VISUAL RELATIONSHIP UNDERSTANDING

BY

ZIH-SIOU HUNG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Associate Professor Svetlana Lazebnik

ABSTRACT

This thesis addresses two visual understanding tasks: visual relationship detection (VRD) and video action recognition. The majority of the thesis is focused on VRD, which is our main contribution.

Relations amongst entities play a central role in image and video understanding. In the first three chapters, we discuss visual relationship detection, whose goal is to recognize all (*subject*, *predicate*, *object*) tuples in a given image. Due to the complexity of modeling (*subject*, *predicate*, *object*) relation triplets, it is crucial to develop a method that can not only recognize seen relations, but also generalize to unseen cases. Inspired by a previously proposed visual translation embedding model, or VTransE [1], we propose a context-augmented translation embedding model that can capture both common and rare relations. The previous VTransE model maps entities and predicates into a low-dimensional embedding vector space where the predicate is interpreted as a translation vector between the embedded features of the bounding box regions of the *subject* and the *object*. Our model additionally incorporates the contextual information captured by the bounding box of the *union* of the subject and the object, and learns the embeddings guided by the constraint *predicate* $\approx$ *union* (*subject*, *object*) $-$ *subject* $-$ *object*. In a comprehensive evaluation on multiple challenging benchmarks, our approach outperforms previous translation-based models and comes close to or exceeds the state of the art across a range of settings, from small-scale to large-scale datasets, from common to previously unseen relations. It also achieves promising results for the recently introduced task of scene graph generation.

In the final part of the thesis, we consider action understanding in videos. In many scenarios, we observe moving objects instead of still images. Thus, it is also important to capture motion information and recognize the action being performed. Recent work either applies 3D convolution operators to extract the motion implicitly or adds an additional optical flow path to leverage temporal features. In our work, we propose to use a novel correlation operator to establish a matching between consecutive frames. This matching encodes the movement of objects through time. Combined with the classical appearance stream, the proposed method hence learns the appearance and motion representations in parallel. On the challenging Something-Something dataset [2], we empirically demonstrate that our network achieves comparable performance to the state-of-the-art method.

To my family, for their love and support.

ACKNOWLEDGMENTS

First, I must thank Professor Svetlana Lazebnik for her advice and guidance throughout my master studies. I learned a lot from her both in terms of research style and technical knowledge. Without her support, this work would have been impossible.

Another person who has a large impact on my work is Dr. Arun Mallya. His advice was invaluable to me on many occasions. He taught me the way to start a research project, design experiments, and finally get the results published. I feel privileged to have had the opportunities to work with him. I am also very grateful to Professor Alexander Schwing for helpful discussion about the video action recognition work. I learned enormously from many enlightening discussion with him.

Finally, I would like to express my gratitude to my family for their unconditional trust and unfailing emotional support. It is their love that raise me up again when I get weary.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Performance on object detection and localization has improved greatly over the last few years with the introduction of the deep R-CNN model [3] and its successors [4, 5, 6, 7]. The next natural step is to go beyond detecting individual objects and start reasoning about semantic relationships between multiple objects, which could be useful for applications such as image captioning [8], retrieval [9, 10], and visual question answering [11]. In this thesis, we discuss two relationship understanding tasks: visual relationship detection (VRD) [12] and video action recognition. The former one concentrates on image data, while the latter puts more emphasis on temporal modeling. Since VRD is our main contribution, the majority of the thesis focuses on VRD. We only talk briefly about action recognition.

1.1 THE VRD TASK

VRD focuses on understanding interactions between pairs of object entities in the image. These interactions can be spatial, comparative, or action-based, and are represented as (*subject*, *predicate*, *object*) triplets such as (*desk*, *beneath*, *laptop*), (*tower*, *taller than*, *trees*), or (*person*, *eat*, *pizza*). VRD has two goals: detection of object instances participating in an interaction, and correct prediction of the interaction type. Inferring the relations between object pairs is not always straightforward visually, and depends on context. For instance, (*person*, *hold*, *umbrella*) and (*person*, *hold*, *guitar*) are dissimilar in an image even though they share the same predicate '*hold*'. The very large output space makes this task even more challenging. Consider the Stanford VRD dataset [12], which has 100 classes of objects, 70 classes of predicates, and a total of 30k training relationship annotations. The number of possible interaction triplets, including unusual cases such as (*dog*, *ride*, *horse*), is $100 \times 100 \times 70 = 700k$, meaning that most relationships do not even have a training example. This sparsity necessitates the development of methods that can recognize the predicate even if it occurs with a novel subject or object.

To improve generalization to rare or unseen relationships, we propose a novel framework called Union Visual Translation Embedding, or **UVTransE**. Our starting point is the recently introduced **VTransE** method of Zhang et al. [1], which maps entities and predicates into a low-dimensional embedding vector space where the *predicate* is interpreted as a translation vector between the embedded appearance features of the *subject* and the *object*. More concretely, if $\boldsymbol{s}$, $\boldsymbol{p}$, and $\boldsymbol{o}$ are vectors representing the subject, the predicate, and the object in the learned embedding space, VTransE assumes that a relationship ($\boldsymbol{s}$, $\boldsymbol{p}$, $\boldsymbol{o}$) exists if $\boldsymbol{s} + \boldsymbol{p} \approx \boldsymbol{o}$. This formulation was inspired, in turn, by translation embeddings for relational data [13].

VTransE does a good job of predicting relationships that it has seen in training time; however, it is not well-suited to recognizing unseen relationship triplets. This is due to two critical issues. First, VTransE calculates object vectors based on the features from subject and predicate only. That is, subject and predicate vectors ($\boldsymbol{s}$, $\boldsymbol{p}$) in the learned embedding space completely determine the object $\boldsymbol{o}$ as $\boldsymbol{s} + \boldsymbol{p}$. Consider an unusual relationship, such as (*dog*, *drive*, *car*). Since the triplet is rare in the training set, the *dog* and *drive* vectors are not trained to produce this particular object *car*, so we end up with $\boldsymbol{s} + \boldsymbol{p} \not\approx \boldsymbol{o}$. In addition, the VTransE embedding is not sufficiently flexible for modeling cases where many possible objects can satisfy a predicate with a given subject, since fixing $\boldsymbol{s}$ and $\boldsymbol{p}$ roughly determines $\boldsymbol{o}$.

In order to overcome the above two problems, we propose an extension to VTransE that not only enables triplets to be recognized in unseen cases, but also enables entities to have a distributed representation in the embedding space. Like VTransE, we model objects and predicates as embedding vectors; however, our predicate embedding vectors are not constrained to represent the translation between the subject and the object. Our idea is that by subtracting the embeddings of the *subject* and the *object* from the embedding of the entire box of the union of *subject* and *object* should provide an embedding corresponding to the predicate of interest, or $\boldsymbol{u} - \boldsymbol{s} - \boldsymbol{o} \approx \boldsymbol{p}$. For example, emb(*person* $\cup$ *horse*) $-$ emb(*person*) $-$ emb(*horse*) $\approx$ emb(*ride*). (Note that here and in the following, whenever we talk about the *union box* or *union feature*, we mean the bounding box of the union of the subject and object, and the features extracted from this box.) By removing object-related information from the contextual union feature, we hope to leave behind an embedding that contains information only about the predicate, leading to better zero-shot performance. Even though our modification of the VTransE formulation may seem straightforward, our experiments will demonstrate that UVTransE model can much better handle the challenges of VRD. For example, as shown in the results in Figure 3.1, the learned predicate embedding '*touch*' can model both (*person*, *touch*, *skateboard*) and (*person*, *touch*, *glasses*), even when (*person*, *touch*, *glasses*) has not been seen during training.

Similar to prior works like [12, 14], we also incorporate a recurrent language model that uses word embeddings to learn about the semantic relatedness between different objects or different relations in an attempt to counteract the data sparsity problem. It has been shown that words with similar meaning are close to each other in word embedding spaces such as word2vec [15] and GloVe [16]. Such semantic similarity might help us in detecting relation triplets not seen during training. For instance, given that we have seen *(person, ride, motorbike)* during training time, at test time, if we have an image containing the relation *(person, ride, bicycle)*, we might be able to detect this relationship since motorbike and bicycle are semantically similar. Accordingly, we design a language module that benefits the overall detection task, including zero-shot cases. An overview of our UVTransE model, and its relationship with the language model, are shown in Fig. 1.1.

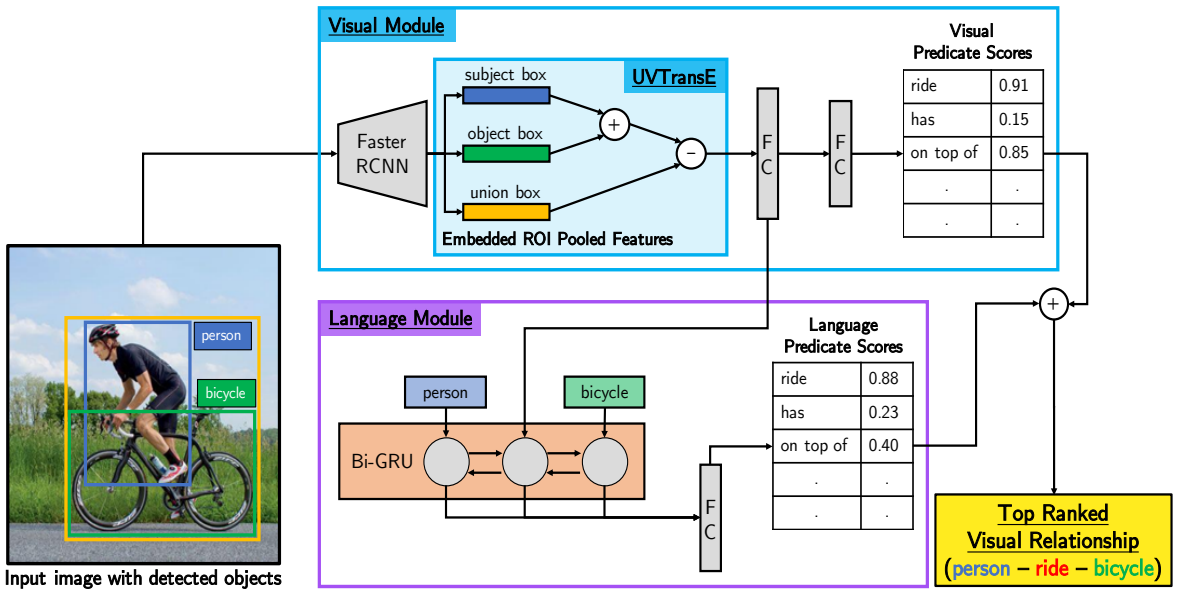

Figure 1.1: Overview of our UVTransE visual relationship detection model. Given an image, Faster R-CNN is fist used to detect objects. For each pair of detected objects, appearance and spatial features are extracted and fed into the visual module, which computes the UVTransE embedding: $union - (subject + object)$. The predicate embedding output by UVTransE may be optionally sent to a Bi-GRU language model. Finally, triplets (s, p, o) are ranked based on scores from the visual, language, and object detection modules.

