	53.0	-	-	-	-	-	-
Message Passing+[132]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	22.0	27.4	43.4	47.2	75.2	83.6
Frequency[132]	17.7	23.5	27.6	27.7	32.4	34.0	49.4	59.9	64.1	25.3	30.9	40.5	43.7	71.3	81.2
Frequency+Overlap[132]	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2	28.6	34.4	39.0	43.4	75.7	82.9
MotifNet-NOCONTEXT[132]	21.0	26.2	29.0	31.9	34.8	35.5	57.0	63.7	65.6	29.8	34.7	43.4	46.6	78.8	85.9
MotifNet-LeftRight[132]	**21.4**	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	**30.5**	35.8	44.5	47.7	81.1	88.3
RelDN, L_0 only	20.8	28.1	32.5	**36.1**	36.7	36.7	66.7	68.3	68.3	30.1	36.4	**48.9**	**50.8**	93.7	97.7
RelDN	21.1	**28.3**	**32.7**	**36.1**	**36.8**	**36.8**	**66.9**	**68.4**	**68.4**	30.4	**36.7**	**48.9**	**50.8**	**93.8**	**97.8**
RelDN (X-101-FPN)	22.5	31.0	36.7	38.2	38.9	38.9	67.2	68.7	68.8	32.6	40.0	51.7	53.6	94.0	97.8

Table 5.7: Comparison with state-of-the-arts on VG. L_0 **only** is the RelDN without our losses. We also include results of our model with ResNeXt-101-FPN as the backbone for future work reference.

	Relationship		Phrase		Relationship Detection						Phrase Detection					
	free k				k = 1		k = 10		k = 70		k = 1		k = 10		k = 70	
Recall at	50	100	50	100	50	100	50	100	50	100	50	100	50	100	50	100
PPRFCN*[135]	14.41	15.72	19.62	23.75	-	-	-	-	-	-	-	-	-	-	-	-
VTransE*	14.07	15.20	19.42	22.42	-	-	-	-	-	-	-	-	-	-	-	-
SA-Full*[82]	15.80	17.10	17.90	19.50	-	-	-	-	-	-	-	-	-	-	-	-
DR-Net*[13]	17.73	20.88	19.93	23.45	-	-	-	-	-	-	-	-	-	-	-	-
ViP-CNN[55]	17.32	20.01	22.78	27.91	17.32	20.01	-	-	-	-	22.78	27.91	-	-	-	-
VRL[56]	18.19	20.79	21.37	22.60	18.19	20.79	-	-	-	-	21.37	22.60	-	-	-	-
CAI*[146]	20.14	23.39	23.88	25.26	-	-	-	-	-	-	-	-	-	-	-	-
KL distilation[128]	22.68	31.89	26.47	29.76	19.17	21.34	22.56	29.89	22.68	31.89	23.14	24.03	26.47	29.76	26.32	29.43
Zoom-Net[122]	21.37	27.30	29.05	37.34	18.92	21.41	-	-	21.37	27.30	24.82	28.09	-	-	29.05	37.34
CAI + SCA-M[122]	22.34	28.52	29.64	38.39	19.54	22.39	-	-	22.34	28.52	25.21	28.89	-	-	29.64	38.39
RelDN, L_0 only (ImageNet)	21.62	26.12	28.59	35.18	19.57	22.61	21.62	26.12	21.62	26.12	26.39	31.28	28.59	35.18	28.59	35.18
RelDN (ImageNet)	21.52	26.38	28.24	35.44	19.82	22.96	21.52	26.38	21.52	26.38	26.37	31.42	28.24	35.44	28.24	35.44
RelDN, L_0 only (COCO)	26.67	32.55	33.29	41.25	24.30	27.91	26.67	32.55	26.67	32.55	31.09	**36.42**	33.29	41.25	33.29	41.25
RelDN (COCO)	**28.15**	**33.91**	**34.45**	**42.12**	**25.29**	**28.62**	**28.15**	**33.91**	**28.15**	**33.91**	**31.34**	**36.42**	**34.45**	**42.12**	**34.45**	**42.12**

Table 5.8: Comparison with state-of-the-art on VRD (− means unavailable / unknown). Same with Table 5.7, L_0 **only** is the RelDN without our losses. "Free k" means considering k as a hyper-parameter that can be cross-validated.

5.5.5 Comparison to State of the Art

OpenImages: We present results compared with top 5 models from the Challenge in Table 5.9.

We surpass the 1st place *Seiji* by 4.7% on Private set and 2.9% on the full set, which is in fact

Team ID	Public	Private	Overall
radek	0.289	0.201	0.227
toshif	0.256	0.228	0.237
tito	0.256	0.237	0.243
Kyle	0.280	0.235	0.249
Seiji	0.332	0.285	0.299
RelDN*	0.327	0.299	0.308
RelDN	0.320	0.332	**0.328**

Table 5.9: Comparison with models from OpenImages Challenge. RelDN* means using the same entity detector from *Seiji*, the champion model. Overall is computed as 0.3*Public+0.7*Private. Note that this table uses the official mAP$_{rel}$ and mAP$_{phr}$ metrics.

a significant margin considering the low absolute scores and the large amount of test images (99,999 in total). Even using the same entity detector as *Seiji*, we noticeable gaps (1.4% and 0.8%) on the two sets.

Visual Genome: Table 5.7 shows that our model is better than state-of-the-arts on all metrics. It outperforms the previous best, MotifNet-LeftRight, by a 2.4% gap on Scene Graph Detection (SGDET) with Recall@100 and by a 12.7% gap on Predicate Classification (PRDCLS) with Recall@50. Note that although our entity detector is better than MotifNet-LeftRight on mAP at 50% IoU (25.5 vs. 20.0), our implementation of Frequency+Overlap baseline (Recall@20: 16.2, Recall@50: 19.8, Recall@100: 21.5) is not better than their version (Recall@20: 21.0, Recall@50: 26.2, Recall@100: 30.1), indicating that our better relationship performance mostly comes from our model design.

We also observe that our losses achieve smaller gains over the standard cross-entropy loss setup than it does on OpenImages_mini. The reasons are two-fold: 1) One of the few dominant relationship types in the Visual Genome dataset is possessive, e.g., "ear of man", which has much fewer entity confusion issues; 2) The $Recall@k$ metric is less strict than mAP. If there is an image with only one ground truth, then Recall@100 will always be 100% as long as this ground truth target is within the top 100 model predictions, regardless of the ranking of the 100 outputs. As such, the small improvements in ranking the top 100 will not affect the score. Nevertheless, the improvements from our loss is still non-trivial and consistent on all metrics under different values of k.

In addition, we also show results using a better backbone, ResNeXt-101-FPN [113, 58], for the entity detector in Table 5.7.

VRD: Table 5.8 presents results on VRD compared with state-of-the-art methods. Note that

