University of California
Santa Barbara

Compositional Networks for Detecting and Localizing Activities

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

A S M Iftekhar

Committee in charge:

Professor B.S. Manjunath, Chair
Professor Kenneth Rose
Professor Shiv Chandrasekaran
Professor Suya You

December 2023

The Dissertation of A S M Iftekhar is approved.

Professor Kenneth Rose

Professor Shiv Chandrasekaran

Professor Suya You

Professor B.S. Manjunath, Committee Chair

November, 2023

Compositional Networks for Detecting and Localizing Activities

Dedicated to

my mother, Saleha Khatoon,

my wife, Nishat Subah Peau,

my father, MD Abdur Razzaque,

my brothers, Prof. Islam and Rizvi,

my sister-in-laws, Anika and Dr. Afroz,

my precious nieces, Asheeta and Ruzaina,

And to my friends, Sakib and Dr. Prince.

Acknowledgements

I began my Ph.D. journey with both excitement and worry. I was eager to learn, but also nervous about the challenges ahead. Thankfully, I received unwavering support and encouragement from both personal and academic quarters. I'd like to extend my heartfelt gratitude to all who stood by me in this section.

First and foremost I want to thank Prof. B.S. Manjunath. As my supervisor and committee chair he was instrumental on my research achievements. His patience and invaluable feedback consistently guided my work. He taught me to navigate the inevitable setbacks during my Ph.D., always highlighting the silver linings. I also like to thank all my committe members: Prof. Shivkumar Chandrasekaran, Prof. Kenneth Rose, and Prof. Suya You for their important contributions to my research. A very special acknowledgment to Prof. Suya You , who despite facing various adversaries always helped me. I would like to acknowledge the funding sources: US Army Research Laboratory (ARL) under agreement number W911NF202015 and NSF award SI2-SSI #1664172 for supporting my research.

Special thanks to all my collaborators over the years. My first mentor, Dr. Oytun Ulutan, played a pivotal role in jump starting my research. I had some amazing mentors: Dr. Hao Chen, Dr. Xinyu Li, Dr. Kaustav Kundu, and Dr. Joseph Tighe during my internship at AWS AI lab. I also forever grateful to Dr. Subhasis Das who was my manager during my internship at Zoox, Inc. Satish Kumar and Raphael Ruschel from the Vision Research Lab (VRL) have been invaluable collaborators, deeply influencing my research. Their contributions cannot be overstated. I want to acknowledge Dr. Ekta Prashnani and Dr. R. Austin Mcever for their contributions to my work.

I thoroughly enjoyed my time at VRL. I would love to thank my VRL lab mates Satish Kumar, Raphael Ruschel, S.Shailja, Devendra Jangid, Bowen Zhang, Amil Khan,

Connor Levenson, Chandrakanth Gudavalli, Dr. Jiaxiang Jiang, Dr. Mike Goebel, Dr. Ekta Prashnani, Dr. R. Austin Mcever, Dr. Oytun Ulutan, Dr. Poyu You, Dr. Angela Zhang for creating many beautiful memories over the time.

I consider myself immensely fortunate to have been surrounded by a steadfast network of family and friends throughout my Ph.D. journey. The sacrifices my parents made to ensure I reached this point in my life are beyond the realm of words. My wife, Peau, has been a pillar of unwavering love and support. I am blessed with the most supportive elder brothers, Prof. Raisul Islam and A S M Rizvi, who have backed me without reservations. Heartfelt thanks go out to my sisters-in-laws, Anika Sharin and Dr. Sabrina Afroz, for their consistent encouragement. My nieces, Asheeta and Ruzaina, have profoundly enriched my perspective on life. Mashnoon Alam Sakib and Dr. Golam Dastageer Prince, the closest friends of my life have always showered me with their love and kindness. To each of you, and to the supportive Bangladeshi community in Santa Barbara, my heartfelt gratitude for shaping my journey and illuminating my path with warmth.

Curriculum Vitæ

A S M Iftekhar

Education

December 2023	**Doctor of Philosophy** Electrical and Computer Engineering University of California, Santa Barbara, USA.
December 2020	**Master of Science** Electrical and Computer Engineering University of California, Santa Barbara, USA.
Feb 2017	**Bachelor of Science** Electrical & Electronic Engineering Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

Experience

04/2019-12/2023	Graduate Student Researcher, Vision Research Lab, UCSB
06/2022-09/2022	Perception Intern, Zoox Inc
06/2021-09/2021	Applied Science Intern, AWS AI Labs

Publications

A S M Iftekhar, Satish Kumar, R. McEver, Suya You, B Manjunath, *GTNet: Guided Transformer Network for Detecting Human-Object Interactions*, In proceedings of the Pattern Recognition and Tracking XXXIV at SPIE commerce+ defense Program, April 2023, Orlando, FL.

A S M Iftekhar*, Raphael Ruschel*, Satish Kumar, Suya You, B Manjunath, *DDS: Decoupled Dynamic Scene-Graph Generation Network*, under review in IEEE Transactions of Image Processing (TIP), published in Arxiv, Jan 2023.(*Equal contribution)

Satish Kumar, **A S M Iftekhar**, Ekta Prashnani, B.S.Manjunath *LOCL: Learning Object-Attribute Compositionality using Localization*, British Machine Vision Conference (BMVC), November 2022, London, United Kingdom.

A S M Iftekhar*, Hao Chen*, Kaustav Kundu, Xinyu Li, Joseph Tighe, Davide Modolo, *What to Look at and Where: Semantic and Spatial Refined Transformer for Detecting Human-Object Interactions*, accepted for **oral presentation**, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, June 2022. (*Equal contribution)

O Ulutan*, **A S M Iftekhar***, B.S. Manjunath, *VSGNet: Spatial Attention Network for Detecting Human Object Interactions Using Graph Convolutions*, In Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, June 2020.(*Equal contribution.)

S Kumar, I Arevalo, **A S M Iftekhar**, B.S. Manjunath, *MethaneMapper: Spectral Absorption aware Hyperspectral Transformer for Methane Detection*, accepted in **CVPR highlights**, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, June 2023.

R. McEver, B. Zhang, C. Levenson, **A S M Iftekhar**, B.S. Manjunath, *Context-Driven Detection of Invertebrate Species in Deep-Sea Video*, International Journal of Computer Vision (IJCV), Springer, Jan 2023.

Satish Kumar, **A S M Iftekhar**, Michael Goebel, Tom Bullock, Mary H MacLean, Michael B Miller, Tyler Santander, Barry Giesbrecht, Scott T Grafton, B.S. Manjunath, *StressNet: Detecting Stress in Thermal Videos*, In proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, Jan 2021.

M Nahian, **A S M Iftekhar**, Mohammad Tariqul Islam, SM Mahbubur Rahman, Dimitrios Hatzinakos, *CNN-Based Prediction of Frame Level Shot Importance for Video Summarization*, Int. Conference on New Trends in Computing Sciences 2017, Jordan, Oct 2017.

Honors & Awards

2023	ECE Department Dissertation Fellowship, UCSB.
2023	Conference Travel Grant, SPIE commerce + defense Program
2022	CVPR Conference Travel Grant.
2022	Conference Travel Grant, UCSB Graduate Student Association.
2013	University Merit Scholarship, Bangladesh.

Served as Reviewer

2023 - Present	IEEE Transaction on Image Processing (TIP)
2022 - Present	Conference on Computer Vision and Pattern Recognition (CVPR)
2022 - Present	European Conference on Computer Vision (ECCV)
2021 - Present	Winter Conference on Applications of Computer Vision (WACV)

Abstract

Compositional Networks for Detecting and Localizing Activities

by

A S M Iftekhar

The development of automated methods capable of detecting and localizing actions is crucial for a variety of applications, ranging from surveillance and autonomous driving to content moderation. This thesis focuses on creating action detection methods that deliver robust performances. At the heart of these methods' robustness lie two fundamental elements: the detection of atomic actions and the ability for compositional understanding.

Atomic actions are those that are identifiable from a single image or a short video. In this research, we developed innovative methods to detect and localize such actions that achieve state-of-the art performance. The key strength of these methods lies in their ability to refine visual features both spatially and semantically, enabling precise identification of action-specific regions. For scalability, we further developed a multi-branch network to recognize new composition of objects and actions. Our design ensures that each branch learns decoupled features, allowing the network to transfer previously learned concepts to identify new compositions. This approach outperforms existing methods by a good margin as our extensive experiments on benchmark datasets demonstrate. Further, the correct identification of the attributes of the participating objects in actions helps to detect unknown compositions. Therefore, we have created a network utilizing spatially localized learning to correctly associate objects and attributes. This network achieves state-of-the-art performance in object-attribute association on cluttered scenes.

The developed methods in this thesis can do robust action detection at scale and serve as a base for numerous future applications.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In today's interconnected world, characterized by the ubiquitous presence of cameras and the seamless accessibility of the internet, the urgency to engineer automated systems capable of delving into the intricate analysis of videos and images has surged exponentially. The volume and diversity of visual content have reached a magnitude where the reliance on manual labor for comprehensive analysis has become impractical. Furthermore, a host of complex issues, including privacy concerns and biases in content assessment, have compounded the challenges inherent in constructing automated systems for visual content analysis.

Amid this complex backdrop, the focal point of this thesis resides in the creation of automated methods for content analysis. Specifically, we developed methods to detect actions within varied contextual settings. These methods have significant applications in both security and consumer domains. Through our analysis, we have identified two pivotal properties essential for an action detection method to achieve robustness. These properties are:

Atomic Action Detection: For any robust action detection method, the precise localization and detection of atomic actions is a fundamental requirement. Atomic actions encompass activities that can be inferred from a single image or a brief video snippet lasting 1-2 seconds [17]. These actions include basic movements like walking, standing, or sitting on a chair. Various related fields [1, 18] have coined diverse terms such as scene-graphs, HOIs for such actions. The identification of these actions are extremely important since multiple atomic actions can combine to form more complex composite actions. Consider the process of exiting a car, for instance. This action involves several atomic actions such as opening the door, standing up, all contributing to the overall action.

The Figure 1.1 provides two examples of atomic action detection. This localization process entails pinpointing each element participating in a specific action while concurrently predicting the action's class. The first half of our thesis delves into the development of robust models capable of proficiently performing atomic action detection.

Compositional Ability: The notion of compositionality refers to the capacity to combine familiar concepts to infer unfamiliar ones, a cognitive skill of great significance in human comprehension. Existing action detection methods often exhibit subpar performance in detecting actions beyond a predefined set. Nonetheless, encounters with unfamiliar actions are quite common. Two of such unusual actions are shown in Figure 1.2. Humans can trivially identify these actions using the power of compositionality over previously learned concepts, even in the absence of prior exposure to the exact combination of actions. Our underlying hypothesis is that achieving truly effective action detection in images and videos dependent on the method's compositional ability. Through the development of a series of innovative networks, the latter part of our thesis provides compelling evidence that supports our hypothesis.

Figure 1.1: Two examples of atomic action localization. We marked humans and objects involved in the action with green and red bounding boxes. The images are taken from V-COCO dataset [1].

The series of action detection methods that will be described in this thesis have extensive applications. Before going into the details of our developed methods, we will be giving a brief overview of these applications.

1.1 Motivation

Security Applications: In the domain of security, the extensive deployment of surveillance cameras in urban settings has accentuated the demand for advanced automated action detection methods. For instance, New York City alone has around 15,000 roadside cameras [19], generating hours of data. Most large cities in the world has similar kind of surveillance camera statistics. Manually monitoring these cameras are expensive and time-consuming. This approach becomes unsustainable for large-scale systems due to its inability to scale effectively.

By employing advanced methods to automatically detect and localize actions, the dependence on manual monitoring can be significantly diminished. This not only augments operational efficiency but also addresses concerns surrounding the privacy of citizens. If an automated method processes surveillance videos, the individual privacy rights can be

A dog dancing A cat skateboarding

Figure 1.2: Two examples of unusual actions. Humans can leverage previously learned concepts such as dancing, skateboarding to identify these actions. Second half of this thesis focuses on developing methods that demonstrate similar kind of compositional capabilities as humans. The images are taken from the UnRel dataset [2].

protected more effectively.

Consumer Applications: In the broader consumer context, automated action detection methods are extremely important. A quick example is the web content moderation. For filtering out contents based on the actions in the images and videos, we need advanced methods. These methods will help intuitions such as schools, corporate organizations to remove undesired web-contents from their systems. Moreover, these methods can identify and block malicious content websites.

Additionally, advanced action detection methods are crucial for the development of autonomous cars. In the paradigm of self-driving vehicles, the effective use of action detection methods hold the promise of enhanced road safety and optimal traffic management. By discerning different actions – be it pedestrians crossing the street or cyclists navigating through intersections – autonomous vehicles can anticipate and adapt to dynamic scenarios with a level of nuanced awareness that is unprecedented. This usage of action detection methods constitutes a paradigm shift that has the potential to save lives, mitigate accidents, and fundamentally redefine transportation systems.

1.2 Challenges and Contributions

1.2.1 Atomic Action Detection Models

Among the different types of atomic actions, those involving humans hold paramount importance. Consequently, the initial portion of this thesis is dedicated to the development of methods that can effectively detect human-object interactions (HOIs) within images. Defined as atomic actions where one participating entity is a human, HOIs serve as foundational elements for a range of other computer vision applications, including scene understanding [20–22] and visual question answering [23, 24].

The existing methodologies in this field typically adhere to a two-stage sequential process. Initially, off-the-shelf object detectors [25] are deployed to detect all humans and objects present in the image. Following this, the second stage involves generating predictions for interaction classes corresponding to each unique pairing of a human and an object.

While this sequential methodology has demonstrated strong performance across standard benchmark datasets, it is not without its challenges. Specifically, there are two primary issues: 1) The exhaustive pairing of humans and objects leads to significant difficulties in filtering out non-interacting human-object pairs; and 2) Identifying the salient spatial regions within images that effectively highlight interactions between human-object pairs remains a complex task. We provide solutions to these challenges in Chapter 3 and Chapter 4.

In chapter 3, we utilize relative spatial configuration and structural relations between humans and objects to help filter out non-interacting human object pairs. Relative spatial configuration refers to the relative spatial layout between a human and an object. We use this configuration to refine or guide our visual features. This refinement helps to filter out non-interacting pairs. Additionally, we utilize a graph convolution network to

utilize structural connections between human-object pairs.