1.2 THE VIDEO ACTION RECOGNITION TASK

Next, we introduce the task of video action recognition. It involves the identification of different human actions from video clips. Due to the success of Convolution neural network (CNN) on ImageNet Classification [17], CNN has been adapted to capture not only the appearance feature, but also motion information for video analysis. There are two common strategies for action classification. The first way is extending 2D convolution to 3D [18] and capture the temporal information implicitly. The alternative approach is using a two-stream network [19], which consists of an appearance stream and a motion stream. The appearance stream is similar to the normal convolution neural networks which extract RGB features from still image frames, while the motion stream applies convolution on top of the pre-calculated optical flow input for explicit motion modeling.

In our work, we would like to explicitly model the motion flow while removing the costly optical flow computation. Thus, we designed a novel architecture that uses correlation operator, which is proposed by FlowNet [20], to instantiate correspondences between consecutive frames. These correspondences encodes the motion through time explicitly. Different from traditional two-stream networks, everything in our module is differentiable, and we are able to learn the motion features

through backpropagation. Compared to 3D CNNs, our model has the benefit of factorizing out the computation of appearance and motion. This enables us to explicitly model the temporal information, and learn distinct weights for different representations of a video.

1.3 THESIS OUTLINE

The remainder of this thesis is structured as follows. In Chapter 2, we review algorithms for both visual relationship detection and video action recognition. We also point out the connections of our work and previous methods.

In Chapter 3, we start by giving the technical details of UVTransE, and present an extensive empirical evaluation of our UVTransE method on multiple datasets and settings, from small-scale to large-scale, and from common relationships to zero-shot recognition. In particular, we show that we decisively outperform VTransE and most other competing methods on both the general and zero-shot settings of the VRD dataset [12], UnRel dataset [21], two subsets of Visual Genome [22], and the Open Images Challenge [23]. On the latter two datasets, we also apply our methods to the recently proposed task of *scene graph generation*. A scene graph, introduced by Johnson *et al.* [9], encapsulates all the relations amongst the object entities in an image. Its nodes correspond to objects and directed edges correspond to their pairwise relationships. We generate scene graphs via a simple two-stage approach, where we first detect objects, or nodes, and then infer relationships, or edges, using our UVTransE approach. Our experiments will show that this approach is competitive with more sophisticated state-of-the-art approaches designed to jointly reason about multiple edges of the graph, such as Neural Motifs [24].

Finally, in Chapter 4, we introduce our video correlation network, which utilizes correlation operation for motion extraction, and apply it on the challenging Something-Something dataset [2]. Through our experiments on Something-Something dataset [2], we will demonstrate that our network could achieves competitive results over 3D CNNs and two-stream networks.

CHAPTER 2: RELATED WORK

2.1 VISUAL RELATIONSHIP DETECTION

Detecting visual relationships in one form or another has been an active area of recognition research for at least the last decade. Most earlier works focused on predicting specialized types of predicates such as spatial relations [25, 26], or targeted human-centric relationships [27, 28, 29, 30, 31, 32]. Such phrase or relationship detections were used in applications such as object recognition [33, 34, 35], image classification [36], and text grounding [37, 38].

Recently, Lu *et al.* [12] introduced the generic visual relationship detection (VRD) task and a dataset that became one of the main benchmarks. They also proposed a VRD method that established the basic template for many follow-up works, including ours: first objects are detected, then object pairs are fed to a classifier that combines their appearance features with a language prior on the relationship triplet occurrence. Zhang *et al.* [1] projected features from the detected objects into a low-dimensional space and predicted the relationship using a learned relation translation vector. This VTransE method is the main departure point for our own work. Dai *et al.* [39] proposed a deep relational network method exploiting the statistical dependencies between objects and their relationships, while Liang *et al.* [40] proposed building a semantic action graph capturing possible relations and learning to traverse it using a reinforcement learning formulation. In Zhang *et al.* [41], the authors employed a novel triplet-softmax loss to learn the joint visual and semantic embedding. Very recently, Zhang *et al.* [42] defined margin-based losses to address common types of errors existing in relationship prediction, resulting in a method that performs remarkably well on the detection of common relationship triplets, but does not necessarily generalize to rare or zero-shot relationships. As part of its visual representation, this method also uses the bounding box of the union of subject and object, however, it does not use a subtractive model for combining the union with the subject and object boxes, as we propose. Zhuang *et al.* [43] designed a context-aware interaction classifier with good generalization to the zero-shot case. Plummer *et al.* [14] obtained strong zero-shot performance through the use of multiple visual-language cues learned with Canonical Correlation Analysis (CCA). Yu *et al.* [44] used a large amount of external textual data to distill useful knowledge for triplet learning. Peyre *et al.* [21] focused on weakly supervised learning of relationships (not a setting we consider), and also introduced the UnRel dataset exhaustively annotated for a set of unusual triplets such as *(elephant, wear, glasses)*. This is one of the benchmarks used in our work.

As stated in the Introduction, we also apply our UVTransE method to scene graph generation. Most scene graph generation methods consider the surrounding context of a node as a valuable

cue, and apply context propagation mechanisms to exchange information between neighboring nodes over a candidate scene graph. In Xu *et al.* [45], two sub-graphs, representing objects and relationships respectively, are created. Node features, which are used to predict relation types, are updated based on the messages passed between the two graphs. Similarly, Li *et al.* [46] proposed constructing a dynamic graph, where messages are passed across different feature representations to refine the scene graph. Zellers *et al.* [24] designed a Stacked Motif Network to extract contextual cues, which are propagated across objects and relations. Yang *et al.* [47] developed an attentional graph convolutional network to place attention on reliable edges when information is exchanged between vertices in the candidate scene graph. In Chapters 3.2.3 and 3.2.4, we apply our method to generate scene graphs on Visual Genome and Open Images datasets in a very straightforward way: we first run object detectors to find the nodes of the scene graph, and then use UVTransE to find the relations. Even though we are predicting each relationship independently, we will show that our results are competitive with those of more context-aware methods.

2.2 VIDEO ACTION RECOGNITION

As stated in the Introduction, there are two major approaches for video action recognition. Since the introduction of two-stream network [19], several improvements have been made to achieve better performance. Feichtenhofer *et al.* [48] demonstrated that fusing at the convolution stages with a novel spatial temporal pooling can boost accuracy. Feichtenhofer *et al.* [48] also showed that multiplicative interactions of spacetime features are benificial for action recognition. On the other hand, 3D CNNs learn spatial and temporal features at the same time by learning 3D filters in space and time. Carreira *et al.* [18] extended the successful 2D Inception v1 [49] architecture to 3D, and got a significant performance boost on Kinetics dataset [50]. Tran *et al.* [51] and Xie *et al.* [52] both factorized 3D convolution into 2D spatial convolution and 1D temporal convolution. They showed that this factorization reduced overfitting and led to a better accuracy. Wang *et al.* [53] incorporated non local block, which is a generalization of self attention layer, into 3D CNNs to capture long range temporal and spatial dependency. In our work, we combine the strength of both worlds, and propose the correlation operator to learn the dynamics of videos.

CHAPTER 3: VRD APPROACH

3.1 THE UVTRANSE METHOD

In this chapter, we discribe our UVTransE method in detail. We split the VRD task into two stages. In the first stage, we use an off-the-shelf object detection model, such as Faster R-CNN [6], to predict object bounding boxes and per-class confidences in an image. For the second stage, we learn a model to score all possible triplets (s, p, o) where s is a *subject* box, p is a *predicate* or relation label, and o is an *object* box. Next, we describe our UVTransE relationship scoring model, which is illustrated in Figure 1.1.