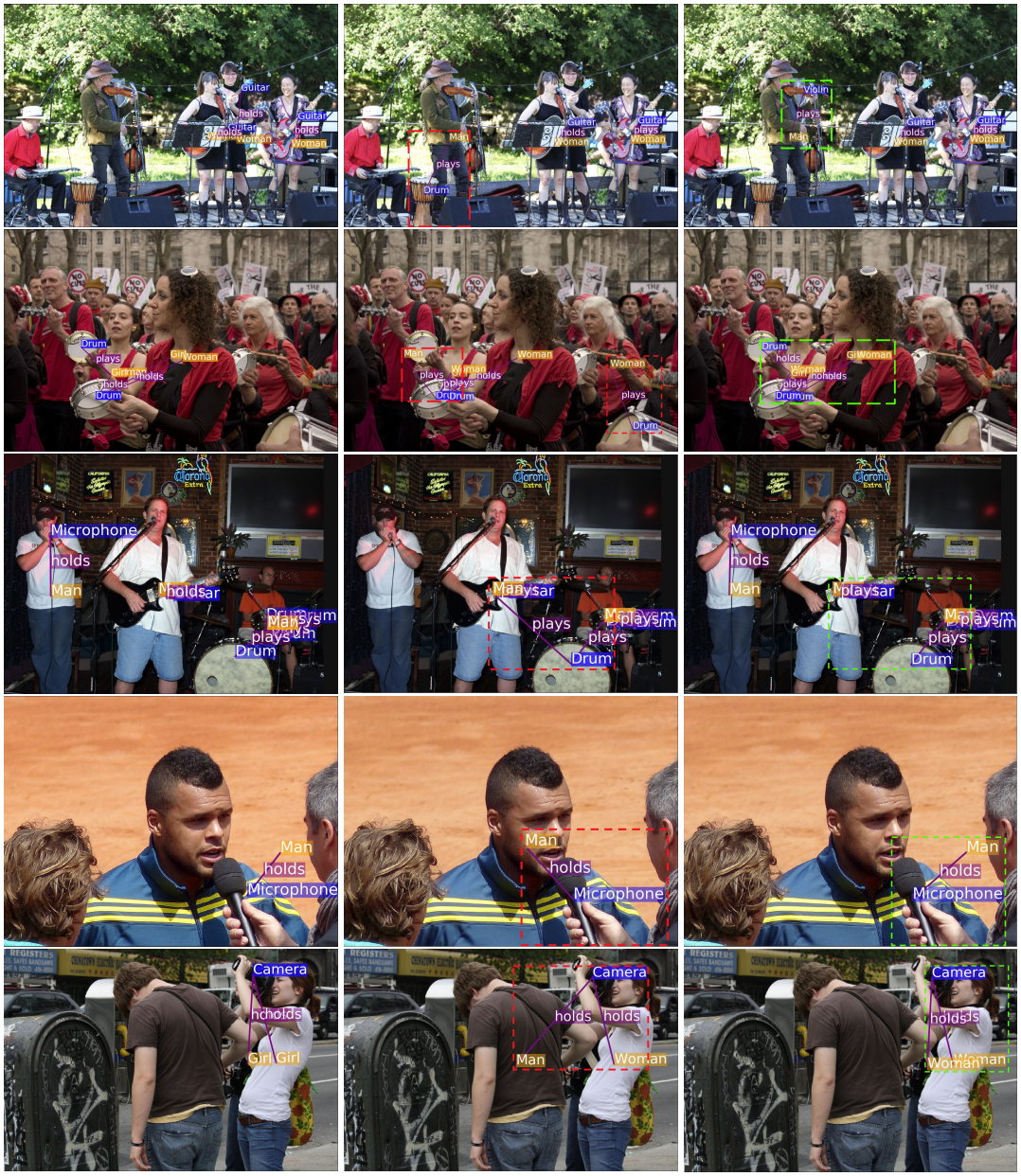
only [122] specifically states that they use ImageNet pre-trained weights while others remain unknown. Therefore, we show results for pre-training on either ImageNet or COCO. Our model is competitive with those methods when pre-trained on ImageNet, but significantly outperforms when pre-trained on COCO. The gap between L_0 *only* and the full model is smaller when pre-trained on ImageNet than on COCO. We believe the stronger localization features from pre-training on COCO is much easier for our model and losses to leverage.

5.5.6 Qualitative Results

In Figure 5.6 we provide four example images where our losses correct the false predictions made by the L_0 only model. Both the Entity Instance Confusion and the Proximal Relationship Ambiguity issues are included here. In the fourth row, the L_0 only model is confused between two entity instances, *i.e.*, which person is holding the microphone, while our losses manage to refer to the correct one. In the third row the relationship between the guitar player and the drum is ambiguous. Here, the L_0 only model fails by predicting a false-positive, but our model trained with all losses correctly detects no relationship there.

5.6 Summary

In this work we present methods to overcome two major issues in scene graph parsing: Entity Instance Confusion and Proximal Relationship Ambiguity. We show that traditional multi-class cross-entropy loss does not take advantage of intrinsic knowledge of structured scene graphs and is therefore insufficient to handle these two issues. To address that, we propose Graphical Contrastive Losses which effectively utilize semantic properties of scene graphs to contrast positive relationships against hard negatives. We carefully design three types of losses to solve the issues in three aspects. We demonstrate efficacy of our losses by adding it to a model built with the same pipeline, and we achieve state-of-the-art results on three datasets.



(a) ground truth (b) L_0 only (c) all losses

Figure 5.6: Example images where RelDN with only L_0 predicts incorrectly while our loss succeeds. For each image we check the number of its ground truth relationships, then we output the same number of top predictions from a model to see its ranking accuracy. Red boxes in (b) highlight the false predictions from RelDN with L_0 only and green boxes in (c) highlight the correct ones from RelDN with all losses.

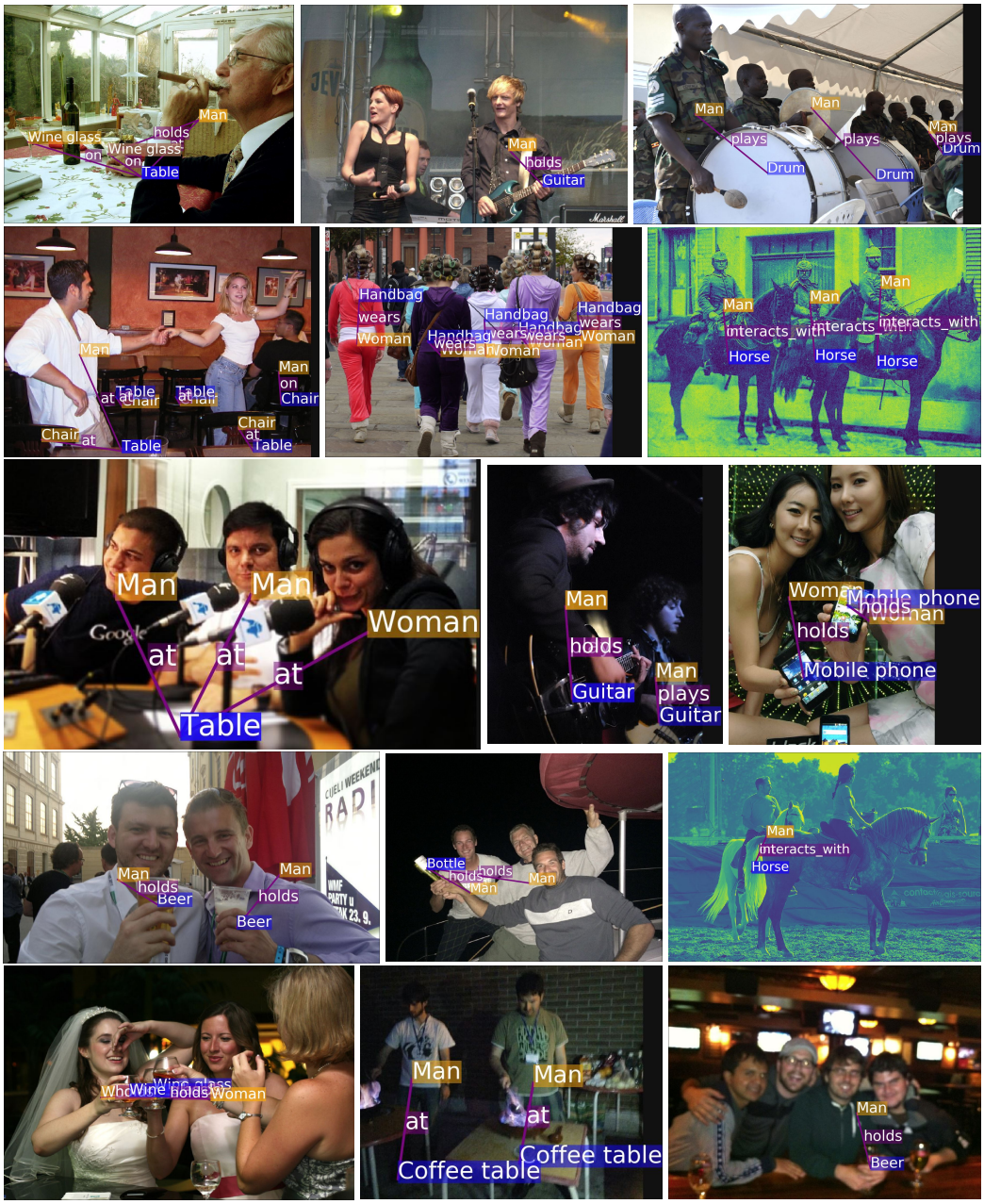


Figure 5.7: Example images of the 100 image subset with ground truth relationships. The subset contains five predicates where the Entity Instance Confusion and Proximal Relationship Ambiguity commonly occur.