In chapter 4, we develop an attention [26] architecture to find salient spatial context for HOI detection. Here, spatial context refers to specific regions within an image that are crucial for identifying HOIs. Given the diversity and complexity of HOIs, it is impractical to develop heuristic-based approaches for discerning the relevant spatial contexts for each unique interaction. Moreover, the spatial context associated with a particular interaction often lacks a rigid, predefined structure. Therefore, we utilize a Transformer [26] based architecture to find out the salient spatial context.

Transformers, with their query-key based attention architecture, have proven highly effective in identifying correlations across different segments of input data. We utilize human-object pairs as queries in the Transformer architecture. These queries are subsequently guided by both the relative spatial configuration and the semantic category of the involved object. In chapter 2 we already established the utility of relative spatial configuration for guiding visual features. We enhance our guiding mechanism by integrating object semantics, defined here as the categorical identity of the object. These categories are pivotal in characterizing the nature of the interaction. For instance, when the object in a human-object pair is a football, the likelihood of the interaction to be eating is substantially reduced. As such, the combination of relative spatial configuration and object semantics offers an effective guidance mechanism. Our guided query based transformer architecture has demonstrated outstanding ability in accurately detecting HOIs by leveraging salient spatial contexts.

In summary our key contributions to advancing atomic action detection models are:

- VSGNet: A Visual Spatial Graph Network utilizing Graph Convolutional Network alongside relative spatial configurations for enhanced Human-Object Interaction (HOI) detection.

- GTNet: A Guided Transformer Network designed to identify salient spatial regions relevant to HOI detection, utilizing a Transformer-based architecture.

1.2.2 Compositional Models

Scene-Graph Generation: As previously discussed, a defining characteristic of action detection models is their capability to compose previously learned concepts to identify novel actions. To enhance compositional skills in action detection models, we choose scene-graph generation (SGG) as our primary focus. Unlike our earlier atomic action detection models, which are limited to identifying human-involved actions, SGG provides a more expansive scope. It creates a structured representation of a scene by predicting subject-object-relation triplets from the given data. In this context, atomic action detection can be considered a specialized subset of SGG, where the relationships are specifically interactions like eating, walking, and so on [12].

More specifically, our objective is to identify relationship triplets that fall outside a predefined set. Current methodologies struggle significantly in this aspect, primarily due to the interdependent learning of object and relationship features. This coupled learning leads existing methods to form associations between specific relationships and fixed object combinations, thereby inhibiting their ability to recognize novel relationship triplets.

To address the challenge of interdependent feature learning, chapter 5 introduces our innovative decoupled SGG model. At the core of this model is the separation of relationship and object feature learning, a design choice aimed at overcoming the limitations of existing methods. To achieve this, we employ a multi-branch architecture: one branch is tasked with relationship detection, while another focuses on object detection. Our innovative design ensures that each branch learns decomposed cues of different relationships

and objects. As a result, when encounter an unfamiliar unseen relationship triplet, our model can successfully detect it from the previously learned concepts.

In our model each branch follows a Transformer-based encoder-decoder architecture. We employ two separate sets of queries in our two decoders, each set tailored to learn generalized representations of relationships and objects. This is achieved by extracting information from feature maps that are uniquely encoded by our encoders.

Object-Attribute (O-A) Compositions: The attributes of objects participating in an action can serve as indicators for detecting the action, as noted by [27]. For example, consider a video where full-sized tomatoes are initially shown. If subsequent frames reveal the same tomatoes in a diced form, we can infer that a tomato-cutting action has occurred, even in the absence of explicit visual confirmation. Therefore, a robust action detection model needs to detect both objects and attributes. Understanding O-A compositions in a holistic manner plays a crucial role in this aspect. The formal study of predicting unseen O-A combinations is known as Compositional Zero-Shot Learning (CZSL) [13, 28, 29].

Despite significant progress in CZSL, present models fail to associate correct O-A in a cluttered scene. For instance, identifying a ripe, red apple amidst surrounding leaves on a tree proves challenging for existing algorithms, due to the overpowering influence of confounding elements. This in turn is due to the bias towards seen O-A composition during training time of existing algorithms.

To tackle the mentioned challenge, in chapter 6, we propose a novel model that uses weakly supervised localization to detect correct O-A composition. Our model follows a two step approach. In the first step, we developed a Localized Feature Extractor (LFE) that associates an object with its attributes by minimizing the interference from other attribute-related visual cues within the image. It's important to note that the standard

CZSL benchmark datasets lack localization information. To circumvent these limitations, we have developed a weakly supervised method specifically for localized feature extraction. In the second step, we utilize a composition classifier that leverages these localized and distinctive visual features to accurately predict O-A pairs.

In summary, with respect to the compositional models, our contributions are:

- DDS: A Decoupled Dynamic Scene-Graph Network that has two branches to learn decoupled features for relationship triplets prediction
- LOCL: Learning Object-Attribute (O-A) Composition using Localization – that generalizes compositional zero shot learning to objects in cluttered realistic settings.

1.3 Dissertation Organization

The organization of the thesis is as follows,

Chapter 2, **Visual Spatial Graph** describes the model with spatial refinement and graph convolution for HOI detection. It includes detail comparison with existing works along with qualitative results.

Chapter 3, **Guided Transformer** introduces the novel transformer architecture to find and leverage salient spatial context to detect HOIs. It provides deep analysis behind the success of the model by providing attention maps and quantitative analysis.

Chapter 4, **Decoupled Dynamic Scene-Graph Generation** gives detail account of the multi-branch encoder-decoder model for scene-graph generation. Additionally, it has strong evidences showing the effectiveness of decoupled model in detecting unseen relationship triplets.

Chapter 5, **Learning Object-Attribute Compositions Using Localization** introduces the two step approach for correctly associating objects and attributes.

Chapter 6, **Discussion** provides a comprehensive summary of the key contributions made in this thesis. Additionally, it offers preliminary guidance for future research endeavors.

Chapter 2

Visual Spatial Graph

2.1 Introduction

The task of human object interaction (HOI) detection in images involves identifying and localizing interactions between a human and an object. As previously noted, HOI detection focuses on a specific type of atomic action where one of the entities involved is a human. It can be considered a part of the task of visual scene understanding [20–22], visual question answering [23, 24, 30, 31], and activity recognition in videos [3, 32, 33]. This and the next chapter describe two developed robust HOI detection methods.

Many current methods in HOI detection, such as [34–36], share a common structure. They begin by extracting human and object features using an object detection framework. These features, along with additional data (like pose and relative geometric locations), are then exhaustively paired and input into a multi-branch deep neural network to identify the relationship between humans and objects. Even though such an approach achieves good results for detecting HOIs, exhaustive pairing treats interacting and non-interacting human-object pairs in a similar manner. However, for a robust HOI detection network, the emphasis should be on the interacting pairs. An effective approach to reduce the

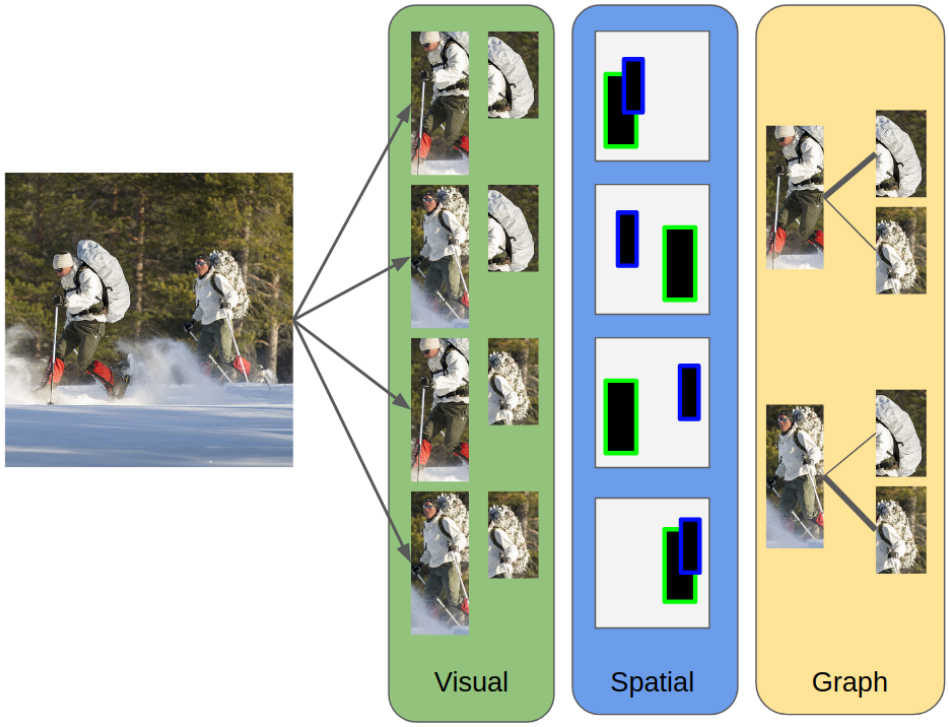

Figure 2.1: Visual, Spatial and Graph branches of our proposed VSGNet model. Visual branch analyzes humans/objects/context individually, Spatial branch uses spatial configurations of the pairs to refine visual features and the Graph branch utilizes the structural connections by Graph convolutions which uses interaction proposal scores as edge intensities between human-object nodes.

impact of non-interacting human-object pairs involves explicitly integrating interaction data or the spatial relationships between them. Interactions, such as a person on a skateboard or holding a bat, exhibit unique spatial and structural characteristics that can enhance HOI detection. To capitalize on this, we developed Visual-Spatial-Graph Network (VSGNet) architecture.

For utilizing spatial configurations, VSGNet uses a spatial attention branch that explicitly uses the spatial relations of the pairs to refine the visual features. Instead of modeling humans and objects individually, our module uses the spatial configurations of the pairs to extract attention weights which refine the visual features. Although a few past works [6, 35] have used these types of spatial configurations as features for classification

directly, these models do not combine the visual information with spatial information. These features are more useful for refining the visual features and providing an attention mechanism for modeling the interactions of the human-object pairs explicitly.

For modeling the interactions, an image can be defined as a graph. Nodes in this graph are the humans and objects, in which case the edges define the interactions. As the edges between nodes define interactions between pairs, our model utilizes the interaction proposal scores as the intensities of the edges in the graph. Interaction proposal scores are generated from the spatially refined visual features and they quantify if the human-object pair is interacting.

To summarize, the proposed VSGNet for HOI detection refines the visual features using spatial relations of humans and objects. This approach amplifies the visual features of spatially relevant pairs while damping the others. Additionally, this model uses graph convolutional networks to model the interactions between humans and objects. The resulting model consists of multiple specialized branches. We evaluate our model on V-COCO [1] and HICO-DET [6] datasets and demonstrate 4 mAP (8%) and 3 mAP (16%) improvement over the state of the art methods.

Technical Contributions:

- We propose a new spatial attention branch that leverages the spatial configuration of human-object pairs and refines the visual features such that spatially relevant human-object pairs are amplified.

- We use a graph convolutional branch which utilizes the structural connections between humans and objects. The interaction proposal score, generated from the spatially refined features, are used to define the edge intensities between human and object nodes.

- We implement a robust pipeline that contains Visual, Spatial and Graph based branches named VSGNet. This model achieves state-of-the-art results for HOI detection task on V-COCO and HICO-DET datasets.

2.2 Related Work

Object Detection: For detecting HOIs the first step is to detect humans and objects properly. With the recent object detection frameworks like RCNN [37], Faster RCNN [25], YOLO [38], Feature Pyramid Network [39] and SSD [40], models are able to detect multi scale objects robustly in images. Following this we utilize a pre-trained Faster-RCNN model in our network for detecting humans and objects. Additionally, we utilize the region proposal network idea from Faster-RCNN and extend it to interaction proposals which predict if an human-object pair is interacting.

Human Object Interaction: Activity recognition is a research area in computer vision that has received interest for a long time. There are different datasets like UCF-101 [41], Thumos [42] with a focus on detecting human actions in videos. But in these datasets, the goal is to detect one action in a short video which is not representative of real life scenarios. To extend human activity recognition in images Gupta et al. [1] introduced V-COCO dataset and Chao et al. [6] introduces HICO-DET dataset. These datasets are different from the previous datasets as they require models to explicitly detect humans, objects and their interactions. This extends the task to include detection of human activities while localizing the humans and the objects.

For the HOI detection task, Gkioxari et al. [36] proposed a human-centric approach arguing that human appearance provides strong cues in both detecting the action and localizing the object. This method does not consider interactions where the object is far away from the human. Qi et al. [43] proposed a graph based network which depends

on detecting an adjacency matrix between various nodes(here, nodes are humans and objects) but does not utilize any spatial relation cues between pairs. Kolesnikov et al. [44] incorporates HOI detection with object detection by individually analyzing humans and objects without considering the spatial relationship between the pairs.

Gao et al. [35] proposed an attention network based on the previous work of [26]. They derived an attention map from the human and object features over the whole convolutional feature map. Although they used a binary spatial map similar to [6], they use the spatial map to extract features and concatenate them with human visual features. As these are two completely different features defining separate things, concatenation does not enforce spatial configurations as much as an attention mechanism. To address this in our network we use the spatial features as attention maps which refines our visual features.

Li et al. [34] integrated pose estimation with the iCAN [35] and predicted the interaction probabilities between a human and object pair. These methods however, do not explicitly leverage the interaction probabilities to detect the relational structure between the human and object pairs. Our VSGNet addresses this by utilizing a graph network for learning interactions and achieves better results without using poses which shows VSGNet can benefit from pose estimation as well.