3.1.1 Union Visual Translation Embedding

Let $\boldsymbol{s}$, $\boldsymbol{o}$, $\boldsymbol{u} \in \mathbb{R}^n$ be the appearance features of the bounding boxes enclosing the *subject*, *object*, and *union* of subject and object, respectively. We want to learn three projection functions $\boldsymbol{f}_s : \mathbb{R}^n \to \mathbb{R}^d$, $\boldsymbol{f}_o : \mathbb{R}^n \to \mathbb{R}^d$ and $\boldsymbol{f}_u : \mathbb{R}^n \to \mathbb{R}^d$ that map the respective feature vectors into a common d-dimensional embedding space, as well as translation vectors $\boldsymbol{p}$ in the same space corresponding to each of the predicate labels present in the data. In our implementation, the functions $\boldsymbol{f}_s$, $\boldsymbol{f}_o$, and $\boldsymbol{f}_u$ are multilayer perceptrons. A relationship (s, p, o) that exists in the training data should impose the constraint $\boldsymbol{f}_u(\boldsymbol{u}) - \boldsymbol{f}_s(\boldsymbol{s}) - \boldsymbol{f}_o(\boldsymbol{o}) \approx \boldsymbol{p}$. To achieve this, similarly to [1], we learn $\boldsymbol{f}_s$, $\boldsymbol{f}_o$, $\boldsymbol{f}_u$, and $\boldsymbol{p}$ by minimizing the following multi-class cross-entropy loss function:

$$L_{\text{vis}} = \sum_{(s,p,o)\in T} -\log \frac{\exp\left(\boldsymbol{p}^\top \hat{\boldsymbol{p}}\right)}{\sum_{q\in P} \exp\left(\boldsymbol{q}^\top \hat{\boldsymbol{p}}\right)}, \tag{3.1}$$

where

$$\hat{\boldsymbol{p}} = \boldsymbol{f}_u(\boldsymbol{u}) - \boldsymbol{f}_s(\boldsymbol{s}) - \boldsymbol{f}_o(\boldsymbol{o})\,, \tag{3.2}$$

T is the set of all relationship triplets existing in the training data,[1] and P is the set of all predicate labels. In practice, we found that we need to constrain the norms of $\boldsymbol{f}_u(\boldsymbol{u})$, $\boldsymbol{f}_s(\boldsymbol{s})$, and $\boldsymbol{f}_o(\boldsymbol{o})$ from getting arbitrary large. To this end, we augment Eq. (3.1) with soft constraints on embedding

[1] If there are multiple training examples with the same (s, p, o), they yield multiple terms in the summations of Eqs. (1) and (3).

weights:

$$L_{\text{vis}} = \sum_{(s,p,o)\in T} -\log \frac{\exp\left(\boldsymbol{p}^\top \hat{\boldsymbol{p}}\right)}{\sum_{q\in P}\exp\left(\boldsymbol{q}^\top \hat{\boldsymbol{p}}\right)} + C\big([\|\boldsymbol{f}_s(\boldsymbol{s})\|_2^2 - 1]_+ + [\|\boldsymbol{f}_o(\boldsymbol{o})\|_2^2 - 1]_+ + [\|\boldsymbol{f}_u(\boldsymbol{u})\|_2^2 - 1]_+\big), \tag{3.3}$$

where $[x]_+ = \max(0, x)$. We experimented with other penalties to encourage the norms to stay close to one but found this one gave the best results. C is a hyperparameter that determines the relative importance of the soft constraints, and its effect will be examined in Chapter 3.2.

Our formulation of Eq. (3.3) differs from VTransE [1] in the addition of the contextual union feature and the norm regularization terms. Ablation studies of Chapter 3.2.1 will show that these modifications are key to improving performance, not only on common cases but also on the zero-shot case.

At test time, given a candidate triplet (s, p, o), we can score the predicate p as

$$z_p = \frac{\exp\left(\boldsymbol{p}^\top \hat{\boldsymbol{p}}\right)}{\sum_{q\in P}\exp\left(\boldsymbol{q}^\top \hat{\boldsymbol{p}}\right)}\,. \tag{3.4}$$

Similarly to [1], we can then define the score of the entire triplet by the sum of softmax detection scores for the subject and object, (z_s, z_o), and the above predicate score z_p:

$$z_{(s,p,o)} = z_s + z_p + z_o\,. \tag{3.5}$$

Alternatively, for some datasets, we obtained better performance by taking the product of the above scores. Dataset-specific details will be given in Chapter 3.2.

3.1.2 Language Module

Similar to prior work [40, 12, 14, 44], we combine UVTransE with a language model that helps to combat data sparsity and learns which relationships are plausible between pairs of object classes. Our language module is a bi-directional GRU (Bi-GRU) [54] that receives encodings of subject, predicate, and object in three successive steps, concatenates the hidden states, and uses them for predicate classification. Further details will be given in Chapter 3.1.3. The loss for our language module L_{lang} is a standard multi-class cross-entropy loss which encourages it to produce the ground

truth predicate. The combined loss for our model is given by

$$L_{\text{total}} = \alpha L_{\text{vis}} + (1 - \alpha) L_{\text{lang}} \,. \tag{3.6}$$

The score $z_{(s,p,o)}$ for a candidate relationship is now given by

$$z_{(s,p,o)} = z_s + z_o + \alpha z_p + (1 - \alpha) z_l \,, \tag{3.7}$$

where α is the weight for the visual module and z_l is the softmax predicate score from the language module. The values of α used in the experiments will be given in Chapter 3.2.

3.1.3 Implementation Details

In detail, the stages of our pipeline are: object detection, extraction of appearance and location features from bounding boxes, UVTransE relation embedding, language module (optional), and relationship scoring. The implementation of each of these components is described below.

Object Detection. Our first step is to run an object detector to locate a set of candidate objects in an image. We train a separate Faster R-CNN detector [6] for each dataset. Our experiments use two backbones: VGG-16 [55] and ResNet-101 [56] (see Chapter 3.2 for dataset-specific details). Each candidate object output by the detector is associated with a bounding box b_i, object class probability z_i, and an ROI-pooled feature vector f_i.

Appearance feature extraction. Our appearance features are based on the ROI-pooled features f_i obtained from the object detector. These are 4096-d for the VGG backbone and 2048-d for the ResNet backbone. More specifically, we use the ROIAlign features of [5], although in our experience, the improvement they give over standard ROIPool features is slight (less than a percentage point in mAP and relationship detection measures). We follow the specification in Chapter 3.1.1, and pass the features f_i through two FC layers with ReLU activation. The output dimensionalities of FC layers are 512 and 256, and we get 256-d appearance features at the end.

Location feature extraction. We encode each single bounding box (subject or object) into a 5-d vector $l_i = (\frac{x_i}{W_I}, \frac{y_i}{H_I}, \frac{x_i+w_i}{W_I}, \frac{y_i+h_i}{H_I}, \frac{A_i}{A_I})$, where $(x_i,\, y_i)$ are the center coordinates, $(w_i,\, h_i)$ are the width and height, A_i and A_I are the areas of region i and image I, and W_I and H_I are the width and height of the image I. To represent union boxes, we compute the following 9-d feature:

$$\begin{aligned} l_{s \cup o} = \Bigg(& \frac{x_s - x_o}{w_o}, \frac{y_s - y_o}{h_o}, \log \frac{w_s}{w_o}, \log \frac{h_s}{h_o}, \\ & \frac{x_o - x_s}{w_s}, \frac{y_o - y_s}{h_s}, \log \frac{w_o}{w_s}, \log \frac{h_o}{h_s}, \frac{A_u}{A_I} \Bigg), \end{aligned} \tag{3.8}$$

where (x_s, y_s, w_s, h_s) and (x_o, y_o, w_o, h_o) are the subject and object box coordinates and A_u is the area of the union box. In our network, all location features $(l_s, l_o, l_{s\cup o})$ are first concatenated into a 19-d vector, which is then fed into a two-layer MLP with intermediate layer dimenension of 32 and output dimension of 16.

UVTransE Module. In this stage, each pair of objects, together with their union features, are sent to UVTransE, which is discussed in detail in Chapter 3.1. After performing UVTransE, the outputs are passed through two FC layers of input-output sizes of 256 (appearance) + 16 (location) $\rightarrow$ 256, and 256 $\rightarrow |P|$ to produce a confidence score per predicate. These scores can be used as-is to output a set of ranked relationships, or can be combined with the scores of the language module.

Language Module. As stated in Chapter 3.1, our language module is based on bi-GRUs [54]. We use GloVe [16] for our word embedding to encode subject and object class names. Then we get the predicate embedding $\hat{\boldsymbol{p}}$ from UVTransE (Eq. 3.2) and put it through a fully connected (FC) layer to get the same dimensionality as GloVe. Next, we feed the subject, predicate, and object encodings into three successive steps of a bi-directional GRU (Bi-GRU) [54]. The hidden states, which are 100-d, are then concatenated across the three time steps and both directions are used for predicate classification with two FC layers of size $600 \rightarrow 256 \rightarrow |P|$.

3.2 EXPERIMENTS

In Chapter 3.2.1, we begin by evaluating our method on the VRD dataset [12], which is moderate in size and is one of the most common benchmarks for relationship detection. Because we are especially interested in the setting of rare and unusual relations, Chapter 3.2.2 presents an evaluation on the UnRel dataset [21], which is small and can only be used for testing. Finally, to demonstrate that our method also works well on larger-scale benchmarks, as well as on the recently introduced task of scene graph generation, Chapters 3.2.3 and 3.2.4 report results on two subsets of the Visual Genome [22] and Google's Open Images [23].

3.2.1 **Results on the Stanford VRD Dataset**

Dataset. We follow the methodology of [12] to evaluate our method on the Stanford VRD dataset [12]. This dataset contains 5,000 images with 100 object categories and 70 predicates. It has around 30k relation annotations, with an average of 8 relations per image. We use the same train/test split as in [12], consisting of 4,000 training images and 1,000 test images. In this specific split, 1,877 relationships in the test set never occur in the training set, thus allowing us to evaluate zero-shot prediction.

Dataset-specific details. We use Faster R-CNN with the VGG-16 backbone to obtain candidate objects. The VGG-16 network is initialized with parameters pre-trained on ImageNet and fine-tuned on the VRD dataset and a subset of Visual Genome. Specifically, because some objects have less than 50 instances in the VRD training set, we take at least 500 instances for each class from Visual Genome. Our object detector has an mAP of 19.1 on the VRD dataset. This is low in absolute terms, partly due to incomplete ground truth annotations, but higher than the 13.98 mAP reported by Zhang et al. [1]. To obtain subject and object boxes for training UVTransE, we use ground truth boxes as well as detected boxes with $IoU \geqslant 0.5$. At test time, for each image, we use the top 30 candidate object boxes returned by Faster R-CNN for mining relationships.

We freeze the weights of the detector while jointly training UVTransE and language modules. The hyper-parameters used for the VRD dataset are $C = 1.0$ (regularization constant, Eq. 3.3) and $\alpha = 0.5$ (visual-language weighting, Eq. 3.7). SGD is used as the optimizer with an initial learning rate of $1e^{-3}$ for the detector, UVTransE, and the language module.