Chapter 6

Video Story Understanding with Character-Aware Relations

Different from short videos and GIFs, video stories contain clear plots and lists of principal characters. Without identifying the connection between appearing people and character names, a model is not able to obtain a genuine understanding of the plots. Video Story Question Answering (VSQA) offers an effective way to benchmark higher-level comprehension abilities of a model. However, current VSQA methods merely extract generic visual features from a scene. With such an approach, they remain prone to learning just superficial correlations. In order to attain a genuine understanding of who did what to whom, we propose a novel model that continuously refines character-aware relations. This model specifically considers the characters in a video story, as well as the relations connecting different characters and objects. Based on these signals, our framework enables weakly-supervised face naming through multi-instance co-occurrence matching and supports high-level reasoning utilizing Transformer structures. We train and test our model on the six diverse TV shows in the TVQA dataset, which is by far the largest and only publicly available dataset for VSQA. Our experiments show that our approach achieves new state-of-the-art results.

6.1 Introduction

Video stories such as TV shows and movies entertain us and enrich our life. We can easily understand the plots and become addicted to the acting of protagonists. However, video story understanding remains a challenging task for artificial intelligence. In this paper, we argue that characters in two aspects play an important role for a better comprehension of video stories. On the one hand, characters lie at the intersection of video and text/subtitle modalities. On the other hand, they are the pivots of plots, embodying who did what to whom.

The task of VSQA is a convincing means of measuring how well a model understands a

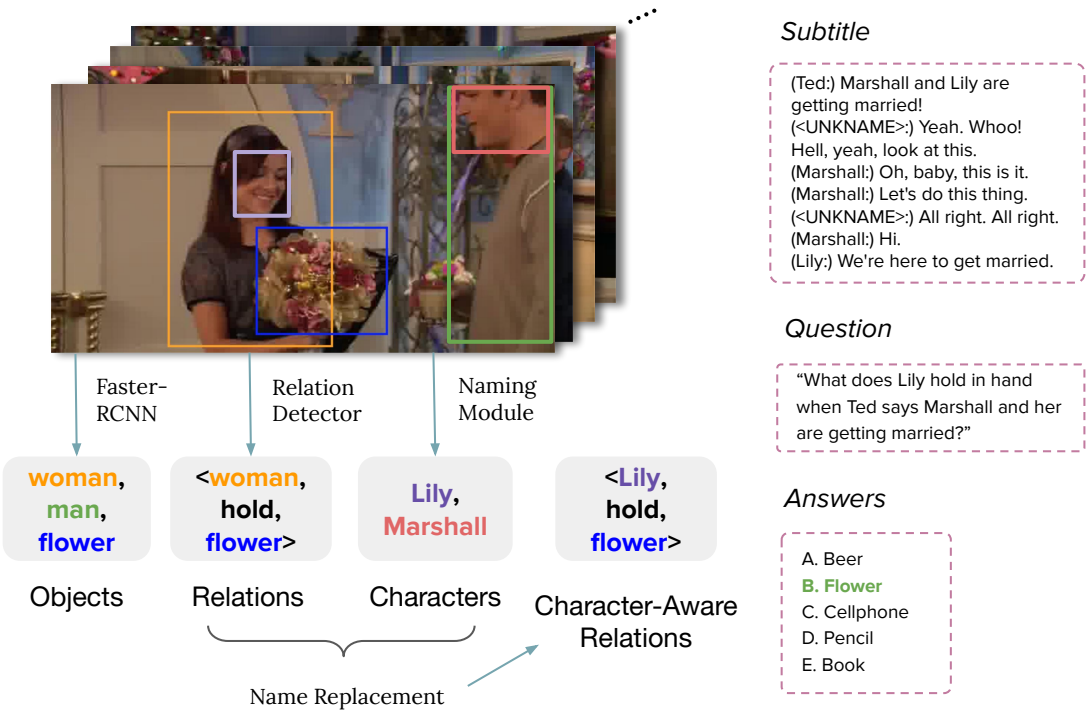


Figure 6.1: Illustration of how fine-granular features may help in VSQA. Character names, visual objects, and their relationships are all necessary factors in answering this question. Character-aware relationships are detected in video frames, where references to humans such as "woman" and "man" are replaced with predicted character names, determined by finding the face bounding box that overlaps the most with the human bounding box.

video. Typically, it is solved in three steps: 1) extracting key features of multimodal contents; 2) fusing those multimodal features; 3) utilizing the fused features to predict the correct answer to a question. For the first step, current state-of-the-art methods [73, 105, 44, 54, 40, 41] mainly focus on global visual features at the image level. In particular, they consider one or more frames as input and extract features that provide a holistic representation of the frames. As a result, a basic understanding of what occurs in the frames is achieved, but substantially meaningful details may be missed due to the coarse granularity of the global features. Such details include individual objects, their relationships and attributes, and perhaps more importantly, the identities of people inside the video. These aspects are often crucial for answering semantic questions such as "What does Lily hold in hand when Ted says Marshall and her are getting married?" (Figure 6.1). Here, the flower (object) that Lily (character) is holding (relationship) are the key factors needed to answer the question, but generic global features usually have very limited power to capture them. Such limitations motivate us to design a framework that focuses on fine-grained visual cues and provides richer knowledge of the depicted scenes.

Several recent works [3, 50, 121] have followed this direction and explored various approaches to incorporate grounded visual features for questions answering tasks. Despite their success, the video story setting is quite different in that characters' identities, especially those of main actors, tend to matter much more than in static images, since they keep reappearing. Moreover, video stories involve a much larger number of interactions between characters, such as "Robin talks to Ted", or between characters and objects, such as "Lily holds flowers". It is very difficult for a model to achieve a genuine understanding of a scene without capturing these character-involved associations. Therefore, we need a better framework that has the capability to mine both detailed visual cues about character identities and their relationships.

To address the above issues, we build a VSQA framework accounting for character-centric relations and a character-aware reasoning network (CA-RN) that combines and connects those features with reasoning abilities. Our framework consists of two main parts. The first part aims at building a scene representation for understanding relations between characters and objects so as to infer what is going on. Through visual relations, we capture two levels of visual semantics: the entity level and the relation level. At the entity level, we detect characters, objects, and their attributes via pre-trained object detectors and multi-instance co-occurrence matching based character identification. At the relation level, the relations between the entities are recognized within each frame, where human-referring words are replaced with predicted character names. For the second part, the multi-modal information (including two-level scene representation and subtitles) are then injected into our Transformer-based CA-RN network, which serves as the semantic reasoning module.

We train and test our model on the six diverse TV Shows from a large-scale video story dataset TVQA [50]. In each video clip, there are corresponding subtitles and several multiple choice questions. The goal of our framework is to correctly predict the right answers to these questions. The key contributions of this paper can be summarized as follows:

- We propose an end-to-end multi-task framework to mining the face–speaker associations and conduct multi-modal reasoning at the same time. It enables weakly-supervised face naming through co-occurrence matching and supports high-level reasoning through Transformer structures.

- We propose to utilize character-aware relations as a stronger representation of visual knowledge of scenes. To the best of our knowledge, this is the first attempt to apply such a strategy to video question answering tasks.
- Experiments on six TV shows confirm that our approach outperforms several state-of-the-art baselines while also offering explicit explanations, especially for those questions that require a deep understanding of video scenes.

6.2 Method

The goal of this work is to make use of the co-occurrence of faces in videos and names in subtitles to continuously refine the detection of character-aware relationships, and finally use the latter for improved video story understanding. As shown in Figure 4.2, our video story understanding framework can be trained in an end-to-end manner and consists of two main modules, one of which predicts the detected face bounding boxes and incorporates character names into the detected relationships by matching the locations of bounding boxes. As a result, multiple forms of visual semantics from each frame are extracted and combined together as an understanding of the scene that the characters are acting in. The other module is a sequential Transformer-based reasoning pipeline, which takes in the input question, answer options, and different modalities, and outputs the predicted answer with the highest softmax score. In the following, we describe the methods to extract character-aware visual semantics and conduct multi-modal reasoning.