2.3 Proposed Method

This section introduces our proposed VSGNet for detecting human-object interactions(HOI). From each given image, the task is to detect bounding boxes for the humans, objects and correctly label the interactions between them. Each human-object pair can have multiple interaction labels and each scene can include multiple humans and objects in them. We streamline the task by employing a pre-trained object detector to identify

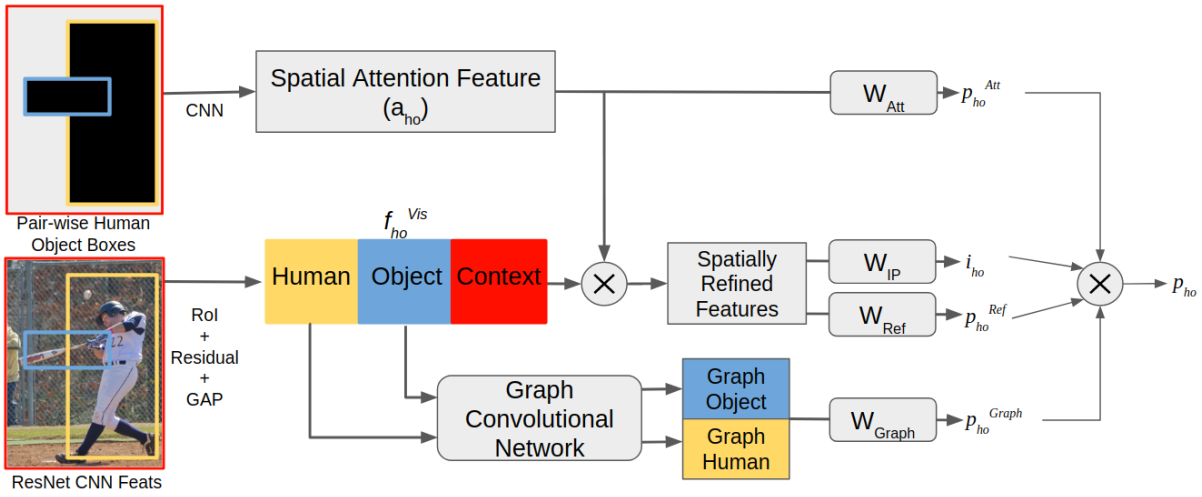

Figure 2.2: Model Architecture. Rounded rectangles are operations, sharp rectangles are extracted features and $\otimes$ is element-wise multiplication. The model consists of three main branches. Visual branch extracts human, object and context features. Spatial Attention branch refines the visual features by utilizing the spatial configuration of the human-object pair. Graph Convolutional branch extracts interaction features by considering humans/objects as nodes and their interactions as edges. Action class probabilities from each branch and the interaction proposal score are multiplied together to aggregate the final prediction. These operations are repeated for every human-object pair.

humans and objects within an image.

Detecting the interactions between human-object pairs is a challenging task. Simple methods such as extracting features from human and object locations individually and analyzing them are ineffective as these methods ignore the contextual information of the surroundings and spatial relations of the human-object pair. Extensions such as using union boxes to model the spatial relations/context also fall short as they don't explicitly model the interactions. To address these issues, we propose a multi-branch network with specialized branches. The proposed VSGNet consists of the Visual Branch (Section 2.3.2) which extracts visual features from human, object and surrounding context individually; the Spatial Attention Branch (Section 2.3.3) which models spatial relations between the human-object pair; and the Graph Convolutional Branch (Section 2.3.4) which consid-

ers the scene as a graph with humans and objects as nodes and models the structural interactions. The proposed model architecture with the branches is shown in Fig.2.2.

2.3.1 Overview

The inputs to our model is image features $\mathbf{F}$ from a backbone CNN (e.g. ResNet-152 [45]) and bounding boxes x_h for human $h \in [1, H]$ and x_o for object $o \in [1, O]$. H and O represents the total number of humans and objects in the scene respectively. Bounding boxes are obtained from a pre-trained object detector. We define the objective of this model as:

- Detect if human h is interacting with object o with an interaction proposal score $i_{h,o}$.
- Predict the action class probability vector $\mathbf{p}_{h,o}$ of size A where A is the number of classes.

2.3.2 Visual Branch

This branch focuses on extracting visual features for the human-object pairs. Following the object detection methods, we use region of interest (RoI) pooling on the human/object regions to extract features. This operation is followed by a residual block (Res) [45] and global average pooling(GAP) operations to extract the visual feature vectors for objects and humans.

$$\mathbf{f}_h = GAP(Res_h(RoI(\mathbf{F}, x_h))) \tag{2.1}$$

$$\mathbf{f}_o = GAP(Res_o(RoI(\mathbf{F}, x_o))) \tag{2.2}$$

where $Res_{\{\}}$ represents residual blocks, $\mathbf{f}_h$ and $\mathbf{f}_o$ are visual feature vectors of sizes R. This operation is repeated for each human h and object o.

Context plays an important role in detecting HOI. Surrounding objects, background and other humans can help detecting the interactions. We include the context in our network by extracting features from the entire input image followed by a residual block and global average pooling.

$$\mathbf{f}_C = GAP(Res_C(\mathbf{F})) \tag{2.3}$$

where $\mathbf{f}_C$ is a feature vector of size R.

Finally, this branch combines all the visual feature vectors by concatenating them and projecting it by a fully connected layer.

$$\mathbf{f}_{ho}^{Vis} = \mathbf{W}_{vis}(\mathbf{f}_h \oplus \mathbf{f}_o \oplus \mathbf{f}_C) \tag{2.4}$$

where $\oplus$ is the concatenation operation, $\mathbf{W}_{\{\}}$ is the projection matrix, $\mathbf{f}_{ho}^{Vis}$ is the combined visual feature vector of dimension D which represents the human-object pair ho.

The feature $\mathbf{f}_{ho}^{Vis}$ can be used directly for classifying actions. We implement this as a base model for comparisons.

2.3.3 Spatial Attention Branch

This branch focuses on learning the spatial interaction patterns between humans and objects. The main task is to generate attention features which are used to refine the visual features by amplifying the pairs with high spatial correlation. This branch is visualized in Fig.2.3.

Given the human bounding box x_h and object bounding box x_o, we generate two

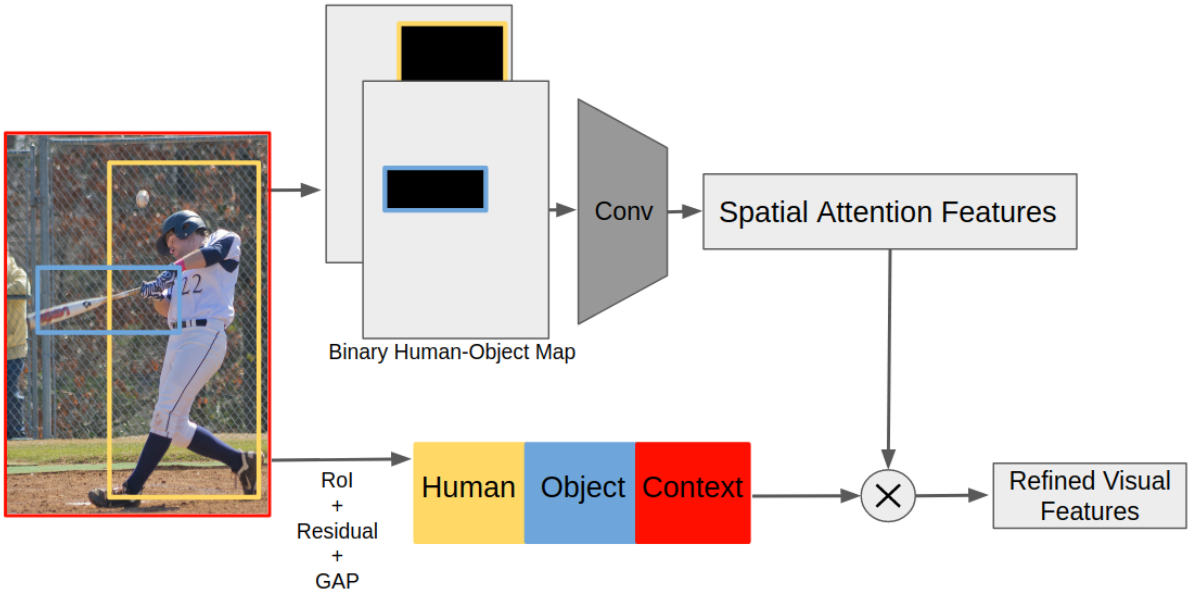

Figure 2.3: Spatial Attention Branch. Initially human, object and context visual features are extracted from the image using RoI pooling. Using binary maps of human and object locations, spatial attention features are extracted using convolutions. These attention features encode the spatial configuration of the human-object pair. Attention features are used to refine the visual features by amplifying the pairs with high spatial correlation.

binary maps. These binary maps have zeros everywhere except in locations defined by human and object box coordinates x_h and x_o for each map respectively. This generates a 2-channel binary spatial configuration map $\mathbf{B}_{ho}$.

Similar to [6, 35], we use 2 layers of convolutions to analyze the binary spatial configuration map. This is followed by a GAP operation and a fully connected layer.

$$\mathbf{a}_{ho} = \mathbf{W}_{Spat}(GAP(Conv(\mathbf{B}_{ho}))) \tag{2.5}$$

where $\mathbf{a}_{ho}$ is an attention feature vector of size D and represents the spatial configuration of the human-object pair ho. As the objects and humans are defined in different channels, using convolutions on the binary spatial configuration maps $\mathbf{B}_{ho}$ allows the model to learn the possible spatial relations between humans and objects.

Since $\mathbf{a}_{ho}$ encodes the spatial configuration, it can be used directly to classify the HOIs as in [6]. We keep this classification as an auxiliary prediction but mainly use $\mathbf{a}_{ho}$ as an attention mechanism for refining visual features. Auxiliary predictions can be defined as:

$$\mathbf{p}_{ho}^{Att} = \sigma(\mathbf{W}_{Att}(\mathbf{a}_{ho})) \tag{2.6}$$

where $\mathbf{p}_{ho}^{Att}$ is the action class probabilities of size A and σ is the sigmoid function.

The attention vector $\mathbf{a}_{ho}$ and the visual feature vector $\mathbf{f}_{ho}^{Vis}$ are set to be the same size D. This allows us to multiply these two vectors together in order to refine the visual features with spatial configuration. We use $\mathbf{a}_{ho}$ as an attention function and multiply $\mathbf{a}_{ho}$ and $\mathbf{f}_{ho}^{Vis}$ elementwise.

$$\mathbf{f}_{ho}^{Ref} = \mathbf{a}_{ho} \otimes \mathbf{f}_{ho}^{Vis} \tag{2.7}$$

where $\otimes$ is element-wise multiplication and f_{ho}^{Ref} is the spatially refined feature vector of size D.

The refined feature vector is then used to predict the interaction proposal score of human-object pair ho and to predict the action class probabilities.

$$i_{ho} = \sigma(\mathbf{W}_{IP}(\mathbf{f}_{ho}^{Ref})) \tag{2.8}$$

$$\mathbf{p}_{ho}^{Ref} = \sigma(\mathbf{W}_{Ref}(\mathbf{f}_{ho}^{Ref})) \tag{2.9}$$

where i_{ho} is the interaction proposal probability of size 1 and $\mathbf{p}_{ho}^{Ref}$ is the action class probabilities of size A.

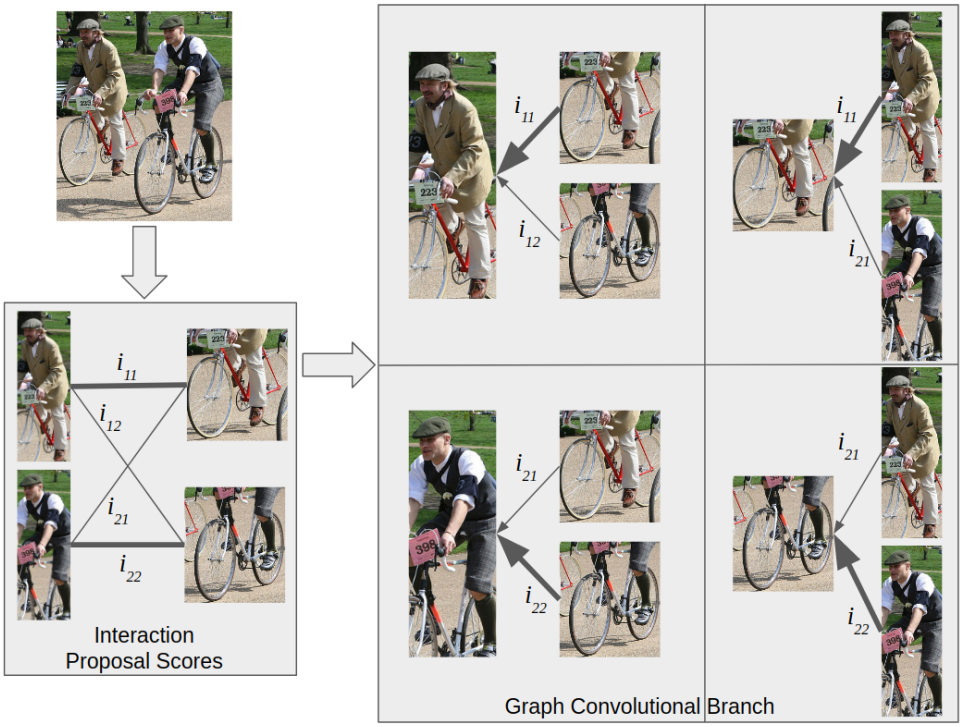

Figure 2.4: Graph Convolutional Branch. This model learns the structural connections between humans and objects. For this task, we define the humans and objects as nodes and only connecting edges between human-object pairs. Instead of using visual similarity as the edge adjacency, we propose to use the interaction proposal scores. This allows the edges to utilize the interactions between human-object pairs and generates better features.

2.3.4 Graph Convolutional Interaction Branch

This branch uses a graph convolutional network to generate effective features for humans and objects. Graph convolutional networks extract features that model the structural relations between nodes. This is done by traversing and updating the nodes in the graph using their edges. In this setting, we propose to use humans and objects as nodes and their relations as edges.

Instead of having a fully connected graph, we connect each human with every object and each object with every human. However, without this simplification, proposed model can also be extended to fully connected settings.

Given the visual features $\mathbf{f}_h$, $\mathbf{f}_o$ and connecting the edges between humans and objects, graph features $\mathbf{f}'_h$ and $\mathbf{f}'_o$ are defined as follows:

$$\mathbf{f}'_h = \mathbf{f}_h + \sum_{o=1}^{O} \alpha_{ho} \mathbf{W}_{oh}(\mathbf{f}_o) \tag{2.10}$$

$$\mathbf{f}'_o = \mathbf{f}_o + \sum_{h=1}^{H} \alpha_{oh} \mathbf{W}_{ho}(\mathbf{f}_h) \tag{2.11}$$

where α_{ho} defines the adjacency between h and o and $\mathbf{W}_{oh}$, $\mathbf{W}_{ho}$ are mapping functions which project the object features to human feature space and vice versa. Previous works [43,46] defined the adjacency as visual similarity. In our task, however, adjacency defines interactions between nodes of visually unsimilar things which are human and object. Following this idea, we define adjacency values between h and o pair as:

$$\alpha_{ho} = \alpha_{oh} = i_{ho} \tag{2.12}$$

where i_{ho} is the interaction proposal score which are generated from the spatially refined visual features and measure the interactions of the human-object pair. Pairing up the graph features, classification predictions are calculated as:

$$\mathbf{p}_{ho}^{Graph} = \sigma(\mathbf{W}_{graph}(f'_h \oplus f'_o)) \tag{2.13}$$

where $\oplus$ is concatenation operation and $\mathbf{p}_{ho}^{Graph}$ is the action class probabilities of size A.