Evaluation metrics. Our evaluation methodology is consistent with [12]. Given a test image, the VRD model being evaluated is used to score all possible predicates between every pair of detected objects, retaining only the top k best-scoring predicates for each pair. Then we rank all these predictions and report **Recall@50** and **Recall@100**, or the fraction of ground-truth triplets that are correctly recalled in the top 50 or 100. The evaluation is done for three setups.

1. **Predicate detection:** To investigate whether the VRD model is good at detecting relations, independent of the quality of object detection, we measure the accuracy of predicate prediction when the ground truth object classes and boxes are given. A few previous works [39, 57] evaluate their predicate detection under the $k = 70$ setting, where k is the number of chosen predicates for each object pair, to achieve better recall. However, we stick to the original setting [12] and evaluate it for $k = 1$.

2. **Phrase detection:** In this setting, a prediction is considered correct if a triplet (s, r, o) is correctly recognized, and the area of intersection over union (IoU) between the predicted $s \cup o$ box and the ground-truth is above 0.5.

3. **Relationship detection:** This is similar to phrase detection, except that it requires the IoU for subject and object box to both be above 0.5, which is more challenging.

Ablation Study. First, we perform ablation studies to evaluate the effectiveness of different components of our model. Table 3.1 shows the performance of our model for different values of the regularization parameter C on the embedding weights (Eq. 3.3). The low performance for $C = 0$ confirms that regularizing the norms of projected subject, object, and predicate vectors is important

	All Test		Zero-shot Only	
	Phr. Det.	Rel. Det.	Phr. Det.	Rel. Det.
$C = 0$	6.48	4.67	4.28	3.08
$C = 0.5$	22.14	18.96	11.21	9.75
$C = 1.0$	**23.92**	**20.22**	**11.77**	**10.21**
$C = 1.5$	23.38	19.99	11.46	9.75

Table 3.1: The effect of C on Recall@50 on the Stanford VRD dataset. **Bold** indicates highest numbers.

for learning effective embeddings in our framework. The value of $C = 1$ gives us the best results so we use it in all subsequent experiments.

In Table 3.2, we compare our method to several baselines using the same trained detector, and thus, the same predicted bounding boxes, detector confidence scores, and visual features to describe the boxes. The simplest baseline, called **Appearance**, is to directly classify the predicate based on the concatenated visual features of the subject, object, and union boxes. The second baseline, **Appearance + spatial**, concatenates spatial features described in Chapter 3.1.3 with the appearance features. Both methods learn a single projection matrix, but use the same weight regularization as described in Chapter 3.1. The results confirm that adding spatial features to purely appearance-based features significantly improves performance. The third baseline, **Summation**, uses summation instead of subtraction in Eq. (3.2). Since this formulation is very similar to UVTransE, we run it to validate the effectiveness of the subtractive model. Across all our metrics, the results for Summation are 1-2% below those of UVTransE. This shows that despite the superficial similarity, the subtractive model better captures the structure of the VRD problem. The baseline in the fourth line of Table 3.2 is our own re-implementation of **VTransE** [1], for which we found that we had to add our regularization terms (Eq. 3.3) to achieve results comparable to [1]. We show the performance of two variants: without a language model (**UVTransE [V]**), or with our Bi-GRU language model (**UVTransE [V+L]**). Both variants include our spatial features. Compared to **VTransE [V]**, **UVTransE [V]** boosts performance significantly both in the general and in the zero-shot case. For the predicate detection task, the absolute improvement is about 5% in the general case and 10% in the zero-shot case, confirming that incorporating the union box in the translational formulation helps to isolate predicate information, particularly for rare and previously unseen cases. Adding the language module benefits **VTransE** and **UVTransE** about the same, although the absolute improvements are smaller in the zero-shot case than in the general case ($\sim$ 2 - 3% vs. $\sim$ 5%) as the language model tends to bias predictions towards relationships seen during training.

Comparison to the state of the art. Next, we compare performance to an extensive collection

	Predicate Det.		Phrase Det.				Relationship Det.			
	All	Zero-shot	All		Zero-shot		All		Zero-shot	
	R@50	R@50	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
Appearance	18.17	7.44	8.59	10.68	5.34	10.11	7.52	9.11	4.82	8.97
Appearance + spatial	38.89	14.35	20.06	24.70	7.98	11.84	17.02	20.54	6.90	10.02
Summation	49.01	18.52	21.93	27.80	10.25	14.94	17.78	21.37	9.47	13.33
VTransE [V] (our impl.)	45.12	12.84	19.74	25.62	7.27	10.61	16.21	20.48	6.31	9.55
VTransE [V+L] (our impl.)	50.11	15.31	26.13	31.40	8.73	12.05	22.23	26.14	7.67	10.99
UVTransE [V]	49.98	22.92	23.92	29.57	11.77	17.41	20.22	24.13	10.21	15.92
UVTransE [V+L]	**55.46**	**26.49**	**30.01**	**36.18**	**13.07**	**18.44**	**25.66**	**29.71**	**11.00**	**16.78**

Table 3.2: Comparisons of baselines to our proposed method on the Stanford VRD dataset. **Bold** indicates highest numbers.

of VRD models from the recent literature, which are summarized in Table 3.3. **VLK** [12] is a two-stage model that uses both appearance features and language priors for relationship prediction. **VTransE** [1] is the main method we build upon, as discussed previously. Note that the results reported by [1] differ from those of our re-implementation discussed above due to the use of different detectors and our inclusion of norm regularization in the training objective. **VRL** [40] applies a deep variation-structured reinforcement learning framework to sequentially discover object relationships and attributes using appearance and language features. **SA-full**, the fully supervised version of the method of Peyre et al. [21], uses appearance and spatial features to handle multi-modal relations and generalize well to unseen triplets. **DR-Net** [39] exploits statistical dependencies between objects and their relationships when modeling relations. **DSR** [57] designs a ranking objective that enforces the annotated relationships to have higher scores than negative examples. **CCA** [14] utilizes multiple CCA embedding cues (both vision and language), along with an SVM for ranking relationship proposals. **LK** [44] distills large-scale external linguistic knowledge from Wikipedia to achieve better performance for rare relationships. **CAIR** [43] builds one classifier for each predicate, but the classifier parameters are also adaptive to the context, i.e. (*subject*, *object*) pairs. **Zoom-Net** [58] encourages deep message interactions between local object features and global predicate features to recognize relationships. **RelDN** [42] is one of the most recent methods that achieves state-of-the-art performance using graphical contrastive losses to better learn subtle subject-object associations. Finally, **LS-VRD** [41] learns a visual and a semantic module that map features from the two modalities into a shared space such that the relations are discriminative.

It must be stated that getting completely apples-to-apples comparisons against the above methods is difficult as they vary in a number of respects. Among the most important is the quality of the underlying object detector, which depends on the network architecture and training protocol (details of training are usually not fully discussed in the papers, nor are the accuracies of the detector always reported). Other factors include the feature descriptors (in particular, whether spatial or linguistic features are included), the use of external data for training detectors or language model, the type of

	Detector (pre-training)	mAP	ROI feature	Spatial feature	Language feature	Joint reasoning	Extra training data
VLK [12]	VGG			-	✓	-	-
VTransE [1]	VGG	13.98		✓	-	-	-
VRL[40]	VGG (ImageNet)			-	✓	✓	-
SA-full [21]	VGG (ImageNet)			✓	-	-	-
CCA [14]	VGG (COCO)			✓	✓	-	-
LK [44]	VGG			✓	✓	-	Wikipedia
CAIR [43]	VGG			✓	✓	-	-
Zoom-Net [58]	VGG			✓	-	✓	-
RelDN [42]	VGG (COCO)		Align	✓	✓	-	-
LS-VRD [41]	VGG (COCO)			-	✓	-	-
DR-Net [39]	VGG (ImageNet)			✓	-	-	-
DSR [57]	VGG			✓	✓	-	-
UVTransE [V+L]	VGG (ImageNet)	19.10	Align	✓	✓	-	Visual Genome

Table 3.3: Summary of state-of-the-art methods on the VRD dataset. The 'Detector' column lists the architecture of the detector and the dataset used for pre-training (if mentioned in the original paper). 'ROI feature' indicates the type of ROI feature used (in papers that do not explicitly mention using ROIAlign, we assume ROIPool is used). 'mAP' lists the accuracy of the detector. 'Spatial feature' and 'language feature' indicate whether bounding box features similar to the ones of Chapter 3.1.3 and a language model similar to the one of Chapter 3.1.2 are used. 'Joint reasoning' indicates whether the method uses context or joint reasoning instead of predicting each pairwise relationship separately. 'Extra training data' indicates whether additional data is used for training either the detector or the language model. In each column, ✓indicates the presence of features, - indicates absence, and blank means the information is not provided in the original paper.

inference performed, the evaluation protocol, and so on. In an attempt to be transparent about these sources of variation, we list them in Table 3.3. With these caveats in mind, Table 3.4 compares our results to published numbers from the above papers the full VRD test set. Our model with the visual feature alone, UVTransE [V], reaches comparable performance to CAIR and Zoom-Net. After including the language module, we outperform all methods except for the most recent RelDN

	Predicate Det.	Phrase Det.						Relationship Det.					
		$k=1$		$k=10$		$k=70$		$k=1$		$k=10$		$k=70$	
	R@50	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
VLK [12]	47.87	16.17	17.03	-	-	-	-	13.86	14.07	-	-	-	-
VTransE [1]	44.76	19.42	22.42	-	-	-	-	14.07	15.20	-	-	-	-
VRL [40]	-	21.37	22.60	-	-	-	-	18.19	20.79	-	-	-	-
SA-full [21]	50.40	16.70	18.10	-	-	-	-	14.90	16.10	-	-	-	-
CCA [14]	-	-	-	16.89	20.70	-	-	-	-	15.08	18.37	-	-
LK [44]⊛	55.16	23.14	24.03	26.47	29.76	26.32	29.43	19.17	21.34	22.56	29.89	22.68	31.89
CAIR [43]	-	24.04	25.56	-	-	-	-	20.35	23.52	-	-	-	-
Zoom-Net [58]	50.69	24.82	28.09	-	-	29.05	37.34	18.92	21.41	-	-	21.37	27.30
RelDN [42]	-	**31.34**	**36.42**	**34.45**	**42.12**	**34.45**	**42.12**	25.29	28.62	**28.15**	33.91	**28.15**	33.91
LS-VRD [41]	-	28.93	32.85	32.90	39.66	32.90	39.64	23.68	26.67	26.98	32.63	26.98	32.59
DR-Net [39]★	80.78	-	-	-	-	19.93	23.45	-	-	-	-	17.73	20.88
DSR [57]★	86.01	-	-	-	-	-	-	-	-	-	-	19.03	23.29
UVTransE [V+L]	**55.46**	30.01	36.18	31.82	40.43	31.51	39.79	**25.66**	**29.71**	27.41	**34.55**	27.32	**34.11**

Table 3.4: Full test set performance on the Stanford VRD dataset. **Bold** indicates highest numbers, underline indicates second-highest.⊛ indicates use of large-scale external Wikipedia data. ★ indicates $k = 70$, instead of $k = 1$ for predicate detection (See Sec. 3.2.1).