6.2.1 Character-Aware Frame Understanding

Face Detection and Feature Extraction. We utilize a state-of-the-art face detector [143] to localize faces in each frame and extract their 256-dimensional visual features $f \in \mathbb{R}^{256}$ using LightCNN [112], considering its effectiveness at general face identification, i.e., identifying different faces of the same person. This is a desirable feature, as we need to distinguish the faces of different people while neglecting the variance within the various appearances of a given person.

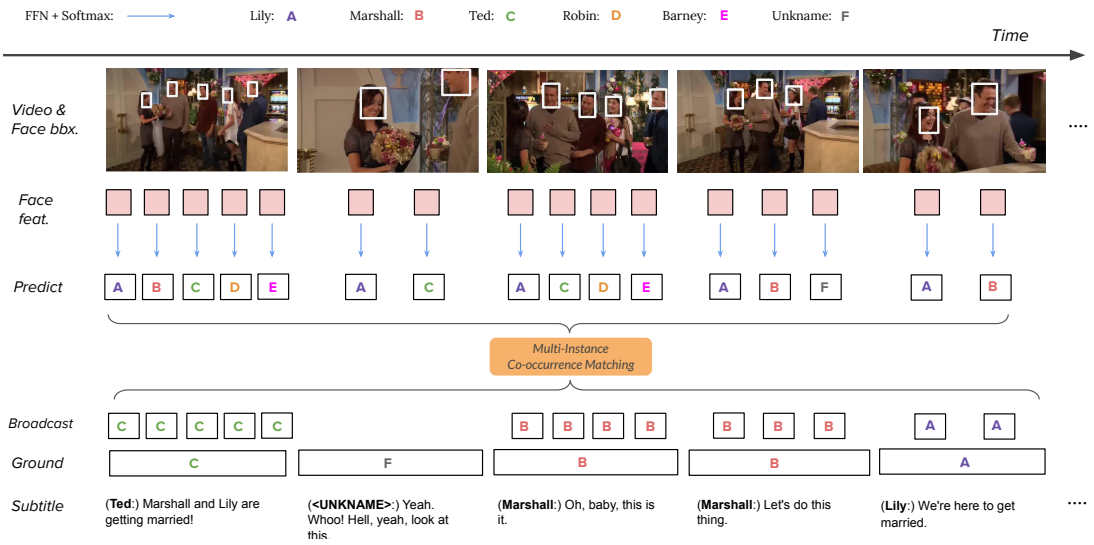


Figure 6.2: An illustration of our weakly-supervised character identification pipeline. The face bounding boxes of all characters are first detected. The extracted face features are then predicted by fully-connected feed-forward layer and Softmax. After broadcasting the character names in subtitle to be a distribution sequence that has the same length of predicted name distribution sequence, a weight KL divergence loss is utilized to conduct multi-instance co-occurrence matching.

Weakly Supervised Character Identification. In order to recognize characters without explicit face name annotations, we first determine the number k of principal characters in the TV series (details in Section 6.2.4). Supporting actors are lumped together as an UNKNAME class here, as they have a smaller impact on the main plot lines. Assume there are n detected faces in a video clip with features $\mathcal{F} = \{f_1, ..., f_n\}$. We first utilize a naming module consisting of several fully connected feed-forward layers and softmax to get a confidence distribution over all names in the character list:

$$p_i = \text{softmax}(W_2 \text{ReLU}(W_1 f_i + b_1) + b_2)) \tag{6.1}$$

where W_1, W_2, b_1, b_2 are the weights and biases of fully-connected layers, and p_i is the confidence distribution over all names in the character list. By doing so for all detected faces, we can construct a sequence of predicted character name distributions: $\mathcal{P} = \{p_1, p_2, ..., p_n\}$. As shown in Figure 6.2, the speaker names in the subtitle maintain a multi-instance co-occurrence with the character faces in the video. Inspired by this, we duplicate the current speaker name to be of the same number as the detected faces in each frame to serve as weak supervision. Note that if the speaker in the subtitle is UNKNAME, the frame will not have a broadcast operation. The

ground character name distribution can be represented by $\mathcal{G} = \{g_{\mathrm{Loc}(1)}, g_{\mathrm{Loc}(2)}, ..., g_{\mathrm{Loc}(n)}\}$, where the localization function $\mathrm{Loc}()$ maps the face bounding box ID to the frame ID, and g_l denotes the one-hot ground name distribution of frame l. Afterwards, the multi-instance co-occurrence matching can be conducted by a regularized Kullback-Leibler divergence between predicted and ground character name distributions:

$$D_{\mathrm{RKL}}(\mathcal{P} \parallel \mathcal{G}) = \sum_{l}^{L} \min_{j \in \mathrm{F}_l} \mathcal{P}(j) \ln \frac{\mathcal{P}(j)}{\mathcal{G}(j)} \tag{6.2}$$

where F_l is the set of faces in frame l. This loss is similar in spirit to Multiple Instance Learning. With this, the model learns to assign the speaker name to the corresponding face, which will minimize the loss.

Regular Visual Object Contexts. We use regular objects and attributes as another form of visual semantics for each frame, similar to [50]. Specifically, we apply Faster-RCNN [87] trained on Visual Genome [48] to detect all objects and their attributes. Object bounding boxes are discarded, as we are targeting pertinent semantic information from the frame.

Incorporating Characters into Visual Semantics. Once we have localized and recognized character faces and names, as well as regular objects, we append each character name detected in a given frame to each of the objects detected in the same frame to augment its visual semantics. We found that this simple strategy works very well as shown in Section 6.3.5, due to the fact that character names are anchors that allow for localizing and disentangling semantic information. For example, visual objects without names attached, such as "food, wine glass", are fairly generic, while "food+Leonard" and "wine glass+Penny" better allow for distinguishing objects from different frames and associating them with relevant people, and hence provide a clearer picture of what is included in the scene.

6.2.2 Character-Aware Relation Detection

This component is designed to extract all relationships that involve the main characters in the scene. We decompose this task into two steps: 1) detecting relationships to build scene graphs; 2) replacing detections of humans in relationships with specific characters.

General Relation Detection. The relation detection module aims at detecting all related objects and recognizing their relationships in a video. We use the approach by [137] to detect relationships in each frame. Specifically, we train the model on the VG200 dataset, which contains 150 objects and 50 predicates, and apply this model on all frames of a given video. The result is a set of $\langle S, P, O\rangle$ triples per frame, where S, P, O represent the *subject*, *predicate*, *object*, respectively. We only keep the $\langle S, P, O\rangle$ triples and discard subject/object boxes since 1) their spatial locations carry little signal about the scene semantics; 2) there are already many spatial relationships among the VG200 predicates, such as "above" and "under". Once the model manages to predict these relations, we immediately know the relative spatial relations between subjects and objects, which we found is sufficient to describe the scenes. We concatenate all the triples in the current frame into a sequence of $N_r \times 3$ (where N_r is the number of $\langle S, P, O\rangle$ triples) and feed it to the following modules.

Character Name Replacement. Given all the faces and relations in a scene, we focus on those relations with human-referring words as subjects or objects, such as "woman" or "man". For each of these human bounding boxes, we obtain the face box that overlaps the most with it to determine the human's face. Once this matching is done for all human boxes in the frame, we replace the human-referring words in those relationships with the previously identified character names. This makes the relationships more concrete, as we know exactly who is involved in each relationship. Figure 6.1 shows examples of detected relationships from frames of *HIMYM*, where human-referring words are replaced with the specific character names. We show in Section 6.3.5 that by applying this name replacement to all relationships, the model is able to capture details that are strongly associated with the question and thus engender more accurate answers.

6.2.3 Character-Aware Reasoning Network

As shown in Figure 4.2, CA-RN model works in an end-to-end manner and updating based on a multi-task loss function. It takes in question, answer options, subtitles, face bounding boxes, and visual semantics, and then outputs the probabilities of each answer option. Next, we describe the details of our Transformer-based multi-modal reasoning model.