The graph convolutional branch is visualized in Figure 2.4. This concludes all of the outputs of the proposed network. Finally we combine the action predictions and the interaction proposal scores by multiplying the probabilities similar to previous works [34–36].

$$\mathbf{p}_{ho} = \mathbf{p}_{ho}^{Att} \times \mathbf{p}_{ho}^{Ref} \times \mathbf{p}_{ho}^{Graph} \times i_{ho} \tag{2.14}$$

where P_{ho} is the final prediction vector of size A.

2.4 Experiments

We first introduce the datasets and our evaluation metrics along with our implementation details and then perform extensive quantitative and qualitative analysis on our model and show the improvements over the existing methods.

2.4.1 Datasets and Evaluation Metrics

Datasets: To evaluate our model's performance, we use the V-COCO [1] and HICO-DET [6] datasets.

V-COCO is derived from COCO [47] dataset. It has 10,346 images. 2533 images are for training, 2867 images are for validating and 4946 images are for testing. The training and validation set images are from COCO training set and the test images are from the COCO validation set. Each person in the images are annotated with a label indicating one of the 29 actions. If an object in the image is related to that action then the object is also annotated. Among these 29 actions, four of them has no object pair and one of them(point) has only 21 samples. Following the previous HOI detection works, we are not going to report our performance in these classes. We report our performance for the rest of the 24 classes.

HICO-DET is a large dataset for detecting HOIs with 38118 training and 9658 testing images. HICO-DET annotates the images for 600 human-object interactions. Following the previous works, in HICO-DET we report our performance in Full, Rare and Non-Rare

Categories. These categories are based on the number of training samples [6].

Metrics: Following [1] we evaluate our performance on two types of average precision(AP) metrics: Scenario 1 and Scenario 2. During AP calculation in both metrics, a prediction for a human-object pair is considered correct (1) if the human and object bounding boxes have an IoU greater than 0.5 with the ground-truth boxes and (2) the interaction class label of the prediction for the pair is correct. For the cases when there is no object(human only), in Scenario 1 a prediction is correct if the corresponding bounding box for the object is empty and in Scenario 2 the bounding box of the object is not considered. This makes Scenario 1 much harsher than Scenario 2 [1]. In HICO-DET our evaluation metrics is similar to the Scenario 1 case of V-COCO.

2.4.2 Implementation Details

Resnet-152 [45] network is used as the backbone feature extractor. We extract the input feature map before the last residual block of Resnet-152. This serves as the input to the rest of the network. We extract 10×10 feature maps for all the humans and objects from the input feature map by region of interest pooling [48]. Extracted RoIs and input feature map(context) pass through a residual block followed by a global average pooling similar to [35]. After these steps, we obtain three feature vectors of size $R = 1024$ for human, object and context. These are fed to the rest of the network. For the spatial attention branch we have used $64 \times 64 \times 2$ binary inputs. Before the element wise multiplication with the attention vector in the spatial attention branch, we project all our input feature vectors to a $D = 512$ dimensional space followed by a ReLU. In our final classification layer for all the branches, we have one linear layer.

For training the network, we utilize off-the-shelf Faster-RCNN [25] to generate human and object bounding boxes. We have filtered the detected bounding boxes by setting 0.6

confidence threshold for human bounding boxes and 0.3 for object bounding boxes. The threshold values are chosen experimentally. Following [36] we did not fine tune the backbone CNN Resnet-152 [45] and Faster-RCNN during our training process. Faster-RCNN was trained on the COCO [47] training set and did not see any image from V-COCO testing sets. Unlike previous works [34, 35], we do not use ground truth boxes during training as object proposals. As our object detector is robust, we directly use the bounding boxes generated from the detector which generates sufficient amount of positive and negative boxes.

Initially, we trained the model on the training set of V-COCO while validating with the validation set. Then we train the model in both training and validation set like [36]. Our initial learning rate is set to 0.01 with a batch size of 8. As optimizer, Stochastic Gradient Descent(SGD) has been used with a weight decay of 0.0001 and a momentum of 0.9. To reduce the training time we increased our learning rate to 0.01 for all the layers except for the spatial attention branch between epoch 9 to epoch 21. We trained the whole model for 50 epochs.

For HICO-DET we use the same hyper-parameters from V-COCO. We train the network individually for 20 epochs in HICO-DET training set without any initialization from the V-COCO model.

During inference, we multiply all the prediction outputs from the different branches of our network as in 2.14. Additionally, we multiply the final prediction output with the detection confidences of the human and object from the object detector. To differentiate between high and low quality detection scores we have adopted Low grade Instance Suppressive Function (LIS) [34]. We additionally remove the incompatible interaction-object pairs by using a post processing similar to iCAN [35] (e.g. if the object is not phone then the interaction can not be talk on the phone).

While making inference most of the existing [34–36] models multiply all the outputs

V-COCO	mAP(Sc 1)	mAP(Sc 2)
InteractNet [36]	40.0	47.98
Kolesnikov et al. [44]	41.0	-
GPNN [43]	44.0	-
iCAN [35]	45.3	52.4
Li et al. [34]	47.8	-
VSGNet	**51.8**	**57.0**

Table 2.1: Comparison of results in V-COCO [1] test set on Scenario 1 and Scenario 2. Our method outperforms the closest method by 8%. For actor only classes (no object), scenario 1 requires the model to detect it specifically as no object, whereas scenario 2 ignores if there is an object assigned to that prediction. Some of these methods did not provide results for scenario 2.

from different modules but these modules are optimized separately while training. Following [49] we have used a single cross entropy loss function for each action class to optimize the network. One thing to note is that as in Eq. 2.14, interaction proposal score is also multiplied in these predictions and included in predictions for every class. This allows the proposal score to quantify if there are interactions between the human-object pair regardless of the class of that interaction. Our experiments show that combining all the predictions and using a single loss function improves the performance.

2.4.3 Comparisons with the State of the Art

We compare our model's performance with five recent state of the art methods [34–36,43,44] in both of the datasets. We report mean Average Precision (mAP) score in the settings provided by [1] and [6].

Table 2.1 shows that our method outperforms all the existing models and achieves an improvement of 4 mAP in scenario 1 for V-COCO dataset. We also reported our performance in scenario 2 which outperforms all the available existing methods who reported their results in that scenario.

Table 2.2 shows the results compared to other methods in HICO-DET and our model

HICO-DET (mAP)	Full	Rare	Non-Rare
HO-RCNN [6]	7.81	5.37	8.54
InteractNet [36]	9.94	7.16	10.77
GPNN [43]	13.11	9.34	9.34
iCAN [35]	14.84	10.45	16.15
Li et al. [34]	17.03	13.42	18.11
VSGNet	**19.80**	**16.05**	**20.91**

Table 2.2: Comparison of results in HICO-DET [1] test set. VSGNet outperforms the closest method by 16%.

achieves the best results among the previous works.

In terms of performance, the closest work to our results is Li et al. [34]. This method is built on top of the exisiting work iCAN [35] by adding an interaction proposal network and utilizing person poses. Addition of interaction proposal and person poses improve ~ 2 mAP in V-COCO and ~ 3 mAP in HICO-DET on top of iCAN with a computational cost of calculating the poses for each human. Our model achieves better results than Li et al. [34] without the pose extraction.

In Table 2.3 we report per-class performances compare with the existing methods which reported per-class APs for V-COCO. Our proposed VSGNet achieves better performance in majority of the classes compared to the other methods. Additionally, per-class performances show that some of the action classes perform badly due to the failure of object detectors (e.g. eat instruments which usually have small objects and commonly become occluded in the images). As our main task is to detect HOIs, we did not fine-tune the existing object detectors according to our needs which can also possibly handle these cases.

Analysis of Backbone CNNs: In addition to all Resnet models [45], we implement our model with various common CNNs used in image analysis. Table 2.5 shows the results of VSGNet implemented with these various backbone CNNs in V-COCO with Resnet152

HOI Class	InteractNet [36]	iCAN [35]	VSGNet
hold-obj	26.38	29.06	**48.27**
sit-instr	19.88	26.04	**29.9**
ride-instr	55.23	61.9	**70.84**
look-obj	20.2	26.49	**42.78**
hit-instr	62.32	74.11	**76.08**
hit-obj	43.32	46.13	**48.6**
eat-obj	32.37	37.73	**38.3**
eat-instr	1.97	**8.26**	6.3
jump-instr	45.14	51.45	**52.66**
lay-instr	20.99	**22.4**	21.66
talk_on_phone	31.77	52.81	**62.23**
carry-obj	33.11	32.02	**39.09**
throw-obj	40.44	40.62	**45.12**
catch-obj	42.52	**47.61**	44.84
cut-instr	22.97	37.18	**46.78**
cut-obj	36.4	34.76	**36.58**
work_on_comp	57.26	56.29	**64.6**
ski-instr	36.47	41.69	**50.59**
surf-instr	65.59	77.15	**82.22**
skateboard-instr	75.51	79.35	**87.8**
drink-instr	33.81	32.19	**54.41**
kick-obj	69.44	66.89	**69.85**
read-obj	23.85	30.74	**42.83**
snowboard-instr	63.85	74.35	**79.9**
Average	40.0	45.3	**51.76**

Table 2.3: Per class AP comparisons to the existing methods in V-COCO Scenario 1. Our method demonstrates superior performance in majority of the classes. We only compared to the methods which have reported the per class AP values. Obj refers object cases where instr refers to instrument [1].

Branches	mAP(Sc 1)	mAP(Sc 2)
Visual (Base)	47.3	52.15
Visual+Graph	48.19	53.12
Visual+Spatial	50.33	55.32
Visual+Spatial+Graph(VSG)	51.76	57.03

Table 2.4: Analysis of the branches. Our base model consists of only the Visual branch. We add the graph branch and the spatial attention branch to this base model separately to analyze their performances. Individually, both branches improve the performance upon the base model. Visual+Spatial model beats the state of the art results and all three branches combined adds another 1.5 mAP.

Branch	mAP (Scenario 1)
VGG-19 [50]	48.37
InceptionV3 [51]	49.39
SqueezeNet [52]	43.4
Resnet34 [45]	50.88
Resnet50 [45]	51.01
Resnet101 [45]	50.01
Resnet152 [45]	**51.76**

Table 2.5: Effects of the backbone CNN on V-COCO dataset. VSGNet is implemented using various common backbone CNNs. Resnet-152 model with VSGNet achieves the best performance.

performing the best.

2.4.4 Ablation Studies

Analysis of Individual Branches: Our overall architecture consists of three main branches. To evaluate how these branches are affecting our overall performance, we evaluate these branches individually in the V-COCO [1] test set. Our evaluation method and metrics are same as Table 2.1. We consider the base model as the Visual branch without the spatial attention or the graph convolutions. In this setting, interaction proposal score I_{ho} and the class probabilities P_{ho} are predicted from the visual features f_{ho}^{Vis} directly.

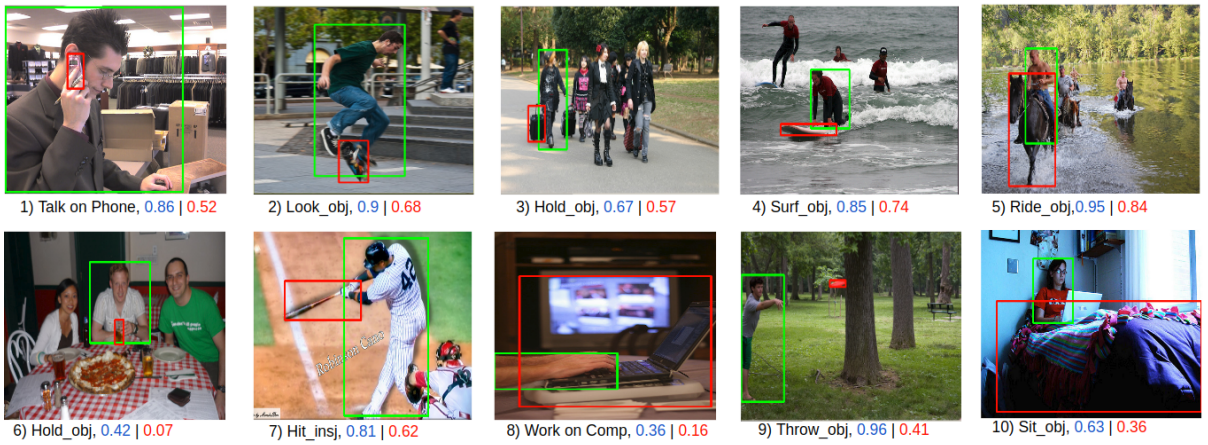

Figure 2.5: Qualitative results. Red values show the confidences for the base model (Visual only) and blue values are the results for the VSGNet. The prediction results and the correct action labels are shown for the human-object pair visualized with the bounding boxes.

We have added the graph network and the spatial network with our base model individually to evaluate each of the branch's performance separately. The results are shown in Table 2.4. With addition of the individual branches, model performance has improves gradually. Visual+Spatial branch achieves state of the art results by itself without the Graph branch. Addition of the graph branch adds additional 1.5 mAP and a total of 4mAP over the state of the art.

An important detail is that the graph branch directly depends on the quality of the interaction proposal score i_{ho} as it is used to determine the edge interactions. Without the spatial attention, visual features generate inferior i_{ho} which affects the graph branch. This is the reason that addition of Graph to Visual branch only adds 0.9 mAP whereas addition of Graph to Visual+Spat makes a larger improvement and adds 1.5 mAP.

Spatial attention branch improves the result by 3 mAP when added to the visual branch. This demonstrates the importance of the spatial reasoning and refining the visual features. Graph and Spatial attention combined improves the performance by about 4.5 mAP over the base model.