	Predicate Det.	Phrase Det.		Relationship Det.	
	R@50	R@50	R@100	R@50	R@100
VLK [12]	8.45	3.36	3.75	3.13	3.52
VTransE [1] [1]	-	2.65	3.51	1.71	2.14
VRL [40]	-	9.17	10.31	7.94	8.52
SA-full [21]	<u>23.60</u>	7.4	8.7	7.1	8.2
CCA [14]	-	10.86	15.23	9.67	13.43
LK [44]⊛	16.98	<u>13.01</u>	<u>17.24</u>	**12.31**	<u>16.15</u>
CAIR [43]	-	10.78	11.30	9.54	10.26
Zoom-Net [58]	-	-	-	-	-
RelDN [42]	-	-	-	-	-
LS-VRD [41]	-	-	-	-	-
DR-Net [39]	-	-	-	-	-
DSR [57]★	60.90	-	-	5.25	9.20
UVTransE [V+L]	**26.49**	**13.07**	**18.44**	<u>11.00</u>	**16.78**

Table 3.5: Zero-shot performance on the Stanford VRD dataset. **Bold** indicates highest numbers, <u>underline</u> indicates second-highest. ⊛ indicates use of large-scale external Wikipedia data. ★ indicates $k = 70$, instead of $k = 1$ for predicate detection (See Sec. 3.2.1).We treat k as a hyper-parameter that can be cross validated for phrase and relationship detection. In our case, $k = 10$.

(which uses a different detector pre-trained on COCO [59]).

Table 3.5 presents a comparative evaluation for the zero-shot setting. Our method surpasses all other methods that use only the given dataset for training, and is comparable to LK, which incorporates external language data. Significantly, several of the strongest methods from Table 3.4, including Zoom-Net, RelDN, and DR-Net, do not report their results for the zero-shot setting at all. At least in some cases, this is because achieving high performance on common relations comes at the cost of very low performance on rare relations. In particular, we tested the RelDN model published by the authors [42] on the zero-shot test set and obtained accuracies close to 0 on all metrics, with almost all rare relationships being confidently classified as 'no relationship'.

Qualitative results. Figure 3.1 shows example predictions by our model for both seen and unseen relationships. There are many plausible detected triplets that are marked as negatives due to the lack of annotations (Missing GT column). In some cases, predicates are not mutually exclusive. For example, (*person*, *on*, *bike*) can also be labeled as (*person*, *ride*, *bike*); however, predicting *ride* for this pair of objects is penalized due to the missing ground truth.

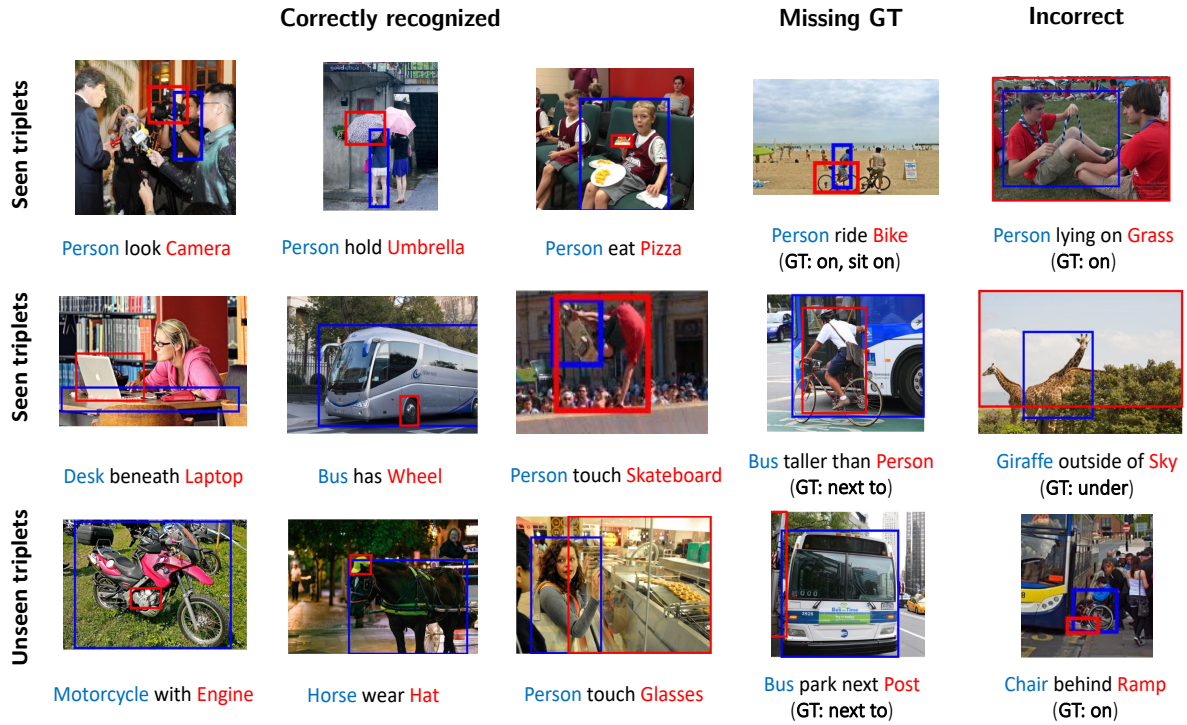

Figure 3.1: Examples of relationship detection on the VRD test split. A triplet is correctly recognized if both the bounding boxes are correctly localized and the predicate matches the ground truth. The 'Missing GT' column shows relationships that were marked as incorrect since they are not present in the ground truth. The 'Incorrect' column shows legitimate mistakes. The last row shows our zero-shot results.

3.2.2 Results on the UnRel Dataset

Dataset. To further investigate the generalization ability of our method, we perform experiments on the UnRel dataset [21]. It consists of 1071 images with 76 unusual triplets, such as (*person*, *ride*, *dog*), (*car*, *under*, *elephant*), etc. The ground truth on this dataset is more exhaustively annotated than on VRD.

UnRel contains too few images for training, so we simply use it as a test set for ourUVTransE model trained on the VRD dataset. The hyperparameters are the same as in Chapter 3.2.1.

Evaluation metrics. Following [21], we evaluate retrieval and localization with mAP over triplet queries (s, p, o) in two settings:

1. **With ground truth:** We are given GT pairs of boxes (b_s, b_o) and then rank them based on their predicate scores z_p (Eq. 3.4). The purpose of this setup is to test the "predicate prediction" part only, without the contribution of the object detector.

2. **With candidates:** Candidate boxes (b_s, b_o) are provided by the object detector and ranked according to the combined score $z_{(s,p,o)}$ (Eq. 3.5). In this setting, we also have to evaluate the

	With GT	With candidates		
		union	subj	subj/obj
Chance	38.4	8.6	6.6	4.2
DenseCap [60]	-	6.2	6.8	-
VLK [12]	50.6	12.0	10.0	7.2
SA-full [21]	62.6	14.1	12.1	9.9
UVTransE [V]	70.6	**19.2**	**17.2**	**14.8**
UVTransE [V+L]	**71.7**	18.0	16.3	14.1

Table 3.6: Retrieval on UnRel (mAP) with IoU=0.3.

accuracy of localization. According to [21], a candidate pair of boxes is positive if its IoU with GT pair is above 0.3. There are three localization metrics: **mAP-subj**: the subject box itself should have at least 0.3 overlap with its GT; **mAP-union**: the entire relationship is localized as one bounding box and it should have at least 0.3 overlap with the GT; **mAP-subj/obj**: Both subject and object boxes should have at least 0.3 overlap with their corresponding GT boxes.

Comparison with state of the art. We compare our results with numbers from four methods reported by [21]. The **chance** baseline randomly orders the proposals. The second method is **DenseCap** [60], where the output bounding box is interpreted as either a subject box or a union box for evaluation, as suggested in [21]. **VLK** [12] is the result from the re-implementation of [12] by [21]. Finally, **SA-full** [21] is, to our knowledge, the state-of-the-art fully supervised method on UnRel. As previously mentioned, our model is only trained on the Stanford VRD dataset, and is evaluated on the UnRel dataset without any changes, similar to the VLK and SA-full methods.

The retrieval results in Table 3.6 show that our model consistently outperforms all other methods. Interestingly, our language module improves the accuracy when the ground truth boxes are given, but degrades it slightly when the objects are provided by the object detector, likely because the language model gets confused if the predicted object classes are wrong, or the boxes are incorrectly localized. This behavior is different from what we observed on zero-shot evaluation for the VRD dataset, since the images in UnRel are deliberately unusual and hard.

Figure 3.2 shows the top triplets retrieved by our model for some representative queries. We use red boxes around images to indicate wrongly retrieved examples. It can be seen that we are able to successfully retrieve examples of rare relations such as (*elephant*, *wear*, *glasses*), and (*hat*, *on top of*, *building*).