Encoder. We first embed the input question, answer options, and all modalities (subtitles and

visual semantics) $\mathcal{I} = \{q, a_{0-4}, s, v_{o,r}\}$ using word and name embeddings. We denote the set of these embeddings as: $\mathcal{E} = \{e_q, e_a, e_s, e_v\}$. They are then fed into a two-layer Transformer encoder consisting of self-attention with four heads to capture long-range dependencies and a bottle-neck feed-forward networks to obtain encoded hidden representations:

$$h_j = \mathrm{FFN}(\mathrm{Attention}(e_j)), \tag{6.3}$$

where $j \in \{q, a, s, v\}$, FFN is a feed-forward module consisting of two FC layers with ReLU in between, and the Attention function here is defined as [102]:

$$\mathrm{Attention}(Q, K) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_h}})K. \tag{6.4}$$

Here $\sqrt{d_h}$ is a scaling factor used to maintain scalars in the order of magnitude and d_h is each head's hidden dimensionality.

Multi-Modal Decoder. Once all inputs are encoded by the encoder, we utilize sequential co-attention decoders to fuse their information and generate an updated representation of the question and answer options. For simplicity, we take visual relations h_r and subtitles h_s as two input modalities. The following framework can easily be extended to more input modalities. As shown in Figure 4.2, the visual relations, question, and answer options are first fed into the a two-layer four-heads co-attention decoder to acquire context-aware-QA representations. Then, subtitles and updated QA representations serve as the input for another co-attention decoder with the same structure. The co-attention decoder can be represented as:

$$h_{c \to i} = \mathrm{FFN}(\mathrm{Attention}(h_i, h_c)), \tag{6.5}$$

where $h_{c \to i}$ is the context-aware-QA representation, and h_c, h_i represent contextual and input QA hidden representations, respectively. Afterwards, context-aware QA representations for different question–answer pairs are then concatenated and processed by a self-attention decoder

w/ ts	Test-Public							Val
Show	BBT	Friends	HIMYM	Grey	House	Castle	All	All
NNS-SkipThought [50]	-	-	-	-	-	-	38.29	38.41
NNS-TFIDF [50]	-	-	-	-	-	-	50.79	51.62
Multi-Stream V only [50]	-	-	-	-	-	-	43.69	-
Multi-Stream [50]	70.19	65.62	64.81	68.21	69.70	69.79	68.48	68.85
T-Conv [123]	71.36	66.52	**68.58**	69.22	67.77	68.65	68.58	68.47
CA-RN (Ours)	**71.89**	**68.02**	67.99	**72.03**	**71.83**	**71.27**	**70.59**	**70.37**
Human	-	-	-	-	-	-	91.95	93.44

Table 6.1: Results on the TVQA test set for models that use time-stamp annotation ('w/ ts'). We compare to other baselines on the six TVQA sub-datasets individually. "V only" means using only global CNN features without subtitles.

w/o ts	Test-Public							Val
Show	BBT	Friends	HIMYM	Grey	House	Castle	All	All
NNS-SkipThought [50]	-	-	-	-	-	-	26.93	27.50
NNS-TFIDF [50]	-	-	-	-	-	-	49.59	50.33
Multi-Stream V only [50]	-	-	-	-	-	-	42.67	-
Multi-Stream [50]	70.25	65.78	64.02	67.20	66.84	63.96	66.46	65.85
T-Conv [123]	67.38	63.97	62.17	65.19	65.38	67.88	65.87	65.85
PAMN [41]	67.65	63.59	62.17	67.61	64.19	63.14	64.61	64.62
ES-MTL [40]	69.60	**65.94**	64.55	68.21	66.51	66.68	67.05	66.22
CA-RN (Ours)	**71.43**	65.78	**67.20**	**70.62**	**69.10**	**69.14**	**68.77**	**68.90**
Human	-	-	-	-	-	-	89.41	89.61

Table 6.2: Results on the TVQA test set for models that do not use time-stamp annotations ('w/o ts'). We compare to other baselines on the six TVQA sub-datasets individually. "V only" means using only global CNN features without subtitles.

to get the final representation for softmax calculation:

$$\begin{aligned} M &= \text{Concat}_{i\in\{(q,a_0),\ldots,(q,a_4)\}} h_{c\to i} \\ p_a &= \text{softmax}(\text{FFN}(\text{Attention}(M))) \end{aligned} \tag{6.6}$$

Finally, we are able to predict the answer y with the highest confidence score $y = \text{argmax}_{a\in\{a_0,\ldots,a_4\}} p_a$.

Multi-Task Loss Function. The feed-forward network in the naming module and the multi-modal reasoning module are jointly trained in an end-to-end manner through the following multi-task loss function:

$$\begin{aligned} \mathcal{L}_{\text{multi-task}} &= \mathcal{L}_{\text{cross-entropy}} + \lambda \mathcal{L}_{\text{mi-co}} \\ &= -\sum_{c=1}^{5} g_c \log(p_c) + \lambda D_{\text{RKL}}(\mathcal{P} \parallel \mathcal{G}), \end{aligned} \tag{6.7}$$

which combines $\mathcal{L}_{\text{cross-entropy}}$ for question answering and $\mathcal{L}_{\text{mi-co}}$ for multi-instance co-occurrence matching, linked together by a hyperparameter λ.

6.2.4 Implementation Details

Principal Character List. We focus on naming the faces of principal characters, since they are highly correlated to the story-line of TV shows. The number k for the character list for each TV show is determined in the following three steps: 1) count the occurrences of all speakers in the subtitles; 2) select all names appearing more than 500 times as principal character candidates; 3) filter out names that make up less than 1/10 of the speakers with the highest occurrence. 4) An additional UNKNAME class is assigned to all other character names not in the principal list. Note that we do not rely on external information of who the principal characters are.

Text and Name Embeddings. After parsing the scene into character-aware relations, we have four types of features to transform into text embeddings: the subtitles, visual semantics, questions, and the candidate answers. Note that once the visual semantics are extracted, we do not need any visual features from the frames. Since the character names are different from their literal meanings (e.g., the character Lily is different from the regular word *lily*), we utilize two separate embeddings. For ordinary words, we rely on 300-dimensional GloVe word vectors [81] to embed the words after tokenization. For character names, we train and update their name embeddings from scratch. In the case of out of vocabulary words, we use averaged character vectors of the words.

Model Training. Our model is trained with Adam stochastic optimization on Tesla V100 GPUs. The λ in the multi-task loss function is simply set to 1. In the training process, we set the batch size as 64. The learning rate and optimizer settings are borrowed from [50].

6.3 Experiments

6.3.1 Dataset

The recently released TVQA dataset [50] is a large-scale video question answering dataset based on 6 popular TV shows: 3 situation comedies (The Big Bang Theory, Friends, How I Met Your Mother), 2 medical comedies (Grey's Anatomy, House M.D.), and 1 crime comedy (Castle). It consists of 152.5K QA pairs (84.8K what, 17.7K who, 17.8K where, 15.8K why, 13.6K how questions) from 21.8K video clips, spanning over 460 hours of video. Each video clip is associated with 7 questions and a dialogue text (consisting of character names

and subtitles). The questions in the TVQA dataset are designed to be compositional in the format "[What/How/Where/Why/...] ____ [when/before/after] ____" and require both visual and language comprehension.

6.3.2 Baselines

We consider several baselines for performance comparison.

Nearest Neighbor Search. These baselines (NNS-TFIDF and NNS-SkipThought) are taken from the original TVQA paper [50]. They compute the cosine similarity between the resulting vectors to predict the answer.

Multi-Stream. [50] combines information from different modalities with LSTMs and cross-attention. The results stem from the official TVQA leaderboard.

Temporal Convolution. It has recently been shown that temporal convolutions (T-Conv) [73, 123] can be a strong alternative to traditional RNN layers for question answering. We follow the structure from [123] and build the T-Conv baseline by replacing the LSTM layers in [50] with temporal convolutions while keeping other modules unchanged.