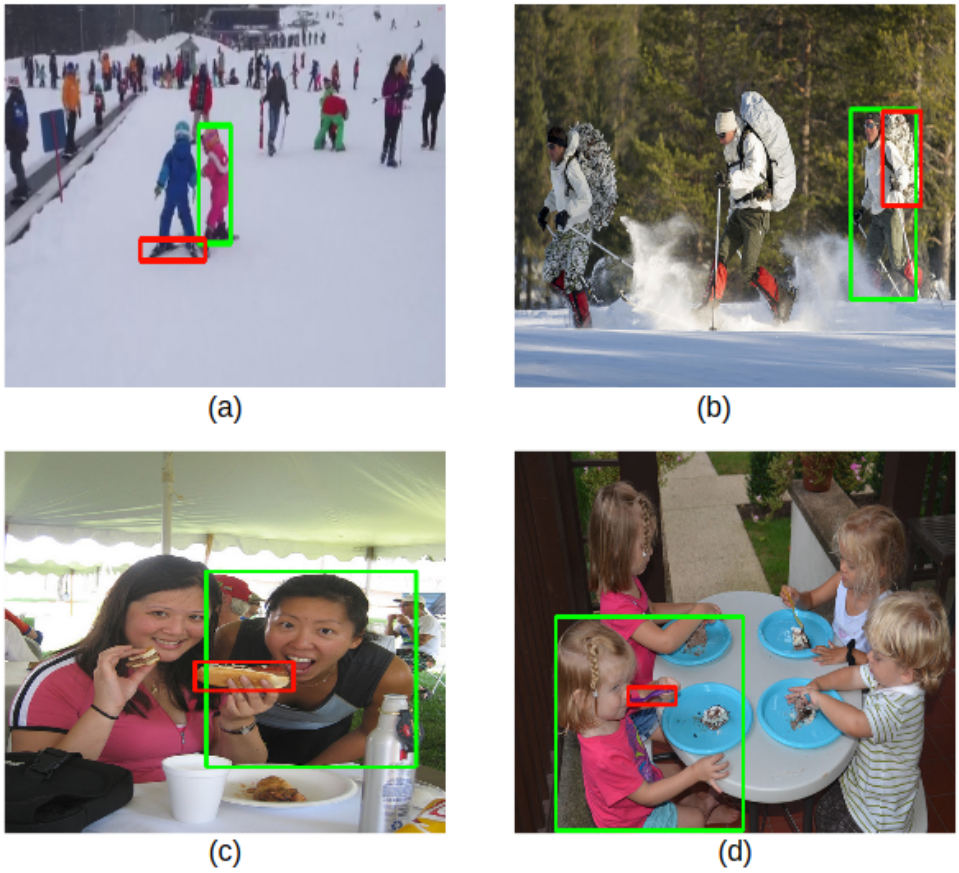

Figure 2.6: Few of the cases where our VSGNet's prediction is wrong due to the confusing visual and spatial cue from the images. (a) Human-object pair is detected to be interacting but they are not, (b) Label mismatch (hold vs carry), (c) confusing scene and (d) object detector fails to detect the fork.

Qualitative Results: Figure 2.5 shows qualitative results and compares the VSGNet with the base model (Visual only). The interaction prediction probabilities for the correct action is visualized. The images show the variance in object sizes, human sizes and different interaction classes. VSGNet performs better than the base model. Even in the cases when the object is not entirely visible (image 9) or the interaction is very subtle (image 2) VSGNet performs well and improves upon the base model.

Failure Cases: When the visual or spatial cues are confusing, the model can fail to predict the correctly. In Figure 2.6 a few failure cases are shown. Our method can fail if

the spatial configuration is confusing (a), confusing ground truth labels (hold and carry in (b)), multiple humans interacting with the same object with similar spatial configuration (c), the object detector fails to detect the objects of interest (d).

2.5 Discussions

2.5.1 Comparisons with Related Works

We compare VSGNet with methods using spatial relations [6, 34, 35], attention [35] and graph convolutions [43, 46].

There have been previous works which use spatial relation maps such [6, 34, 35]. These methods have either used the spatial relation maps directly for classification [6] or concatenated the spatial relation features to their visual features [34, 35]. Directly using them for classification ignores the visual features which in turn only learns relationship between the interaction label and spatial configuration. Concatenation of visual and spatial relations is also inferior to our method. As these are two completely different features defining separate things, concatenation does not enforce spatial configurations as much as an attention mechanism. In contrast, we use the spatial relations to extract attention features which are then used to alter the visual features. This is more effective as it models the relations between the visual feature channels and spatial configuration due to the element-wise multiplication.

Attention models also have been used on HOI task. iCAN [35] model uses an attention model inspired from [26] and models the attention of the human or object region with the whole input scene individually. However, this approach does not consider the relation between the pairs and they only include the spatial configuration at the end. Our approach uses the spatial configuration directly to alter the visual features of the

pairs which amplifies connected ones and dampens irrelevant ones at feature level.

Graph convolutions [43, 46] have been effective in various tasks. These tasks learn or use visual similarity as adjacency values between nodes and extract graph features. However, for our task, interaction proposal scores already defines the adjacencies between human-object node pairs and are used as edge intensities. This approach effectively extracts graph features by traversing relevant object nodes for the humans and relevant human nodes for objects.

2.5.2 Summary

We presented a novel human-object interaction detection model VSGNet which utilizes Visual, Spatial and Graph branches. VSGNet generates spatial attention features which alter the visual features and uses graph convolutions to model the interactions between pairs. The altered visual features generate interaction proposal scores which are used as edge intensities between human-object node pairs. We demonstrated with thorough experimentation that VSGNet improves the performance and outperforms the state-of-the-art.

Chapter 3

Guided Transformer

3.1 Introduction

One of the primary objectives of computer vision systems is to interpret human activities in images and videos. A critical initial step in this endeavor involves the accurate detection of human-object interactions (HOIs). This process entails identifying the interactions between humans and objects and pinpointing their specific locations within the image. In Chapter 2, VSGNet was introduced as a tool for HOI detection, featuring a refinement mechanism designed to eliminate non-interacting HOI pairs. This chapter shifts focus to the identification of key spatial regions critical for effective HOI detection.

In basic detection frameworks for existing HOI detection networks, human and object features are usually extracted with an object detection network. Then, interactions are predicted from the extracted features. To strengthen these features, one could use additional features such as relative spatial configurations [35,53], human pose estimation [54], or more accurate segmentation masks [55].

However, to fully utilize the power of these additional features, the local spatial

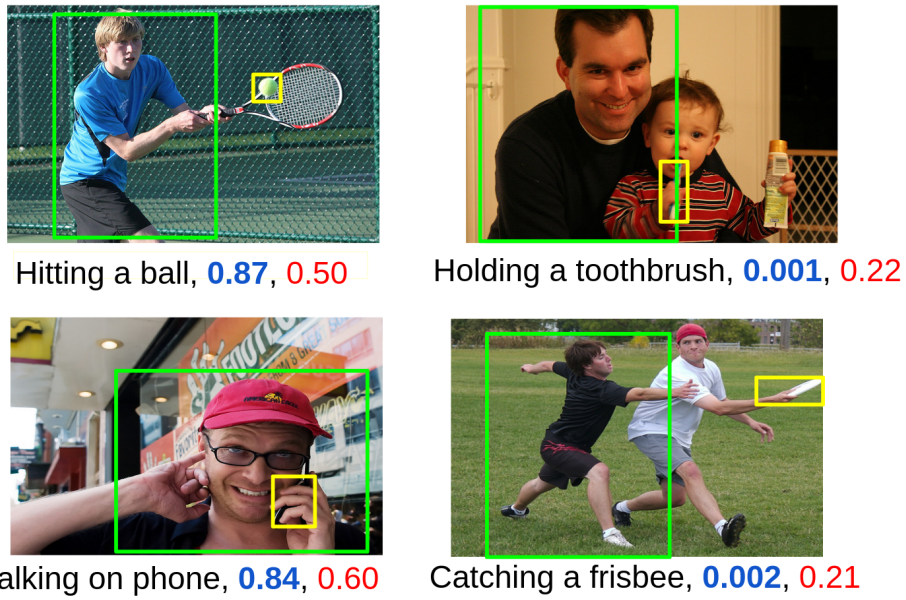

Figure 3.1: GTNet's performance with and without its guidance mechanism. Blue indicates GTNet's predictions with the guidance mechanism; red indicates without. Green bounding boxes indicate the human under consideration; yellow boxes indicate the object. Left column: Simple Scenarios (higher confidence score is better). Right column: Potentially Confusing Scenarios (lower confidence score is better). With the guidance mechanism, it is easier find salient spatial context for detecting different types of HOIs.

context needs to be leveraged more effectively. In computer vision tasks, context often refers directly to the background and surroundings of the objects or people of interest. In the following, we use "context" to refer to spatial regions localizing the human-object interactions. Few current works [4, 35, 56, 57] have tried to find relevant spatial contexts by having separate attention mechanisms for humans and objects but do not leverage any additional features in the attention framework.

Towards this, we developed a unique approach to identify spatial contextual information for detecting HOIs by leveraging spatial configurations (i.e. relative spatial layout between a human and an object) and object semantics (i.e. category of objects represented by word embeddings). This method has proven to be very effective, as can be seen in Figure 3.1, where our proposed Guidance Module improves performance over our

model without the guidance mechanism. We refer to our proposed network as Guided Transformer Network or GTNet.

GTNet takes inspiration from Natural Language Processing (NLP) where the self attention [58] based Transformer architecture [26] has shown significant success in identifying contextual information. To adapt the Transformer architecture for detecting HOIs, GTNet concatenates pairwise human and object features and uses them as queries to the attention mechanism. However, detection of HOIs often depend upon relative spatial configuration (shown in previous chapter) and the type of the objects [59]. Therefore, we combine relative spatial configurations with object semantics to guide the queries throughout the network. With the help of our proposed guided attention, we successfully encode the spatial contextual information in these queries.

The proposed GTNet architecture is shown in Figure 3.2. From the input image and the bounding boxes of the humans and the objects present in it, our baseline module (Section 3.3.1) extracts visual features. We guide these visual features by pertinent spatial configurations and object semantics in our Guidance Module (Section 3.3.2). On top of these guided visual features, we develop the TX module, which enriches the visual features with relevant contextual information using attention (Section 3.3.3). Finally, we propose an early fusion strategy to make our final predictions (Section 3.3.4). We evaluated our network's performance on V-COCO [1] and HICO-DET [6] and achieve state of the art results on both of the datasets. Our contributions in this chapter can be summarized as follows:

- We leverage pairwise spatial contextual information via a novel end to end guided self attention network for detecting HOIs. See Section 3.3.3.
- We design a guidance mechanism that combines relative spatial configurations and object semantics to guide our attention mechanism. See section 3.3.2.

- GTNet achieves state of the art results for the HOI detection task on both V-COCO and HICO-DET datasets. See section 3.4.3.

3.2 Related Works

Human-Object Interaction: With the introduction of benchmark datasets like V-COCO [1] and HICO-DET [6], there is a plethora of works detecting human-object interactions [4, 34, 35, 43, 53–57, 59–64]. Earlier works [36] in this area focus on the visual features of humans and objects. Many subsequent works [4, 35, 56] try to find spatial context for interactions on top of these visual features. Gao et al. [35] present a self-attention mechanism around individual humans and objects. T. Wang et al. [56] leverage this attention mechanism with a squeeze and excitation block [65] from object detection. Recently, graph-based architectures where humans and objects are considered nodes attempt to understand spatial context [4, 57] for the structural relations. ConsNet [59] has leveraged word embeddings of the objects in this graph structure. Additionally, Hou et al. [66, 67] have utilized object affordance to detect HOIs. A few recent works [68–71] have developed a one stage pipeline to detect HOIs rather than the two-stage (object detection + HOI detection) approach. We only compare our method with two stage HOI detection networks. However, none of the existing attention-based works try to utilize additional features like relative spatial configurations or object semantics to find richer spatial contextual features in the attention framework. In this aspect, GTNet proposes a pairwise attention network with a guidance mechanism for detecting HOIs by encoding spatial contextual information. We leverage spatial configurations and object semantic information to guide our attention network.

Many of the current works also use different additional features [6, 34, 53–55, 72] ranging from pose information of humans to 3D representations of 2D images for detecting

HOIs. Our work, VSGNet [53] has shown the effectiveness of relative spatial configurations for refining visual features. We utilize the similar configuration in GTNet. Further, we combine object semantics [59, 73] with spatial configurations [6] for our guidance system. Our empirical results show that this combination performs better as a guidance mechanism.

Transformer Network: Recently, Transformer [26] based networks have achieved the state of the art performances in different vision tasks [68, 74, 75]. For detecting HOIs, one of the first attempts to use transformer like self-attention was [76]. Following that work, Girdher et al. [77] has developed the Actor Transformer network for detecting human activities in videos. Few recent works [68–71] have developed one stage Transformer networks for detecting HOIs. However, all these works rely only on self-attention mechanism to find salient context in the Transformer architecture. In contrast, we guide our joint feature representations of each human-object pair with spatial configurations and object semantics to find salient spatial context. Our state of the art performance over standard datasets validates our design choices for the self-attention network.

3.3 Technical Approach

In this section, we introduce GTNet architecture for Human-Object interaction (HOI) detection. Given an input image, the task is to generate bounding boxes for all humans and objects while detecting the interactions among them (e.g. a person hitting a ball). Each human-object pair can have multiple interactions. We use a pre-trained object detector to detect humans and objects in the images.