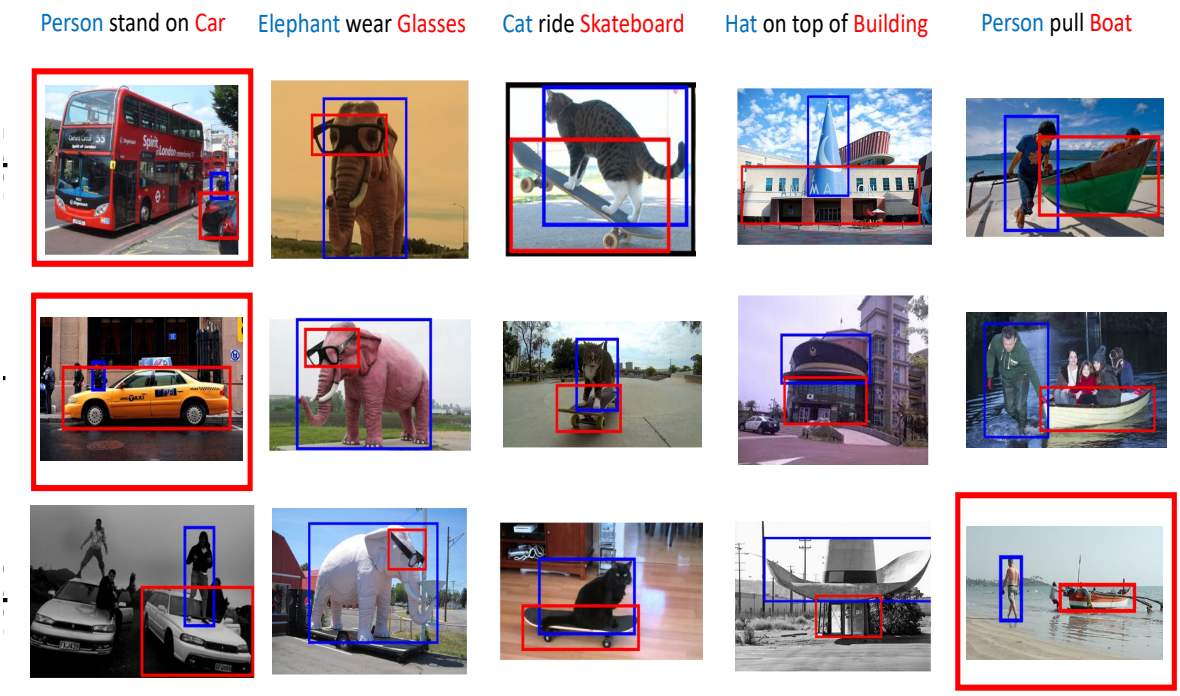

Figure 3.2: Top three retrievals for a set of UnRel triplet queries with our model. A relationship is marked as positive if the subject and object boxes have $IoU \geqslant 0.3$ with the ground truth. Otherwise, it is marked as an error (red box around the entire image).

3.2.3 Results on the Visual Genome Dataset

Datasets. To demonstrate the effectiveness of our model on large-scale datasets that are more geared towards scene graph generation, we perform experiments on two cleaned subsets of Visual Genome [22]. The first one, created by Xu et al. [45], is composed of the most frequent 150 objects and 50 predicates. We call this one **VG-IMP** after the method of [45]. After pre-processing, VG-IMP is split into training and test sets containing 75,651 images and 32,422 images, respectively. The second subset, created by Zhang et al. [1], contains an even larger number of objects and predicates, 200 and 100, respectively. We follow the same 73,801/25,857 train/test split as in [1]. We call this subset **VG-VTransE**.

Implementation details. On the VG-IMP subset, we train a Faster R-CNN detector with a VGG-16 backbone to obtain an mAP of 19.2%. Competing methods using the same backbone report higher performance, namely, 20.0% for Neural Motifs [24], 20.4% for Graph R-CNN [47], and 25.5% for RelDN [42]. Maximizing the accuracy of object detection is not the focus of our work, but to better compete with these methods on the final accuracies for relationship localization, we trained a stronger detector using a ResNet-101 backbone, for an mAP of 23.8%. For the VG-VTransE subset, the only competing method with published results is VTransE [1], which uses a VGG-16 backbone. On that subset, the mAP of our VGG-16 detector is 12.5%, which is sufficient to compete with [1].

We initialize the parameters in Faster R-CNN with ImageNet pre-training. After fine-tuning the parameters on the respective Visual Genome subset, we fix it, and train our UVTransE module along with the language module with initial learning rate of $1e^{-2}$. At test time, for each image, we use the top 50 candidate object proposals ranked by Faster R-CNN for mining relationships.

To get good performance on Visual Genome evaluation metrics (described below), we found it useful to add a 'background' or 'no relationship' class during training. We define positive relation triplets as those where both subject and object have $IoU \geqslant 0.5$. During training, for each image, we sample 32 relations with the ratio of positive to negative triplets being $1 : 3$. On the VRD dataset (Chapter 3.2.1), this kind of sampling improves performance for common relationships, but significantly degrades performance for the zero-shot case, as many unseen relationships get classified as 'background' with high confidence.

For the results of this chapter, we also found it necessary to change Eqs. (3.5) and (3.7) to use product instead of addition:

$$z_{(s,p,o)} = z_s \times z_o \times z_p \,, \tag{3.9}$$

and for **UVTransE [V+L]**,

$$z_{(s,p,o)} = (\alpha z_p + (1-\alpha) z_{lang_p}) \times z_s \times z_o \,. \tag{3.10}$$

In the experiments of this chapter, the UVTransE hyperparameters are $C = 0.1$ and $\alpha = 0.5$.

Evaluation metrics. To evaluate on the VG-IMP subset, we follow a methodology consistent with [45] and report performance for the following three settings.

1. **Predicate Classification (PredCls)**: Given ground truth boxes and their corresponding objects, predict the predicate between object pairs. This is the same as predicate detection of Chapter 3.2.1.

2. **Phrase Classification (PhrCls)**: Given ground truth boxes, recognize the objects and their relations.

3. **Scene Graph Generation (SGGen)**: Predict objects, boxes (IoU $\geqslant 0.5$) and the relations between object pairs directly from an image. This is equivalent to relationship detection in Chapter 3.2.1.

On the VG-VTransE subset, we follow [1] and report the performance for phrase and relationship detection, defined as in chapter 3.2.1.

For both subsets, we use Recall@50 and Recall@100 to evaluate how many labelled relationships are hit in the top 50 or 100 predictions. We follow related works in enforcing that for a given subject

	Detector (pre-training)	mAP	ROI feature	Spatial feature	Language feature	Joint reasoning
IMP [45]	VGG (COCO)		Pool	-	-	✓
MSDN [46]	VGG (ImageNet)		Pool	-	-	✓
Neural Motifs [24]	VGG (ImageNet)	20.0	Align	✓	✓	✓
Graph R-CNN [47]	VGG (ImageNet)	20.4	Align	✓	-	✓
RelDN [42]	VGG (COCO)	25.5	Align	✓	✓	-
LS-VRD [41]	VGG (COCO)	-	-	-	✓	-
UVTransE [VGG+V+L]	VGG (ImageNet)	19.2	Align	✓	✓	-
UVTransE [ResNet+V+L]	ResNet (ImageNet)	23.8	Align	✓	✓	-

Table 3.7: Summary of state-of-the-art methods on the VG-IMP dataset. See caption of Table 3.3 for explanation of the columns.

and object bounding box, the system must not output multiple predicate labels, which is the same as setting $k = 1$ in the VRD dataset [12].

Comparison with state of the art. Table 3.7 summarizes different state-of-the-art methods on the VG-IMP subset. **IMP** [45] uses standard RNNs and learns to iteratively improve its predictions via message passing between predicates. **MSDN** [46] jointly refines the features for different tasks by passing messages along a dynamically constructed graph. **Neural Motifs** [24] proposes a Stacked Motif Network to capture higher-order motifs in scene graphs. **Graph R-CNN** [47] utilizes attentional graph convolutional networks to learn to modulate information flow through unlikely edges in the scene graph.

Comparative evaluation results on VG-IMP are shown in Table 3.8. We can see that our model with the ResNet detector (**UVTransE (ResNet+V+L)**) outperforms all methods except the very recent RelDN, whose detector is even more accurate than ours, and LS-VRD. In particular, we get better performance than several methods that include message passing or graph CNNs to jointly

	PredCls		PhrCls		SGGen	
	R@50	R@100	R@50	R@100	R@50	R@100
IMP [45]	44.8	53.1	21.7	24.4	3.4	4.2
MSDN [46]	63.1	66.4	19.3	21.8	7.7	10.5
Neural Motifs [24]	65.2	67.1	35.8	36.5	27.2	30.3
Graph R-CNN [47]	54.2	59.1	29.6	31.6	11.4	13.7
RelDN [42]	**68.4**	**68.4**	**36.8**	**36.8**	28.3	32.7
LS-VRD [41]	**68.4**	**68.4**	<u>36.7</u>	<u>36.7</u>	27.9	32.5
UVTransE [VGG+V]	59.7	63.3	30.7	31.9	25.2	28.3
UVTransE [VGG+V+L]	61.2	64.3	30.9	32.2	25.3	28.5
UVTransE [ResNet+V]	64.4	66.5	35.0	36.1	<u>29.9</u>	<u>33.2</u>
UVTransE [ResNet+V+L]	<u>65.3</u>	<u>67.3</u>	35.9	36.6	**30.1**	**33.6**

Table 3.8: Full test set performance on the VG-IMP dataset. **Bold** indicates highest numbers, <u>underline</u> indicates second-highest.

	Phr. Det.		Rel. Det.	
	R@50	R@100	R@50	R@100
VTransE [1]	9.46	10.45	5.52	6.04
UVTransE [V]	15.47	19.70	8.52	10.59
UVTransE [V+L]	**17.53**	**21.92**	**9.55**	**11.74**

Table 3.9: Full test set performance on the VG-VTransE dataset. **Bold** indicates highest numbers.

reason about multiple relationships. We also observe that our language module does not enjoy as significant a gain as in Table 3.4. This is likely due to the fact that there are far more relations in the Visual Genome dataset than in VRD, so the training data for the language model is sparser.