PAMN. [41] utilize progressive attention memory to update the belief for each answer. This is also from the official TVQA leaderboard.

ES-MTL. [40] explores two forms of extra supervision for temporal localization and modality alignment. This is the strongest baseline from the official TVQA leaderboard without any additional object-level annotations.

Human Performance. We also give the human results as reported along with the dataset [50] as a reference to gauge how big the gap is to human intelligence.

6.3.3 Experimental Setup

We use the top-1 accuracy as the only metric, following the official guidelines. There are two types of settings we can adopt from the official evaluation rules [50]: with time stamps (w/ ts) and without time stamps (w/o ts), where time stamps refer to ground-truth annotations on the intervals of the video segments that relate the most to the given questions. The former setting assumes we have the time stamps in both training and testing, while in the latter case such

Method	Val Acc. w/ ts	Val Acc. w/o ts
Sub	66.23	66.14
Sub + Objs	68.85	67.35
Sub + Rels	67.55	67.16
Sub + Objs_nm	69.45	68.13
Sub + Rels_nm	68.25	67.85
Sub + Objs + Rels	69.54	68.44
Sub + Objs + Rels_nm	69.76	68.64
Sub + Objs_nm + Rels	70.20	68.68
Sub + Objs_nm + Rels_nm	**70.37**	**68.90**

Table 6.3: Ablation study both with and without the time stamps. "Sub", "Objs", "Rels" and "nm" represent subtitles, objects, relationships and names, respectively.

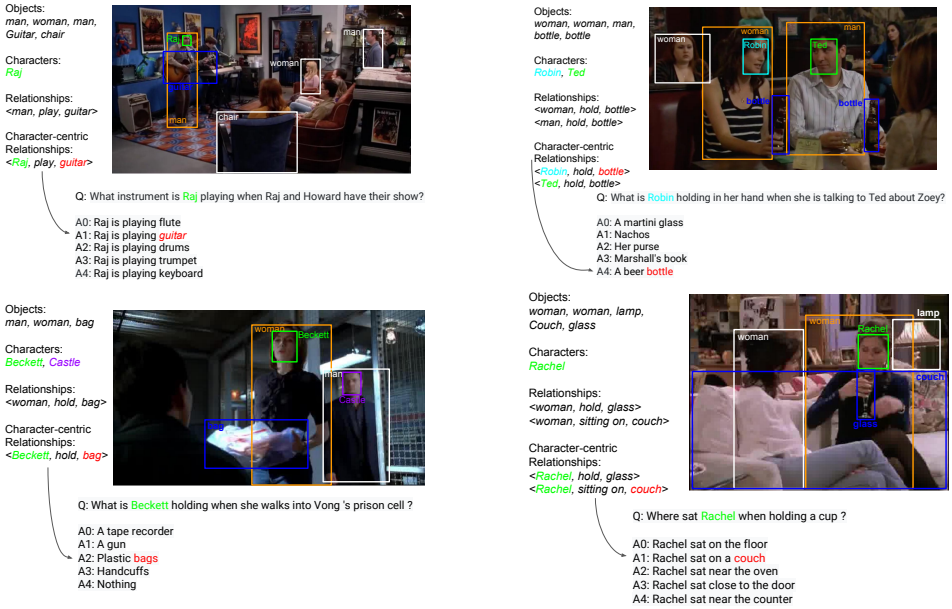


Figure 6.3: Examples of correctly answered questions that benefit from the proposed strategy. Orange and blue boxes are subjects and objects, while white boxes are objects with no detected relationships. Boxes with names are our detected characters, which substitute for the human-referring words in the relationships to obtain a character-aware understanding.

information is not provided. We consider both settings in our comparison with related work.

6.3.4 Comparison to State-of-the-Art

As presented in Tables 6.1 and 6.2, our approach outperforms the best previous method by $1.90/2.01\%$ (absolute) on the val/test set with time stamps, and by $2.68/1.72\%$ (absolute) on the val/test set without time stamps. Considering that there are 15,253 and 7,623 validation and test questions, respectively, the largest gains are $15,253 \times 2.68\% = 409$ and $7,623 \times 2.01\% =$

153 questions on the two sets, respectively. This establishes the strength of our multi-task character-aware reasoning network. We believe the reasons behind the performance boost are the following three: 1) the Transformer structure enables capturing longer dependencies within and between different modalities compared with traditional RNN structures, especially when there is a long subtitle. 2) The multi-task framework allows refining the multi-modal reasoning model and mining the correlation between faces and names at the same time, making the two tasks contribute to each other. 3) The character-aware relations offer more detailed information than global CNN features and enable a deeper scene understanding.

6.3.5 Ablation Study

For further analysis, we conduct an ablation analysis on our proposed model both with and without time stamps in Table 6.3. There are three forms of visual semantics that we incrementally combine together: objects (Objs), relationships (Rels), and character names (nm). We observe that using subtitles only gives a reasonably good result of 66.23% but is still significantly worse than approaches with visual semantics. When objects are added ("Sub+Objs"), the accuracy is boosted by 2.62% (absolute), while additional gains (0.6% absolute) occur with names added ("Sub+Objs_nm"). A further improvement (0.92%) is attained when character-centric relationships ("Sub+Objs_nm+Rels_nm") are integrated, which demonstrates our claim that character naming is a beneficial factor for better video story understanding. We also provide results for further settings where we remove one or two forms of visual semantics, as in "Sub+Rels" and "Sub+Rels_nm". These results are slightly worse than the counterparts using "Objs" instead of "Rels", for which we find two main causes: 1) Objects are usually more diverse than relations, since typically only a small subset of objects are related. 2) Object detectors are generally more accurate than relationship detectors, which makes the "Objs" semantics more reliable than "Rels".

6.3.6 Qualitative Results

In Figure 6.3, we present 4 examples of our model's results based on all forms of visual semantics. In the top right case, the question demands a deep understanding of the scene where Robin is sitting beside Ted and holding a beer bottle. The character-aware relations are particularly

	Validation w/ ts						
Method	what(55.62%)	who(11.55%)	where(11.67%)	how(8.98%)	why(10.38%)	other(1.80%)	all(100%)
Sub	63.47	69.40	57.58	71.02	79.91	75.55	66.23
Sub + Rels	64.80	69.07	62.70	72.04	**80.67**	75.91	67.55
Sub + Rels_nm	65.31	72.59	62.02	72.85	80.29	77.01	68.25
Sub + Objs	66.97	69.01	62.98	72.85	79.60	77.74	68.85
Sub + Objs_nm	67.57	73.04	62.58	71.02	80.35	**78.10**	69.45
Sub + Objs + Rels	68.32	68.39	63.88	72.48	79.79	77.74	69.54
Sub + Objs + Rels_nm	68.35	71.96	63.31	**72.92**	78.33	75.55	69.76
Sub + Objs_nm + Rels	68.38	**73.38**	64.21	72.77	79.60	**78.10**	70.20
Sub + Objs_nm + Rels_nm	**68.62**	73.29	**64.68**	72.42	79.75	77.83	**70.37**

Table 6.4: The influence of question type for different methods with time stamps. “Sub”, “Objs”, “Rels” and “nm” represent subtitles, objects, relations and names, respectively.

helpful when the question includes multiple relations, as in the bottom right example, where Rachel is sitting on a couch and holding a glass at the same time, requiring models to learn this combination of relations in order to answer the question.

6.4 Analysis on the Influence of Question Type

There are six different question types in the TVQA datasets, which benchmarks the ability of a model in terms of different reasoning skills. In this section, we present separate results on the six types of questions, including “what”, “who”, “where”, “how”, “why” and “other”. According to Table 6.4, we give the ablation study on different methods with time stamps to show the influence of input feature combinations on different reasoning skills. Compared to the baseline using only the subtitle (“Sub”), our full model (“Sub + Objs_nm + Rels_nm”) achieves significant performance gain on almost all questions, where the top 3 improvements are in “where”, “what” and “who”, which is as expected since they mostly involve character-aware relations such as “Where does Sheldon sit on when...?” or “What instrument is Raj playing when...?” or “Who walks into the room when...?”. It is also worth noticing that all the best performances for each reasoning skill (marked in bold) are achieved when relations are utilized as a visual semantic, which demonstrates the efficacy of our core idea, *i.e.*, leveraging relations between characters for better video story understanding.