GTNet takes image features $\mathbf{F}$ as input. For extracting features, we use standard feature extractor networks (e.g., resnet [45] and efficientnet [78]). Consider a single image of size $c \times h \times w$ where c, h, and w are the number of channels, height and width

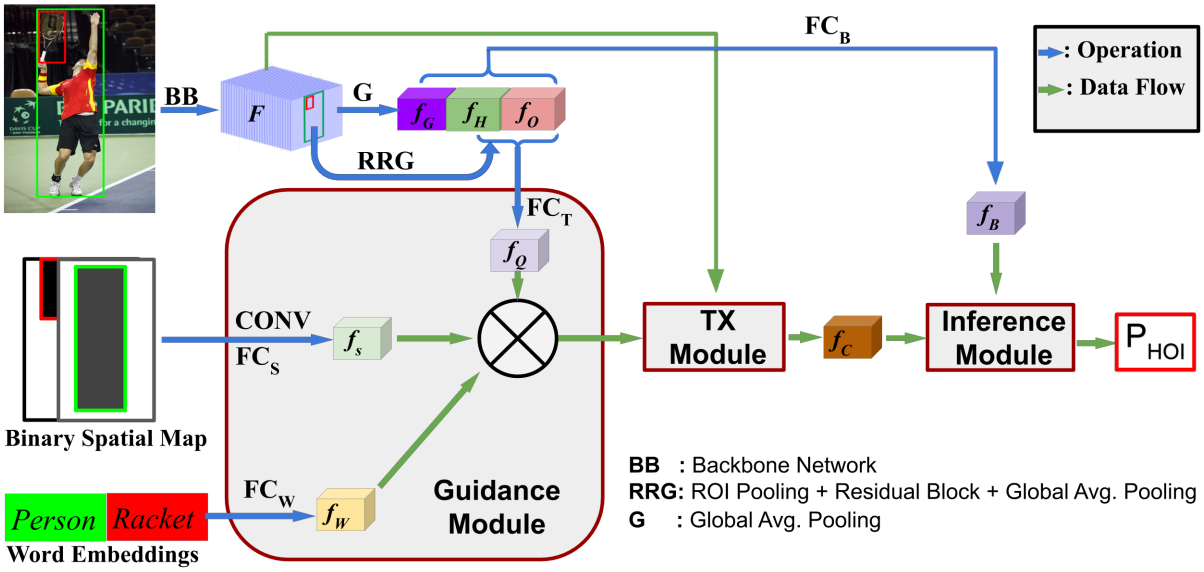

Figure 3.2: Model Overview. We extract human and object feature vectors from the input feature map, $\mathbf{F}$ via two RRG operations. For clarity we do not explicitly show two separate RRG operations. Human and object features are used to generate a query vector, $\mathbf{f}_Q$. Before feeding $\mathbf{f}_Q$ to the TX Module, we guide it via an element wise product with spatial ($\mathbf{f}_S$) and semantic ($\mathbf{f}_W$) guidance feature vectors. Inside the TX Module, contextual information is encoded to the guided query vector to generate a context-aware updated query vector $\mathbf{f}_C$. Finally, we make HOI predictions ($\mathbf{P}_{HOI}$) from the updated query and the baseline feature vectors in the Inference Module. Details of TX Module in Figure 3.4.

of the image. For a given image, $\mathbf{F}$ has a dimension of $C \times H \times W$ in the feature space. The relationship between C, H, W and c, h, w depends on the backbone network.

Along with image features, for each human and object our network takes bounding boxes b_h and b_o as inputs when there are $M \geq 1$ humans and $N \geq 1$ objects present in the image. As mentioned earlier, we use a pre-trained object detector to generate those bounding boxes. Our network will predict an interaction probability vector $\mathbf{p}_{HOI}$ for each human-object pair. In the next sections, we will describe different components of our network.

3.3.1 Baseline Module

Our baseline module extracts human and object feature vectors $\mathbf{f}_H$, and $\mathbf{f}_O$ from the input feature map $\mathbf{F}$. Following existing work [35] we use region of interest pooling followed by a residual block and average pooling to extract these feature vectors. Moreover, we use the overall feature map $\mathbf{F}$ to get generalized context information by extracting feature vector $\mathbf{f}_G$ with average pooling.

To get a joint representation of these feature vectors, we concatenate and project them in the same space using a fully connected (FC) layer.

$$\mathbf{f}_B = \mathbf{FC}_B(\mathbf{f}_H \parallel \mathbf{f}_O \parallel \mathbf{f}_G) \tag{3.1}$$

Here, $\parallel$ represents concatenation. This naive feature vector $\mathbf{f}_B$ is not adequate to detect all fine-grained HOIs. In the next sections, we describe how to refine and couple visual features with human-object pairwise spatial contextual information.

3.3.2 Guidance Module

Vaswani et al. [26] propose the Transformer network for processing sequential data in natural language processing (NLP). This network uses self attention [58] to find contextual dependence in sequential data. The original Transformer network introduces the concept of queries, keys, and values, which we adopt to the context of HOI detection. In this section, we explain the idea of queries and the mechanism to guide it. In the next section, we explain our attention mechanism.

Queries are defined as the pairwise joint representation of human and object feature vectors ($\mathbf{f}_H$, $\mathbf{f}_O$). We get the pairwise representation by concatenation and projection.

$$\mathbf{f}_Q = \mathbf{FC}_T(\mathbf{f}_H \parallel \mathbf{f}_O) \tag{3.2}$$

Figure 3.3: Importance of semantic priors in the guidance mechanism. Both of the human-object pairs have similar relative spatial relations. However, one person is holding a phone and the other person is holding corn. Object semantics along with the relative spatial configurations in the guidance mechanism will help to guide the query vector to encode rich spatial contextual information to distinguish the two interactions: holding corn and talking on the phone.

Here, $\mathbf{f}_Q$ is the query vector, and $\|$ represents concatenation operation. For a single human-object pair query vector has a length of D. This query vector will be used to find relevant spatial context in the feature map.

For action recognition, Girdhar et al. [77] embedded bounding box sizes and locations in the queries as additional information to the network. HOI detection is a more subtle task as, along with the size of the humans and the objects, relative configurations among them are also important. To this end, we propose our guidance module that uses relative spatial configurations and semantic representations of humans and objects to guide the attention mechanism.

Spatial Guidance: Relative spatial configurations have proven to be very useful [6, 34, 35, 53] in detecting HOIs. We use a two-channel binary map with a dimension of $2 \times s \times s$ (s is chosen as 64, see Table 3.6) to encode relative spatial configurations. For a human-object pair, the first channel is 1 in the location of the human-bounding box, whereas

the second channel contains 1 in the location of the object bounding box. Everywhere else is zero in the binary map. Following [6], we use two convolutional layers along with average pooling and linear projection($\mathbf{FC}_S$) to get a feature vector $\mathbf{f}_S$, representing the relative spatial configurations. We utilize $\mathbf{f}_S$ to guide the TX Module.

Semantic Guidance: Although $\mathbf{f}_S$ is a strong cue to detect HOIs, the information it conveys can be confusing without proper knowledge of the object. An example of this is shown in Fig. 3.3 where the interaction looks similar but the objects are clearly different. That is why we combine object semantics with spatial configurations in our guidance mechanism. Instead of just using a look-up table for identifying each object, we use word embeddings (vector representation of words) from the publicly available Glove [79] model. For our specific case, every detected object by the object detector is represented with a vector. For example, phone and football are presented by two different vectors. We express humans with one fixed vector. After concatenation of these human and object word embedding vectors together we get a combined feature vector $\mathbf{f}_W$ with linear projection ($\mathbf{FC}_W$). Along with $\mathbf{f}_S$, we utilize $\mathbf{f}_W$ in the guiding mechanism.

Guidance Mechanism: The information present in $\mathbf{f}_S$ and $\mathbf{f}_W$ is used to guide the queries sent to the TX Module. Guiding here refers to encoding relative spatial configurations and object semantics to the queries before feeding them to the TX Module. This can be achieved either by concatenation or by taking the element-wise product among the spatial, semantic ,and the query vectors. We use element-wise product as it is proven to be more effective guidance mechanism (See Table 3.4) than concatenation.

$$\mathbf{f}_{GQ} = \mathbf{f}_Q \circ \mathbf{f}_S \circ \mathbf{f}_W \tag{3.3}$$

Here, $\circ$ represents element wise dot product. The guided query vector $\mathbf{f}_{GQ}$ is subsequently fed into the TX Module.

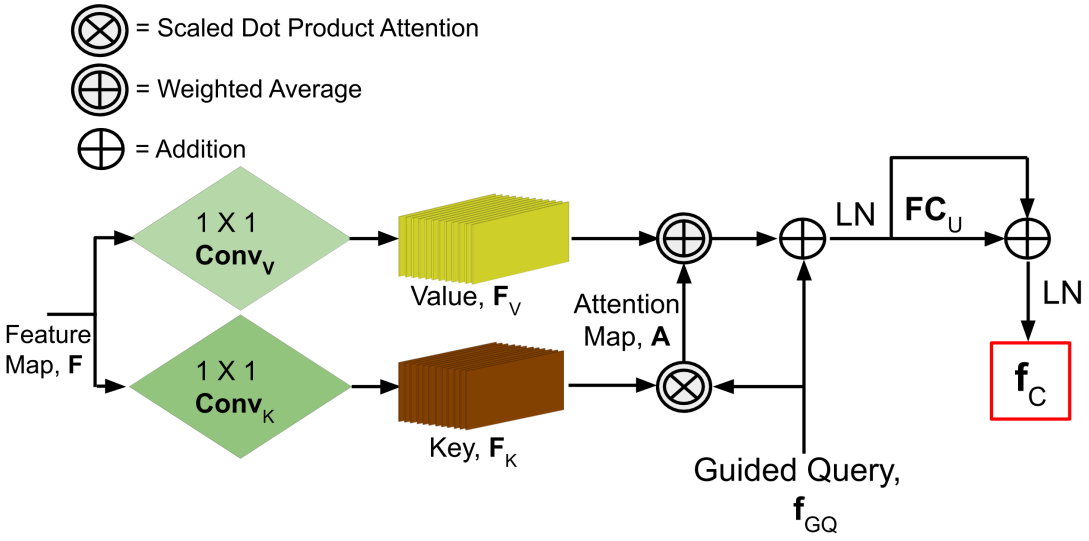

Figure 3.4: TX Module. From the input feature map, we generate key and value by 1×1 convolutions. With scaled dot-product attention, we produce an attention map for a particular human-object pair from the query and the key. This attention map is used to weigh the value to derive the contextually rich feature vector $\mathbf{f}_C$.

3.3.3 TX Module

The TX module is our adaptation of the original Transformer architecture. Figure 3.4 shows the details of TX Module. This module encodes spatial contextual information to the guided query vectors via attention. We leverage the concept of keys and values from the original Transformer architecture in this mechanism.

Keys ($\mathbf{F}_K$) and values ($\mathbf{F}_V$) are generated from the input feature map ($\mathbf{F}$) by two separate 1×1 convolutions.

$$\mathbf{F}_K = \mathbf{conv}_K(\mathbf{F}) \tag{3.4}$$

$$\mathbf{F}_V = \mathbf{conv}_V(\mathbf{F}) \tag{3.5}$$

Here, $\mathbf{conv}_K$ and $\mathbf{conv}_V$ are two different 1×1 convolution operations. $\mathbf{F}_K$ and $\mathbf{F}_V$ can be thought of as two different representations of the input feature map. For a single

image, both of them have the dimension of $D \times H \times W$.

The guided query vectors are used to search keys, $\mathbf{F}_K$ to find pairwise contextual information. Imagine a person is talking on phone in an image. Our guided query vector representing the person and the phone pair finds the important region (i.e. close to the ear) in $\mathbf{F}_K$ to detect the interaction: talking on phone.

This search process is done by scaled dot-product attention (equ. 3.6). In each spatial location of $\mathbf{F}_K$, we take a channel-wise scaled dot-product with the guided query vector. Softmax is applied to present the important context for a particular guided query in probabilities.

$$\mathbf{A} = \mathbf{Softmax}(\frac{\mathbf{f}_{GQ}\mathbf{F}_K^T}{\sqrt{\mathbf{D}}}) \tag{3.6}$$

For a particular guided query, $\mathbf{A}$ is the attention map where each element signifies the probability of that spatial location to be significant for detecting interactions. $\mathbf{A}$ has a dimension of $H \times W$ for each guided query vector. We use the attention map $\mathbf{A}$ to weigh $\mathbf{F}_V$ to get updated contextually rich query vectors. Following Transformer [26], we use fully connected layers with residual connections in this process.

$$\mathbf{f}_C = \sum_{H,W} \mathbf{A} * \mathbf{F}_V \tag{3.7}$$

$$\mathbf{f}_C = LN(\mathbf{f}_C + \mathbf{f}_{GQ}) \tag{3.8}$$

$$\mathbf{f}_C = LN(\mathbf{f}_C + \mathbf{FC}_C(\mathbf{f}_C)) \tag{3.9}$$

Here, LN means layer norm. It is used to stabilize the operations and prevent overfitting during training. Also, $*$ represents the Hadamard product between $\mathbf{A}$ and each channel of $\mathbf{F}_V$. The resultant weighted $\mathbf{F}_V$ was averaged over the spatial dimensions. $\mathbf{f}_C$ is the

contextually enriched feature vector that will be used in the Inference Module.

We stack multiple TX Modules together to get better context representations in the updated queries like the original Transformer architecture [26, 77].

3.3.4 Inference Module

According to [80], fusing features from different layers of a neural network increases the expressive capability of the features. Therefore, we fuse features from our Baseline Module with the TX Module. Moreover, we refine the naive visual features $\mathbf{f}_B$ in the same way as equ. 3.3.

$$\mathbf{f}_{BR} = \mathbf{f}_B \circ \mathbf{f}_S \circ \mathbf{f}_W \tag{3.10}$$

We concatenate $\mathbf{f}_B$, $\mathbf{f}_{BR}$, $\mathbf{f}_C$ together to increase the diversity in the final feature representation. With fully connected layers we make class-wise predictions for each human-object pair over this concatenated feature. Following [34, 53], we also generate an interaction proposal score, $\mathbf{b}_I$ using the baseline features for each human-object pair. This score represents the probability of interactions between a human-object pair irrespective of the class.

$$\mathbf{p}_I = \sigma(\mathbf{FC}_P(\mathbf{f}_B \parallel \mathbf{f}_{BR} \parallel \mathbf{f}_C)) \tag{3.11}$$

$$\mathbf{b}_I = \sigma(\mathbf{FC}_{P_B}(\mathbf{f}_B \parallel \mathbf{f}_{BR})) \tag{3.12}$$

Here, σ represents a sigmoid non-linearity. For each human-object pair, we achieve our final predictions by multiplying these two individual predictions.

$$\mathbf{p}_{HOI} = \mathbf{p}_I \times \mathbf{b}_I \tag{3.13}$$

$\mathbf{p}_{HOI}$ has a length equal to the number of classes in considerations.