Table 3.9 reports results on the VG-VTransE subset, which has an even larger number of object classes and relationships than VG-IMP. Here, as in Chapter 3.2.1, we can once again observe significant improvements over VTransE.

Qualitative results of scene graph generation. In Figure 3.3, we show some example outputs of scene graph generation using **UVTransE[V+L]** on Visual Genome. Through careful inspection, we can see that UVTransE generally fails in two cases: either the object detector cannot find the objects present in the ground truth, which are highlighted with orange boxes, or the spatial configuration makes it hard to predict the predicate. For instance, in the image with the pelican (bottom right), there is a predicted false positive: (*wing-1*, *has*, *wing-2*). In addition, many seemingly correct relations are marked as false positives due to incomplete ground truth. For example, (*racket-1*, *in*, *hand-1*) is a plausible relation in the top left image of Figure 3.3; however, it does not exist in the annotations.

3.2.4 Results on the Open Images Dataset

Dataset. Our final set of experiments is on the Open Images dataset [23], which is even larger than Visual Genome: 94,747 training and 5,775 validation images according to the recommended split. On the other hand, the number of object classes and predicates is smaller, only 57 and 10, respectively. Among the 10 predicates, there is one special predicate, *is*, which is used to describe visual attributes, e.g., (*table*, *is*, *wooden*). Therefore, in addition to relation prediction, we also have to adapt our method to perform attribute prediction.

Implementation details. We use Faster R-CNN with ResNet-101 backbone as the object detector and region feature extractor. We initialize the network with weights pre-trained on COCO [59] and fine-tune on Open Images to achieve an mAP of 51% on our validation split. Similar to the Visual Genome setup described in chapter 3.2.3, we freeze the detector and only train our UVTransE

Figure 3.3: Example scene graphs generated on VG-IMP images. In the images, green boxes are objects detected with $IoU \geqslant 0.5$, while orange boxes are ground truth objects that are not detected by our pipeline. In the scene graphs, green ellipses are true positive relations recognized by our model at Recall@20, orange ellipses are false negatives, and magenta ellipses are false positives (sometimes due to missing ground truth).

module along with the language module with $C = 0.1$ and $\alpha = 0.5$. During training, 25% of triplets in each batch are positive. In test time, we select the top 50 candidate proposals from Faster R-CNN for mining relationships, and the final triplet scores are calculated with Eq. (3.9) and Eq. (3.10) for UVTransE [V] and UVTransE [V+L] predictions, respectively.

In order to tackle the predicate *is*, we use the same object proposals generated by Faster R-CNN and train an additional classifier on each proposal to output the probability for each attribute. The

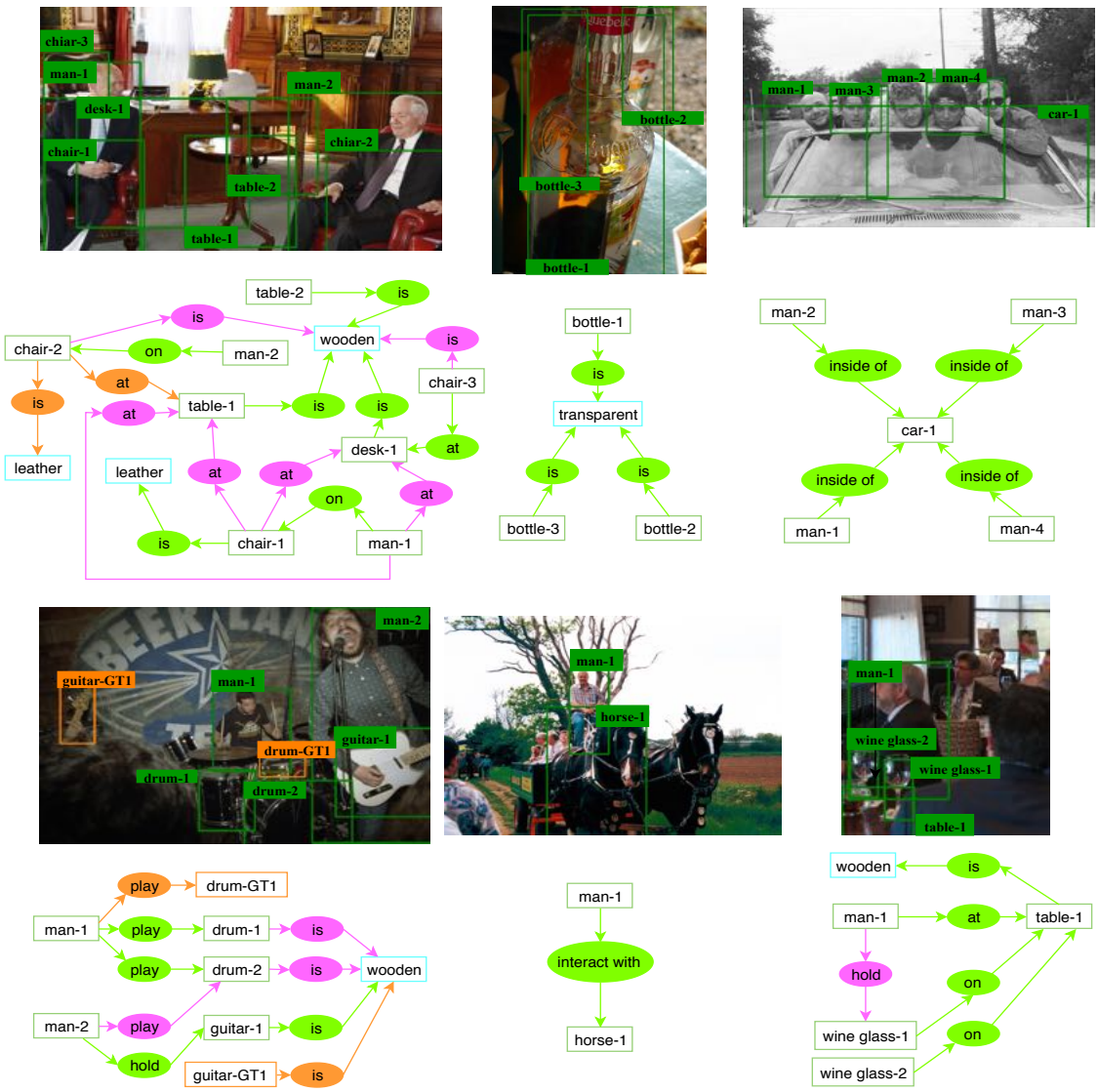

Figure 3.4: Example scene graphs generated on Open Images. In the images, green boxes are objects detected with $IoU \geqslant 0.5$, while orange ones are ground truth objects that are not detected. In the scene graphs, attributes are represented with cyan boxes. Green ellipses are true positive relations recognized by our model at Recall@20, orange ellipses are false negatives, and magenta ellipses are false positives.

attribute score is calculated with

$$z_{(s,is,a)} = z_s \times z_a, \tag{3.11}$$

where a is the attribute and z_a is the output probability from the attribute classifier.

Evaluation metrics. In the Open Images Challenge, results are evaluated based on Recall@50 of relationship detection (R@N$_{rel}$), mAP of relationship detection (mAP$_{rel}$), and mAP of phrase detection (mAP$_{phr}$). The final score is calculated with $0.2 \times R@N_{rel} + 0.4 \times mAP_{rel} + 0.4 \times mAP_{phr}$.

	Public	Private	Full
tito	0.256	0.237	0.243
kyle	0.280	0.235	0.249
RelDN [42]	<u>0.320</u>	**0.332**	**0.328**
UVTransE [V]	0.285	0.246	0.258
UVTransE [V+L]	**0.321**	<u>0.273</u>	<u>0.287</u>

Table 3.10: Results on Open Images Challenge for the top three teams on the public leaderboard vs. our methods. These values are evaluated based on the official mAP_{rel}, mAP_{phr}, and Recall@50 for relationship detection. Public and private correspond to 30% and 70% of test data respectively. Full is $0.3\times$ *public* $+0.7\times$ *private*. **Bold** indicates highest numbers. <u>Underline</u> indicates second highest.

The mAP_{rel} takes the mean of AP for each predicate, where true positive is defined as having correct object boxes ($IoU \geqslant 0.5$), classes, and predicates. The mAP_{phr} is similar to mAP_{rel}, but applied to the union of subject and object boxes instead of individual boxes.

Comparison with state of the art. We compare our results with other models from the official kaggle competition. There are 99,999 test images, and the official test set is split into public and private sets, which contain 30% and 70% of test data, respectively. We present results for both splits in Table 3.10. We also have an additional column, named "Full", for overall performance, which is calculated by $0.3\times$ *public_score* $+0.7\times$ *private_score*. As shown in Table 3.10, we surpass most teams except for RelDN, who once again use a better object detector (Faster R-CNN with ResNeXt-101-FPN). Notice also the large gap between **UVTransE [V+L]** and the second place (**kyle**) considering the low absolute scores and the large amount of test images.

Qualitative results of scene graph generation. Figure 3.4 presents examples of generated scene graphs on Open Images. Our model is able to cover different kinds of relations, including positional predicates such as *on*, attributive predicates such as *is*, and interactive predicates such as *play*. Similar to the results on Visual Genome, UVTransE has a hard time when the spatial configuration is challenging. Take the top left image, which contains two people sitting on chairs as an example. We can see that our model outputs (*man-1*, *at*, *table-1*), whose spatial structure is quite similar to other relationships that involve *at*, such as (*chair*, *at*, *desk*).

CHAPTER 4: VIDEO WORK

4.1 CORRELATION NETWORK

In this section, we describe our proposed correlation network in detail. We start by defining the correlation operation used in our network. We then briefly introduce the backbone architecture that we use for action recognition. Finally, we discuss how we incorporate the correlation operator with the backbone to leverage temporal information.