6.5 Summary

In this work, we propose character-aware scene understanding for improved Video Story Question Answering. Our character-aware reasoning network is trained in a end-to-end multi-task

style to acquire weakly supervised character identification as well as video story understanding. For the experiments on the six TV shows, our full Subtitle + Objects + Relations + Names model achieves the best accuracy against all baselines, which confirms the effectiveness of our multi-task framework and character-aware reasoning model.

Chapter 7

Conclusions

This dissertation has explored two topics: Scene Graph Parsing and its application on multi-modal reasoning tasks.

First we presented the relationship proposal networks, an end-to-end model that takes an image as input and output object pairs that are most likely to be related. The underlying motivation for this work is to alleviate the intrinsic complexity of relationship candidate, i.e., the complexity would become quadratic, and thus very expensive, if all pairs of objects have to be considered for relationship detection. With the proposed network, the number of relationship proposals, i.e., object pairs to be considered, can be reduced from 90,000 down to 2,000, which could lead to speed-up at an order of magnitude level. We also experimentally demonstrate that competitive baselines built upon previous methods could do the same job but show significantly worse performance, which proves the efficacy and necessity of the proposed method.

Second, we discussed the scenario where the number of object and predicate categories could be as large as 80,000, and some inherent issues due to this large scale. Our solution is a model that outputs embeddings instead of discrete relation labels, where the embeddings encode underlying association between subjects, predicates and objects. We observed that the traditional cross entropy loss and triplet loss fail to train the network efficiently, and we designed a new loss called Triplet-Softmax loss that combines these two and successfully drives the learning process. We conducted sufficient experiments on both the original dataset and the subset with only infrequent classes, demonstrating that our model is able to achieve our desired goals with significant advantage over the aforementioned two baselines.

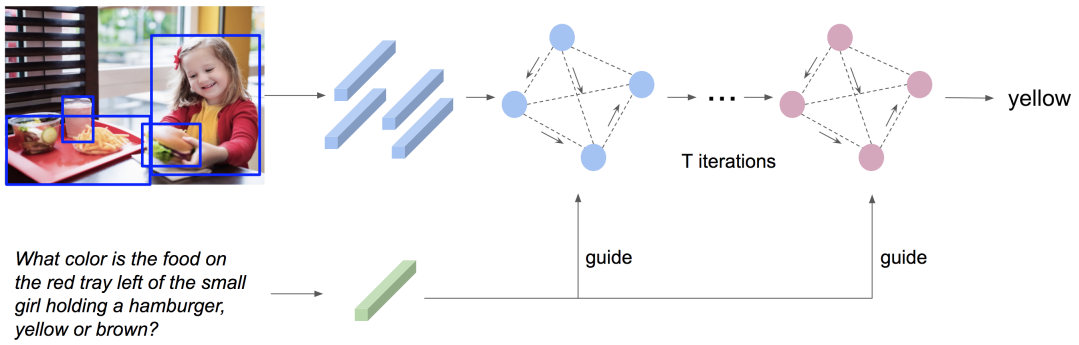
Third, we tackled two problems that exist widely in natural images: 1) when a subject (or object) is related to an object (or subject), there could be multiple objects (or subjects) placed in the close proximity with the same category that might confuse the model due to visual

similarity; 2) when multiple relationships happen within a close distance and they are related in the same way (i.e., their predicates are the same), the model might be confused on which subjects are related to which objects due to visual ambiguity. Such issues happen extensively in natural images but with various visual context, and our goal is to find a universal solution that considers all possible scenarios. Motivated by that, we designed three types of losses that deliberately force the model to contrast positive training samples against negatives, where positive samples are defined as related objects according to the ground-truth. The losses are the same in terms of the general forms but different only in terms of the constraints given to them. We extensively conduct experiments that compare models with and without our proposed losses to show that the losses efficiently improve the models' performance, especially when the image suffers the aforementioned two issues badly. We also proposed a brand new model for end-to-end relationship detection, and together with the proposed losses, the whole system achieve state-of-the-art relationship detection performance.

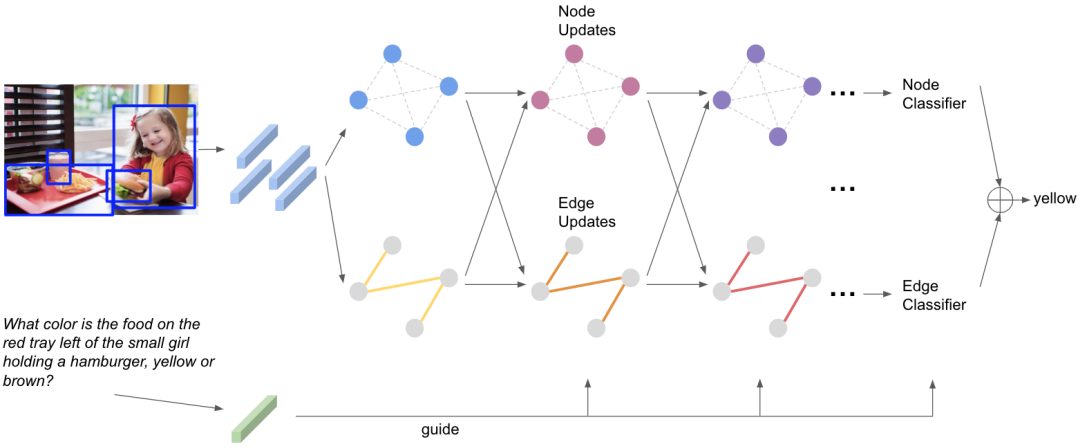
Finally, we presented a pipeline that leverages relationships for better video story understanding. We show that by using detected relationships as richer input features, the model is able to conduct visual reasoning more accurately and explicitly. The detected relationships not only helps the system understand video stories at a deeper level, but also provides knowledge base for analyzing the whole reasoning process.

Chapter 8

Future Direction



(a) Previous pipeline



(b) Potential pipeline

Figure 8.1: Illustration of previous common pipeline and the potential future model.

One potential future direction is to reason over scene graphs. Currently state-of-the-art methods usually gather all detected objects and build a fully connected graph on them and reason over this graph for question answering. The whole pipeline is illustrated in Figure 8.1.

Concretely, the mathematical formulation could be:

$$r_{i,j}^{(t)} = W_r\Big[W_{x_s}x_i^{(t-1)}; W_{x_o}x_j^{(t-1)}; W_s l_i; W_p p_{i,j}; W_o l_j\Big] \odot W_1 q^{(t)}$$

$$\beta_{i,j}^{(t)} = softmax\Big(W_e r_{i,j}^{(t)}\Big)$$

$$\tilde{x}_i^{(t)} = \sum_{j=1}^{N} \beta_{i,j}^{(t)} r_{i,j}^{(t)}$$

$$x_i^{(t)} = W_v\Big[x_i^{(t-1)}; \tilde{x}_i^{(t)}\Big]$$

where $x_i^{(t)}$ represents the feature of node (object) i at iteration t, $l_i, p_{i,j}, l_j$ are the semantic features for detected subject, predicate, object between object i and j. $\odot$ is element-wise multiplication and $q^{(t)}$ is the question representation at iteration t. $r_{i,j}^{(t)}$ is the compositional representation fused by node and relationship features. $\beta_{i,j}^{(t)}$ is the graph attention weight between object i and j. $\tilde{x}_i^{(t)}$ is the updated node feature obtained by gathering weighted representation of edges connected to node i. There are two major advantages of this model over the previously popular one (Figure 8.1a): 1) The new model utilizes scene graphs for visual reasoning, where both nodes and edges are explicitly detected and represented, therefore the whole reasoning process is expected to be more interpretable; 2) Scene graphs are usually sparse for because the number of meaningful relationships (edges) are mostly much less than the square of the number of objects, therefore all the aforementioned computations can be implemented in sparse vectors, which prevents memory overflow and enables more complex edge representations. It is also beneficial to visualize and analyze the information being propagated through the edges in order to see how the reasoning leads to the correct answer, which I believe is an important way to understand how AI could better comprehend visual and textual signals.