3.3.5 Loss Function

As HOI detection is a multi-label detection task (multiple interactions can happen with the same human-object pair), almost all prior works use binary cross entropy loss for each class to train the network. However, as pointed out by [4, 53], confusing labels, missing labels, and mislabels are common in the HOI detection datasets. To handle those scenarios we utilize Symmetric Binary Cross Entropy (SCE) from [81] instead of only using binary cross entropy. This idea is derived from symmetric KL divergence and defined by:

$$\mathbf{SCE} = \alpha\mathbf{CE} + \beta\mathbf{RCE} = \alpha\mathbf{H}(p, q) + \beta\mathbf{H}(q, p) \tag{3.14}$$

where, **CE** is the traditional binary cross entropy, **RCE** is reverse binary cross entropy, **H** is entropy, **p** is target probability distribution and **q** is predicted probability distribution, α and β are the weight values for each type of the loss. CE is useful for achieving good convergence, but it is intolerant to noisy labels. RCE is robust to noisy labels, as it compensates the penalty put on network by CE when the target distribution is mislabeled but the network is predicting a right distribution. We select α and β as 0.5 to balance both kinds of losses.

3.4 Experiments & Analysis

In this section, we first describe our experimental setup and implementation details. We then evaluate GTNet's performance by comparing with previously state of the art methods. Next, we validate our design choices via different ablation studies. Finally, we analyze network's performance qualitatively.

3.4.1 Experimental Setup

Datasets: There are two widely used publicly available datasets for the HOI detection task: V-COCO [1], HICO-DET [6].

V-COCO is a subset of the COCO dataset [47] and contains in total 10,346 images. The training, validation, and testing splits have 2,533; 2,867; and 4,946 images. Among its 29 classes, 4 do not contain any object annotations, and one of the classes has very few samples (21 images). Following previous works, we report our model's performance in the rest of the 24 classes. HICO-DET has in total 47,776 images: 38,118 for training, and 9,648 for testing. With 117 interaction classes, HICO-DET annotates in total 600 human-object interactions. Based on the number of training samples, the dataset is split into Full (all 600 HOI categories), Rare (HOI categories with sample number less than 10), and Non-Rare categories (HOI categories with sample number greater than 10) [6]. We evaluate GTNet's performance in these categories.

Evaluation Metrics: We follow the protocol suggested by both datasets [1, 6] and report our model's performance in terms of mean average precision (mAP). A prediction for a human-object pair is correct if the predicted interaction matches the ground truth and both the human and object bounding boxes have an intersection over union (IOU) score of 0.5 or higher with their respective ground truth boxes. For V-COCO, there

are two protocols (Scenario 1 and Scenario 2) for reporting mAP [1]. When there is an interaction without any object (human only), in scenario 1 the prediction would be correct if the bounding box for the object is $[0, 0, 0, 0]$, in scenario 2 in these human only cases the bounding box for the object is not considered. For HICO-DET there are two settings: default and known. In default setting all images are considered to calculate AP for a certain HOI whereas in known setting only images that contain the particular object involved in that HOI are considered.

3.4.2 Implementation Details

As a backbone of GTNet, we experimented with different architectures and selected resnet-152 [45] based on performance in the training set of V-COCO and HICO-DET. We also report our performance using resnet-50 [45] for a fair comparison with existing methods. Output from the fourth residual block of resnet with a dimension of $1024 \times 25 \times 25$ was used as the the input feature map. By two separate 1×1 convolutions we generate key and value with a dimension of $512 \times 25 \times 25$.

For training, following the policy of previous works [4,59] we use human-object bounding boxes from a pre-trained Faster-RCNN [25] based object detector. For selecting the human and object bounding boxes we use 0.6 threshold for human and 0.3 threshold for object. All projected feature vectors (baseline feature vector $\mathbf{f}_B$, query vector $\mathbf{f}_Q$, spatial guidance vector $\mathbf{f}_S$, semantic guidance vector $\mathbf{f}_W$, guided query vector $\mathbf{f}_{GQ}$, and context rich query vector $\mathbf{f}_C$) have a length of 512.

Hyper parameters of the network were selected by validating on V-COCO's validation set and a small split from the training set of HICO-DET. Our initial learning rate was 0.001 and the optimizer was Stochastic Gradient Descent (SGD) with a weight decay and a momentum of 0.9 and 0.0001. Our network was trained on GeForce RTX NVIDIA

Method	Feature Backbone	Scenario 1	Scenario 2
DRG [4]	ResNet50-FPN	51.0	-
Wan et al. [54]	ResNet50-FPN	52.0	-
Zhong et al. [61]	ResNet - 152	52.6	-
H. Wang et.al. [57]	ResNet50-FPN	52.7	-
Kim et al. [62]	ResNet - 152	53.0	-
Liu et al. [55]	ResNet-50	53.1	-
ConsNet [59]	ResNet-50	53.2	-
IDN [64]	ResNet-50	53.3	<u>60.3</u>
OSGNet [82]	ResNet-152	53.4	-
Sun et al. [83]	ResNet - 101	55.2	-
GTNet (Ours)	ResNet-50	<u>56.2</u>	60.1
GTNet (Ours)	ResNet-152	**58.29**	**61.85**

Table 3.1: Performance comparisons in the V-COCO [1] test set. Many current works do not report their models' performance in Scenario 2. Best results in each category are marked with **bold** and the second best results in those categories are marked with <u>underline</u>.

GPUs (one 2080 Ti for V-COCO, four 2080 Ti and four 1080 Ti for HICO-DET dataset) with a batch size of 8 per GPU. For each input image during training we randomly apply two augmentations from a set of augmentations (affine transformations, rotation, random cropping, random flipping, additive Gaussian noise, etc.). Our model has $\sim 60M$ parameters. Total training time for V-COCO was 7 hours and for HICO-Det was 80 hours. The network was trained with symmetric binary cross entropy (See Section 3.3.5) for 60 epochs on V-COCO dataset and 350 epochs on HICO-Det dataset. In V-COCO, the learning rate is increased to 0.01 between 10 to 33 epochs. In HICO-DET, the learning rate is reduced by a factor of 10 after each 20 epochs.

During inference, for each human-object pair, we multiply class-wise predictions $\mathbf{p}_{HOI}$ from equ. 3.13 with the detection confidence scores of the humans and objects. Also, we apply Low grade Instance Suppressive Function (LIS) [34] to improve the quality of the object detection confidence scores.

Method	Feature Backbone	Full(def)	Rare(def)	None-Rare(def)	Full(ko)	Rare(ko)	None-Rare(ko)
UnionDet [62]	ResNet50-FPN	17.58	11.72	19.33	19.76	14.68	21.27
IPNet [84]	Hourglass-104	19.56	12.79	21.58	22.05	15.77	23.92
PPDM [85]	Hourglass-104	21.1	14.46	23.09	-	-	-
Bansal et al. [86]	ResNet-101	21.96	16.43	23.62	-	-	-
Hou et al. [63]	ResNet-50	23.63	17.21	25.55	25.98	19.12	28.03
ConsNet [59]	ResNet-50	24.39	17.1	26.56	-	-	-
DRG [4]	ResNet50-FPN	24.53	19.47	26.04	27.98	23.11	29.43
IDN [64]	ResNet-50	26.29	22.61	27.39	28.24	24.47	29.37
VSGNet [53]	ResNet-152	26.54	21.26	28.12	-	-	-
ATL [66]	ResNet-50	27.68	20.31	29.89	30.05	22.40	32.34
FCL [67]	ResNet-50	<u>29.12</u>	**23.67**	<u>30.75</u>	<u>31.31</u>	**25.62**	<u>33.02</u>
GTNet (Ours)	ResNet-50	26.78	21.02	28.50	28.80	22.19	30.77
GTNet (Ours)	ResNet-152	**29.71**	<u>23.23</u>	**31.64**	**31.64**	<u>24.42</u>	**33.81**

Table 3.2: Performance comparisons in the HICO-DET [6] test set. Def and ko mean default and known settings respectively. Best results in each category are marked with **bold** and the second best results in those categories are marked with <u>underline</u>.

3.4.3 Results

We now compare GTNet's performance with the current state of the art methods. As mentioned in Section 3.4.1, we follow the evaluation protocol suggested by the V-COCO [1] and HICO-DET [6] datasets. Moreover, for fair comparison we only compare our method with two stage HOI detection networks.

For testing on V-COCO, we use the object detection results from our work VS-GNet [53], which come from an object detector trained on COCO. For testing on HICO-DET, we use the object detection results provided by DRG [4], which is used by the state of the art models [4, 64]. These detections come from an object detector trained on COCO and finetuned on the training set of HICO-DET. GTNet achieves state of the art results in all datasets. Also, as most prior works either use resnet-152 [61, 62, 82] or resnet-50 [4, 55, 59, 64] as feature backbone, we report our network's performance using both of these backbones.

In Table 3.1 and Table 3.2 GTNet's performance can be seen in V-COCO [1] and HICO-DET [6] datasets. Our network outperforms all other existing methods on the V-COCO dataset in both protocols. The current state of the art method [83] used a

Method	Feature Backbone	Full(def)	Rare(def)	None-Rare(def)
iCAN [35]	ResNet-50	33.38	21.43	36.95
Li et al. [34]	ResNet-50	34.26	22.9	37.65
Peyre et al. [73]	ResNet-50-FPN	34.35	27.57	36.38
GTNet (Ours)	ResNet-50	<u>44.71</u>	33.80	<u>47.97</u>
GTNet (Ours)	ResNet-152	**49.35**	<u>36.93</u>	**53.07**

Table 3.3: Performance comparisons in the HICO-DET [6] test set with oracle object detector. Def means default setting. Best results in each category are marked with **bold** and the second best results in those categories are marked with <u>underline</u>.

fusion method to fuse features from seven branches. We clearly outperform them with both of our backbones.

Moreover, GTNet outperforms all other existing methods in the difficult default settings. Also, though Transformer based network's needs good amount of data to train [87], we manage to achieve second best performance in the RARE category. Actually, HICO-DET's rare category has 138 HOI classes with an average of 3 samples per class while 45 classes in rare category has only 1 sample in the training set. We expect to achieve much better performance in this category with a healthy number of samples.

The closest work to our network is DRG [4], which utilizes a disjoint self-attention mechanism for each human and object in a dual relation graph network. By contrast, GTNet leverages the query, key, and value concept to encode pairwise contextual information in the queries along with a guiding mechanism and achieves significant improvement (7 mAP improvement in V-COCO and 5 mAP improvement in HICO-DET).

3.4.4 Oracle Object Detector

The performance of all two-stage HOI detection models heavily relies on the quality of the object detector. In Table 3.3, we evaluate HOI detection performance using an oracle object detector. By oracle object detector, we mean providing the two-stage models with ground truth object bounding boxes. In this context, our model exhibits

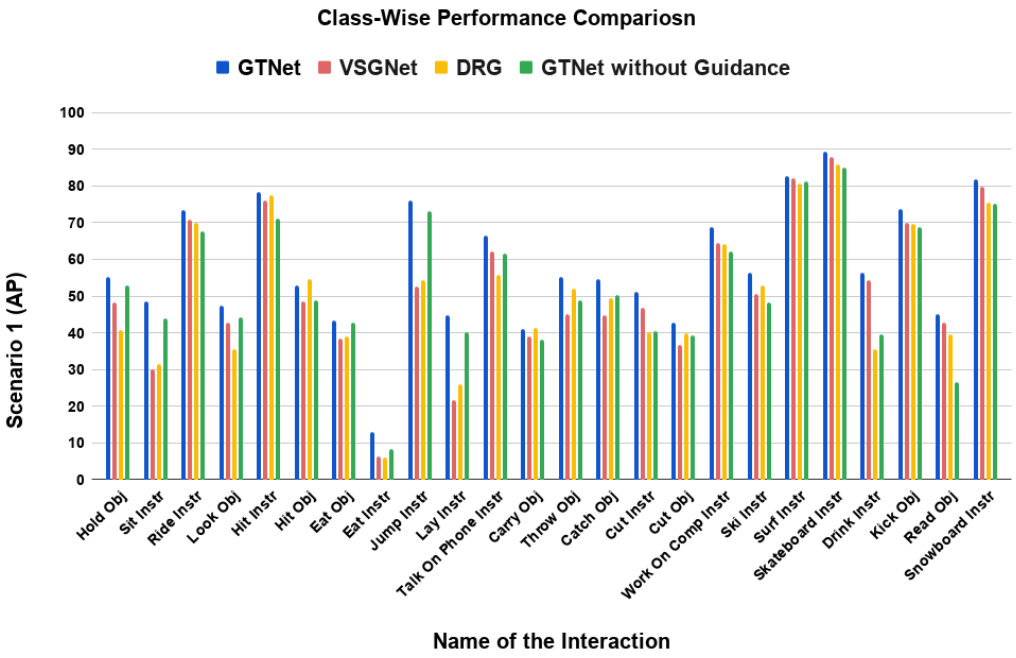

Figure 3.5: Class-wise performance comparison of GTNet with VSGNet [3] and Gao et al. [4] in V-COCO test set. Moreover, we also compare GTNet's performance without the guidance mechanism to show the effectiveness of the Guidance Module. Obj is object and instr is instrument [1].

markedly superior performance, achieving a significant improvement over the current state-of-the-art. This underscores the effectiveness of our architecture, suggesting that when paired with a superior object detector, our model outshines its competitors.

3.4.5 Classwise Performance Analysis

We compare GTNet's class-wise performance in V-COCO [1] test set with few SOTA methods in Figure 3.5. We also compare our network's performance without the guidance mechanism to demonstrate the effectiveness of the Guidance Module. Without the guidance mechanism, GTNet's performance is close to the state of the art methods, and with its guidance mechanism it exceeds the recent methods in most of the classes. For a few classes (jump instr, lay instr) the improvement from the recent state of the art meth-

	Scenario 1	Scenario 2
GTNet	**58.29**	**61.85**
without Spatial Guidance	57.27	61.39
without Semantic Guidance	53.46	57.10
without Guidance Module	52.45	56.61
without TX Module	51.65	55.81
without reverse cross entropy loss	56.35	59.85
without interaction proposal score	57.27	60.97
without data augmentation	56.00	57.18
guided by concatenation	57.02	61.20

Table 3.4: Ablation studies of GTNet in V-COCO test set. GTNet achieves state of the art performance with the guidance mechanism. It is interesting to observe that, in the same dataset, semantic guidance performs better than spatial guidance when tested independently. Also, it shows the effectiveness of symmetric cross entropy, interaction proposal score and data augmentation.

ods are more than 10 AP. Moreover, in a small number of classes, the object detectors perform badly. As a result the overall AP values are low in those classes. Eating with utensils (instruments) are usually very small and not clearly visible in many images, so the object detectors miss them. Even with poor object detection results in eat instrument class, GTNet improves that class's performance by ~ 6 AP over recent state of the art methods.