Correlation operator Suppose that there are two feature maps f_1 and f_2 both with dimension $C \times H \times W$, where C, H, and W indicate the channel, height, and width respectively. The correlation of patch p_1 in f_1 with patch p_2 in f_2 encodes the similarity between these two patches. To make the computation more tractable, the correlation operator we use is defined between two pixels only. Thus, the correlation between a single pixel (x_1, y_1) in f_1 and another pixel (x_2, y_2) in f_2 is given by Eq. 4.1

$$c(x_1, y_1, x_2, y_2) = \frac{\sum_{c=1}^{C} f_1^c(x_1, y_1) f_2^c(x_2, y_2)}{\|f_1(x_1, y_1)\|_2 \, \|f_2(x_2, y_2)\|_2} \tag{4.1}$$

In our model, we further limit (x_2, y_2) to be within a $K \times K$ neighborhood of (x_1, y_1) and perform the correlation operation for every pixel in f_1. Therefore, the output size of the correlation operator is $K \times K \times H \times W$. The $K \times K$ dimension can then be flatten to generate a feature map of size $K^2 \times H \times W$, where K^2 becomes the output channel dimension. To extend this operation to a video clip with T frames, we compute correlation for every pair of adjacent frames in the given input clip. Since this gives us $T - 1$ output only, we pad the sequence with a self-correlation of the first frame to make the output length consistent with the input.

Backbone The backbone we used is specified in Table 4.1. We follow [18] to inflate the original ResNet50 [56] architecture, and [51] to separate out the 3D convolution into temporal and spatial convolution. Since we want to keep as much temporal information as possible for correlation calculation, we perform temporal striding at the later stages (*res4* and *res5*).

Correlation network In our correlation network, we follow the two-stream architectures, and have one appearance and one motion stream. For the motion stream, we remove the *res5* stage shown in Table 4.1, and apply the correlation operator after *res2*, *res3* and *res4*. After calculating the correlation features, we apply convolutions with stride to reduce all spatial dimension to 14×14.

Layers	I3D Res50	Output size
res1	$1 \times 7 \times 7$, 64, stride $1 \times 2 \times 2$	$16 \times 112 \times 112$
pool1	$1 \times 3 \times 3$, stride $1 \times 2 \times 2$	$16 \times 56 \times 56$
res2	$\begin{pmatrix} 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 3 \times 1 \times 1, 64 \\ 1 \times 1 \times 1, 256 \end{pmatrix} \times 3$	$16 \times 56 \times 56$
res3	$\begin{pmatrix} 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 3 \times 1 \times 1, 128 \\ 1 \times 1 \times 1, 512 \end{pmatrix} \times 4$	$16 \times 28 \times 28$
res4	$\begin{pmatrix} 1 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 3 \times 1 \times 1, 256 \\ 1 \times 1 \times 1, 1024 \end{pmatrix} \times 6$	$8 \times 14 \times 14$
res5	$\begin{pmatrix} 1 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 512 \\ 3 \times 1 \times 1, 512 \\ 1 \times 1 \times 1, 2048 \end{pmatrix} \times 3$	$4 \times 7 \times 7$
global average pool, fc		#classes

Table 4.1: The I3D ResNet backbone for building correlation network.

We then concatenate these feature maps, and feed into a residual block similar to *res3* stage, but with 3 blocks only. Finally, we average pool the features to get an output of size 512. This vector is then concatenated with the output of appearance stream for the final action classification.

4.2 EXPERIMENTS

Dataset We evaluate our model on Something-Something v1 dataset [2]. This dataset includes 110k videos of 174 different low-level actions, each lasting between 2 to 6 seconds. It is composed of humans performing actions with everyday objects. The same action is performed with different objects so that models are forced to understand the actions instead of recognizing the objects. It is therefore an interesting question whether our model actually recognizes the actions.

Training and Testing During training, we apply temporal jittering and sample a clip of 16 frames, covering roughly 2.67 seconds, from the training data. We resize the input video to have shorter side randomly sampled in [256, 320] pixels, and perform a center crop of size 224×224. We set $K = 49, 25,$ and 13 respectively for *res2*, *res3* and *res4*. We use a minibatch size of 64 with

Methods	Top 1 Accuracy
TRN	42.0
R(2+1)D	45.7
NL I3D	44.4
GCN	46.1
S3D-G	**48.2**
Ours [RGB]	45.6
Ours [Corr]	39.8
Ours [RGB + Corr]	**48.2**

Table 4.2: Comparing the result of our correlation network with the state-of-the-art on Something-Somthing v1dataset.

synchronous batchnorm. We first train the appearance stream for 35k iterations with initial learning rate 0.01 and learning rate decay at 25k. Next, we train the motion stream for 45k with the same initial learning rate and learning rate decay at 35k. Finally, we finetune the overall network for 20k. We set the weight decay to 10^{-4} and the dropout of 0.5 is applied after the average pool. For testing, we sample 10 clips uniformly spaced out in the video and average the clip-level predictions as the video-level results.

Results We compare our correlation network with the state-of-the-art methods shown in Table 4.2. **TRN** [61] extends the relational reasoning module [62] to temporal domain to learn the temporal dependencies between video frames at multiple time scales. **R(2+1)D** [51] factorized the 3D convolution into spatial and temporal component for better accuracy. **NL I3D** [40] designed the non-local block which applied self attention layer to model the long range temporal and spatial interactions. **GCN** [63] represented video as space-time region graphs and applied graph convolution to learn the object relationships in video. **S3D-G** [52] applied separable 3D convolution at the top of the network, and 2D convolution at the bottom for better speed-accuracy trade-offs.

As shown in Table 4.2, our model with the additional correlation features improved its RGB counterpart by roughly 3%. This shows the effectiveness of the correlation modeling, and its ability to model the temporal information.

CHAPTER 5: CONCLUSIONS

In this thesis, we introduced the UVTransE framework for visual relationship detection, which extends the VTransE framework [1] by adding a union box feature to the subject and object box features for learning the embedding of the predicate. While our original motivation was primarily to improve zero-shot performance of VTransE, extensive experiments have demonstrated that our UVTransE model achieves state-of-the-art results in multiple challenging scenarios, from small-scale to large-scale, on both the full test set and zero-shot settings. The latter is a significant contribution, since some other state-of-the-art methods, like RelDN [42] achieve high accuracy on common relationships at the cost of low zero-shot performance. We obtain consistent improvements over prior work while keeping the formulation straightforward. The simplicity of our model combined with its versatility and high performance thus makes it a good practical choice for advanced visual reasoning tasks such as scene graph generation.

In addition, we tackled the action recognition task and designed a correlation network. Unlike previous approach, our network establishes frame to frame correspondences and makes the computation of motion explicit and end-to-end trainable. Through our experiments, we demonstrate its superior performance for action recognition on Something-Something dataset with RGB input only.

REFERENCES

[1] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *CVPR*, 2017.

[2] R. Goyal, S. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, and et al., "The something something video database for learning and evaluating visual common sense of higher-order image features." in *ICCV*, 2017.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[4] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[7] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *CVPR*, 2016.

[8] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *CVPR*, 2018.

[9] J. Johnson, R. Krishna, M. Stark, J. Li, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *CVPR*, 2015.

[10] N. Prabhu and R. V. Babu, "Attribute-graph: A graph based approach to image ranking," in *ICCV*, 2015.

[11] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Deep compositional question answering with neural module networks," in *CVPR*, 2016.

[12] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *ECCV*, 2016.

[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013.

[14] 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 *ICCV*, 2017.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

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

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[18] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[19] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *NIPS*, 2014.

[20] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

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

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, 2017.

[23] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv:1811.00982*, 2018.

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

[25] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *CVPR*, 2008.

[26] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *IJCV*, 2008.

[27] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with rcnn," in *ICCV*, 2015.

[28] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *ICCV*, 2013.

[29] A. Mallya and S. Lazebnik, "Learning models for actions and person-object interactions with transfer to question answering," in *ECCV*, 2016.

[30] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *CVPR*, 2011.

[31] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*, 2011.

[32] B. Yao and F.-F. Li, "Grouplet: A structured image representation for recognizing human and object interactions." in *CVPR*, 2010.

[33] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *CVPR*, 2010.

[34] A. Papazoglou, L. D. Pero, and V. Ferrari, "Discovering object aspects from video," *Image and Vision Computing*, 2016.

[35] R. Salakhutdinov, A. Torralba, and J. B. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *CVPR*, 2011.

[36] T. Mensink, E. Gavves, and C. G. M. Snoek, "COSTA: co-occurrence statistics for zero-shot classification," in *CVPR*, 2014.

[37] 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," in *EMNLP*, 2016.

[38] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," *IJCV*, 2017.

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

[40] X. Liang, L. Lee, and E. P. Xing, "Deep variation-structured reinforcement learning for visual relationship and attribute detection," in *CVPR*, 2017.

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

[42] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph generation," in *CVPR*, 2019.

[43] B. Zhuang, L. Liu, C. Shen, and I. Reid, "Towards context-aware interaction recognition for visual relationship detection," in *ICCV*, 2017.

[44] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation." in *ICCV*, 2017.

[45] D. Xu, Y. Zhu, C. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *CVPR*, 2017.

[46] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *ICCV*, 2017.

[47] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for scene graph generation," in *ECCV*, 2018.

[48] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2014.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842

[50] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, and et al., "The kinetics human action video dataset." in *CVPR*, 2017.

[51] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[52] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *ECCV*, 2018.

[53] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *CVPR*, 2018.

[54] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *TSP*, 1997.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[57] K. Liang, Y. Guo, H. Chang, and X. Chen, "Visual relationship detection with deep structural ranking," in *AAAI*, 2018.

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

[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*, 2014.

[60] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *CVPR*, 2016.

[61] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," *European Conference on Computer Vision*, 2018.

[62] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," *NIPS*, 2017.

[63] X. Wang and A. Gupta, "Videos as space-time region graphs," *European Conference on Computer Vision*, 2018.