References

[1] OpenImages VRD Challenge. https://storage.googleapis.com/openimages/web/challenge.html.

[2] 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.

[3] 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 *CVPR*, 2018.

[4] 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.

[5] M. Bauml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *CVPR*, 2013.

[6] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620, 2017.

[8] R. Cadene, H. Ben-Younes, M. Cord, and N. Thome. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2019.

[9] 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.

[10] 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.

[11] K. Chen, R. Kovvuri, and R. Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 2017.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[13] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017.

[14] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pages 48–64. Springer, 2014.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

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

[17] M. Elhoseiny, S. Cohen, W. Chang, B. Price, and A. Elgammal. Sherlock: Scalable fact learning in images. In *AAAI*, 2017.

[18] 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.

[19] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[20] H. Fan and J. Zhou. Stacked latent attention for multimodal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2018.

[21] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[22] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[23] 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.

[24] J. Gao, R. Ge, K. Chen, and R. Nevatia. Motion-appearance co-memory networks for video question answering. In *CVPR*, 2018.

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

[26] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object intaractions. *CVPR*, 2018.

[27] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[28] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233, 2014.

[29] A. Gordo and D. Larlus. Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] 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.

[31] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[32] T. Gupta, K. J. Shih, S. Singh, and D. Hoiem. Aligned image-word representations improve inductive transfer across vision-language tasks. In *ICCV*, 2017.

[33] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[34] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017.

[35] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016.

[36] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[38] S. Jin, H. Su, C. Stauffer, and E. Learned-Miller. End-to-end face detection and cast grouping in movies using erdos-renyi clustering. In *ICCV*, 2017.

[39] 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.

[40] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo. Gaining extra supervision via multi-task learning for multi-modal video question answering. *IJCNN*, 2019.

[41] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo. Progressive attention memory network for movie story question answering. In *CVPR*, 2019.

[42] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018.

[43] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016.

[44] K.-M. Kim, S.-H. Choi, J.-H. Kim, and B.-T. Zhang. Multimodal dual attention memory for video story question answering. *ECCV*, 2018.

[45] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[46] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *International Conference on Machine Learning*, pages 595–603, 2014.

[47] 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.

[48] 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. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[49] 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.

[50] J. Lei, L. Yu, M. Bansal, and T. L. Berg. TVQA: localized, compositional video question answering. In *EMNLP*, 2018.

[51] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.

[52] G. Li, H. Su, and W. Zhu. Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks. *arXiv preprint arXiv:1712.00733*, 2017.

[53] Q. Li, J. Fu, D. Yu, T. Mei, and J. Luo. Tell-and-answer: Towards explainable visual question answering using attributes and captions. *arXiv preprint arXiv:1801.09041*, 2018.

[54] X. Li, J. Song, L. Gao, X. Liu, W. Huang, X. He, and C. Gan. Beyond RNNs: Positional self-attention with co-attention for video question answering. *AAAI*, 2019.

[55] Y. Li, W. Ouyang, X. Wang, and X. Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 7244–7253. IEEE, 2017.

[56] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *arXiv preprint arXiv:1703.03054*, 2017.

[57] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu, and T. Mei. Rethinking visual relationships for high-level image understanding. *arXiv preprint arXiv:1902.00313*, 2019.

[58] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[59] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.

[60] J. Liu, L. Wang, M.-H. Yang, et al. Referring expression generation and comprehension via attributes. In *CVPR*, 2017.

[61] 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.

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

[63] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[64] P. Lu, L. Ji, W. Zhang, N. Duan, M. Zhou, and J. Wang. R-vqa: learning visual relation facts with semantic attention for visual question answering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1880–1889. ACM, 2018.

[65] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, 2017.

[66] C. Ma, C. Shen, A. Dick, Q. Wu, P. Wang, A. van den Hengel, and I. Reid. Visual question answering with memory-augmented networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6975–6984, 2018.

[67] 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.

[68] M. Malinowski, C. Doersch, A. Santoro, and P. Battaglia. Learning visual question answering by bootstrapping hard attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20, 2018.

[69] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[70] 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.

[71] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[72] A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, 2013.

[73] S. Na, S. Lee, J. Kim, and G. Kim. A read-write memory network for movie story understanding. In *ICCV*, 2017.

[74] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016.

[75] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307, 2017.

[76] A. Newell and J. Deng. Pixels to graphs by associative embedding. In *NIPS*, 2017.

[77] W. Norcliffe-Brown, S. Vafeias, and S. Parisot. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*, pages 8334–8343, 2018.

[78] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations*, 2014.

[79] O. Parkhi, E. Rahtu, Q. Cao, and A. Zisserman. Automated video face labelling for films and tv material. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[80] B. Patro and V. P. Namboodiri. Differential attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7680–7688, 2018.

[81] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *EMNLP*, 2014.

[82] J. Peyre, I. Laptev, C. Schmid, and J. Sivic. Weakly-supervised learning of visual relations. In *ICCV*, 2017.

[83] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[84] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1946–1955. IEEE, 2017.

[85] 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.

[86] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[87] 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.

[88] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016.

[89] 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.

[90] 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.

[91] 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.

[92] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[93] 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.

[94] 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.

[95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[97] Z. Su, C. Zhu, Y. Dong, D. Cai, Y. Chen, and J. Li. Learning visual knowledge memory networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7736–7745, 2018.

[98] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Improved weak labels using contextual cues for person identification in videos. In *FG*, 2015.

[99] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4223–4232, 2018.

[100] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[101] 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.

[102] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[103] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[104] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. ReNet: A recurrent neural network based alternative to convolutional networks. *ArXiv e-prints*, abs/1505.00393, May 2015.

[105] B. Wang, Y. Xu, Y. Han, and R. Hong. Movie question answering: Remembering the textual cues for layered visual contents. *AAAI*, 2018.

[106] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[107] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016.

[108] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019.

[109] P. Wang, Q. Wu, C. Shen, and A. van den Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1182, 2017.

[110] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381, 2017.

[111] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4622–4630, 2016.

[112] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 2018.

[113] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[114] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[115] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018.

[116] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. *arXiv preprint arXiv:1808.00191*, 2018.

[117] X. Yang, H. Zhang, and J. Cai. Shuffle-then-assemble: Learning object-agnostic visual relationship features. In *ECCV*, 2018.

[118] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[119] Z. Yang, J. Yu, C. Yang, Z. Qin, and Y. Hu. Multi-modal learning with prior visual relation reasoning. *arXiv preprint arXiv:1812.09681*, 2018.

[120] 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.

[121] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699, 2018.

[122] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018.

[123] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *ICLR*, 2018.

[124] D. Yu, J. Fu, T. Mei, and Y. Rui. Multi-level attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4709–4717, 2017.

[125] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[126] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018.

[127] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[128] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[129] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems*, 29(12):5947–5959, 2018.

[130] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[131] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[132] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018.

[133] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[134] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3107–3115. IEEE, 2017.

[135] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2017.

[136] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, 2017.

[137] J. Zhang, Y. Kalantidis, M. Rohrbach, M. Paluri, A. Elgammal, and M. Elhoseiny. Large-scale visual relationship understanding. In *AAAI*, 2019.

[138] J. Zhang, K. Shih, A. Tao, B. Catanzaro, and A. Elgammal. An interpretable model for scene graph generation. *arXiv preprint arXiv:1811.09543*, 2018.

[139] J. Zhang, K. Shih, A. Tao, B. Catanzaro, and A. Elgammal. An interpretable model for scene graph generation. *arXiv preprint arXiv:1811.09543*, 2018.

[140] J. Zhang, K. Shih, A. Tao, B. Catanzaro, and A. Elgammal. Introduction to the 1st place winning model of openimages relationship detection challenge. *arXiv preprint arXiv:1811.00662*, 2018.

[141] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical contrastive losses for scene graph parsing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[142] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer, 2016.

[143] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S3FD: Single shot scale-invariant face detector. In *ICCV*, 2017.

[144] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. Open vocabulary scene parsing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017.

[145] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1291–1300, 2017.

[146] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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