3.4.6 Ablation Studies

Effect of various Components and Training Policy: GTNet consists of several small modules and a unique training policy including the use of symmetric cross entropy loss, data augmentation, etc. In this section, we examine their effectiveness in our overall performance in V-COCO test set (Table 3.4). As shown in Table 3.4 without the TX module, our baseline network achieves a mAP of 51.65. The performance improves slightly with the introduction of TX module. It is expected because, without the guidance mechanism, it is difficult to encode rich spatial contextual information to the visual

Number of Channels	Scenario 1	Scenario 2
64	57.21	61.1
128	57.46	61.6
256	57.48	61.25
512	**58.29**	**61.85**

Table 3.5: Performance of GTNet with different number of channels for key and value in V-COCO test set.

s	Scenario 1	Scenario 2
4	56.84	60.63
16	57.30	60.99
32	57.95	60.63
64	**58.29**	**61.85**

Table 3.6: Performance of GTNet with different size of binary spatial map, s in V-COCO test set.

features. With the help of the guidance mechanism, we improve our result by more than 6 mAP, highlighting the importance of the guiding mechanism in the attention framework. Also, our experiments demonstrate that semantic guidance is more effective than spatial guidance. Additionally, in Table 3.4 we have shown GTNet's performance without reverse binary cross entropy loss (just using binary cross entropy loss), data augmentation, and interaction proposal scores. All these actually helps our network to achieve superior performance. Moreover, as mentioned in Section 3.3.2 guidance can be achieved by either concatenation or product. As can be seen in Table 3.4, with guidance by concatenation we achieve 57.02 mAP compared to 58.29 mAP achieved with guidance by product.

Number of Channels and Size of the Spatial Map : We take 1×1 convolution over the input feature map to generate key and value with different number of channels. As can be seen in Table 3.5 with the increasing number of channels the network seems to perform a little better.

Similar trend can be seen in Table 3.6 with the increasing size of binary spatial map,

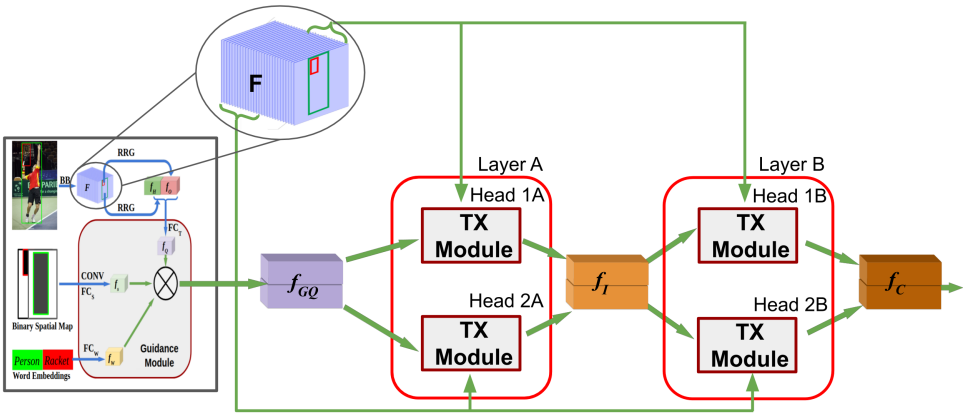

Figure 3.6: Stacking of TX Module. Two layers, two heads combination. Red colored boxes representing each layer. Each TX module inside the red box represents a head. Guided query vector $\mathbf{f}_{GQ}$ and input feature map $\mathbf{F}$ are divided into two equal parts to feed into two heads (Head 1A, Head 2A) in Layer A. The output of Layer A is concatenated to create $\mathbf{f}_I$ and fed to the two heads (Head 1B, Head 2B) in Layer B. The final spatial context rich feature vector is the output of Layer B ($\mathbf{f}_C$) which will be fed to the Inference Module.

heads \layers	1	2	3
1	56.2	56.0	55.9
2	55.9	**58.29**	56.2
4	56.3	55.9	55.8

Table 3.7: GTNet's performance on V-COCO test set for different number of heads and layers.

s in the spatial guidance mechanism. We choose 512 as the channel size for key and value and 64 as the size of the binary spatial map considering memory constraint in our GPU.

Stacking of TX Modules: To get a better feature representation in the updated queries we stack multiple TX modules. Following [77], we use the concept of heads and layers.

Each layer is made up of H heads. The input to the first layer is $\mathbf{f}_{GQ}$, the guided query. This input is split into H parts, one for each head. Similarly, the feature map $\mathbf{F}$ is split into H parts such that the input to each head is one part of $\mathbf{F}$ and one part of $\mathbf{f}_{GQ}$. The input to each intermediate layer is the concatenated output of the previous

Backbone	Scenario 1(mAP)
Vgg19 [50]	53.4
Mobile Net [88]	53.1
Dense Net [89]	54.2
Resent-34 [45]	54.6
Resnet-50 [45]	54.6
Resnet-101 [45]	55.0
Resnet-152 [45]	**56.4**

Table 3.8: GTNet's performance with different backbones. As can be seen, Resnet-152 achieves the best mAP in Scenario 1 of V-COCO test set.

layer and the feature map **F**. All layers get the same **F** but a modified query vector. We always use same number of heads in all the layers. Fig 3.6 illustrates an example of a configuration with two heads and two layers. We experiment with different combinations and find that two heads and two layers combination perform best in V-COCO test set (Table 3.7). Similarly, for HICO-DET we empirically choose a combination of two heads and three layers. As can be seen, the network's performance is not very sensitive to different combinations.

Different Backbone Networks: We test with different Convolutional Neural Networks(CNN) as our feature extractor backbone networks. Table 3.8 shows the performance of different backbones. We use Resnet-152 as our backbone as it achieves best result among different backbones.

3.4.7 Qualitative Results

Visualized Detections: In Fig. 3.7 we visualize a few predicted HOIs by GTNet. Each column in the figure represents a different situation. GTNet performs well across various situations. Moreover, our network performs poorly when interacting human is not fully present in the image. This case is shown in column (e) in Fig. 3.7.

Activation Maps: Figure 3.8 shows some detection performances of GTNet along with

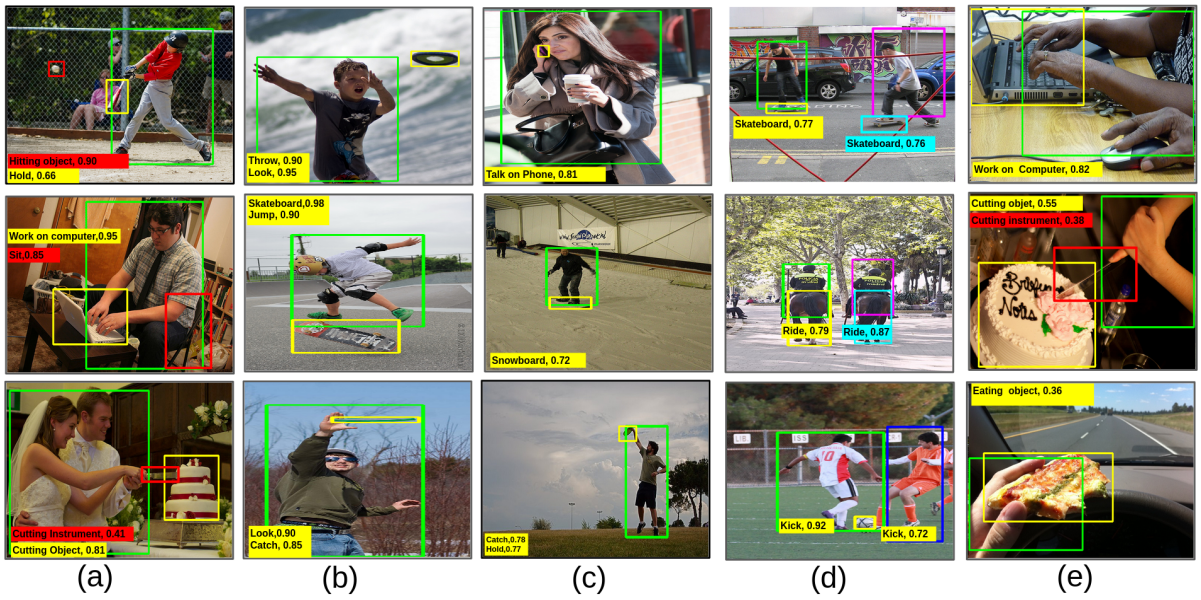

Figure 3.7: Visualized HOI detection results. Each human bounding box includes predicted interaction labels with confidence scores for that human. Interaction labels and the bounding boxes of the interacting objects are in the same color. Each column represents a different situation: (a) a single human is interacting with multiple objects, (b) no contact between interacting human and object, (c) interacting object is small or not fully visible, (d) multiple humans are interacting with either same object or different objects, (e) interacting human is not fully present in the image.

the class activation maps [5] for particular human-object pairs. As can be seen in Figure 3.8, GTNet correctly identifies the interactions with high confidence even when multiple interactions are happening together (image 1, 5). From the class activation maps it is clear that our network is finding relevant context for the interacting human-object pair.

3.4.8 Limitations

There is room for improving the proposed model further. The attention mechanism depends on the spatial and semantic information for detecting HOIs. Therefore, if the human is not fully present in the image then our network may fail to detect interaction for

Figure 3.8: Analysis of performance in V-Coco test set. The top row is showing images with a particular human-object pair. The bottom row is showing class activation maps [5] generated for these pairs along with the interaction probability for particular interactions. The red region in the class activation map means the network is putting more attention in these areas. When there are same actions done by different human-object pairs (rightmost and leftmost images), the feature map gets activated in all relevant regions for the particular interaction. However, our pair wise guided attention strategy forces the network to consider the region significant to a particular pair.

that human as the guidance mechanism has trouble generating effective guidance in these cases. Moreover, we use pre-trained object detector, as a result GTNet's performance depends on the quality of object detection. Also, some of the interactions are visually challenging/confusing to detect. A few examples of such failure cases are shown in Fig. 3.9.

3.5 Discussion

3.5.1 Comparison with Other Attention Mechanisms

We undertook a comparative analysis of GTNet in relation to another contemporary attention mechanism-based HOI detection network, notably DRG [4]. The DRG model interprets pair-wise spatial-semantic features of humans and objects by viewing them as

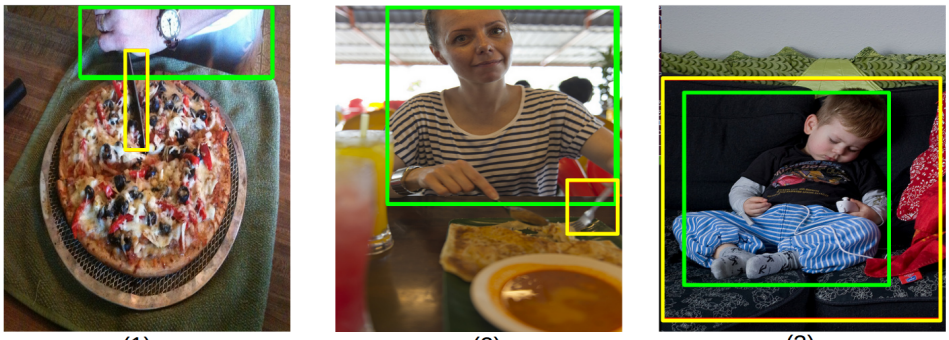

Figure 3.9: GTNet fails to detect the interactions between the marked human object pairs due to following reasons: (1) human is not fully present, (2) object detector fails to detect the spoon, (3) confusion between lay/sit interaction.

nodes within their dual relation graph. To detect HOIs, the model capitalizes on the self-attention mechanism, aggregating these features.

However, merely relying on spatial-semantic features in the attention mechanism, without integrating the actual visual features, can pose challenges for the network in comprehending and capturing the true spatial context. This potentially limits the efficacy and accuracy of the model.

In contrast, GTNet employs a more encompassing approach. It scans the entire feature map using specific queries. These queries, influenced by spatial-semantics, serve as guidance, aiding the network in identifying the pertinent spatial context. This method ensures a more nuanced and contextually informed understanding, which, as our empirical findings suggest, provides a more accurate HOI detection. Our quantitatively superior results act as a testament to the efficacy and robustness of our methodology. We would like to point out our developed model, VSGNet (described in chapter 2) is the first attempt to utilize relative spatial configurations to refine visual features. We utilize VSGNet's refinement strategy with object semantics to guide our attention mechanism. This way, our network can encode rich contextual information in the visual features.

3.5.2 Summary

We propose a novel guided self-attention network to leverage contextual information in detecting HOIs. Our pairwise attention mechanism utilizes a Transformer-like architecture to make visual features context-aware with the help of the guidance module. To the best of our knowledge, we are the first to propose a Guidance Module to guide the Transformer like attention mechanism to detect HOIs. We demonstrate by detailed experimentation that GTNet shows superior performance in detecting HOI and achieves the state of the art results in the standard datasets.

Chapter 4

Decoupled Dynamic Scene Graph

4.1 Introduction

Dynamic Scene-Graph (DSG) provides a graph structure presenting the relationships among different objects in a scene. It aims to create the scene-graph by predicting relationship triplets composed of $\langle subject, object, relationship \rangle$ at each frame of an input video. This acts as a foundational block for various computer vision tasks [3,32]. Current DSG generation systems [12, 90, 91] operate in a constrained setting where the possible triplets are predefined for a given set of relationships and objects. However, in a more realistic deployment scenario, it is likely that the network will encounter triplets that it has not seen before. Therefore, a method should be able to transfer the learned concepts of relationships and objects to compose unseen triplets. Our analysis has shown (Table 4.4) poor performance in detecting these unseen triplets using the state-of-the-art (SOTA) method. This poor performance is mainly attributed to the learning of highly dependent feature representations of relationships and objects. The proposed decoupled dynamic scene-graph (DDS) addresses this issue. Fig. 4.1 shows the core idea of the proposed DDS network. This architecture utilizes two different branches to learn decoupled features for

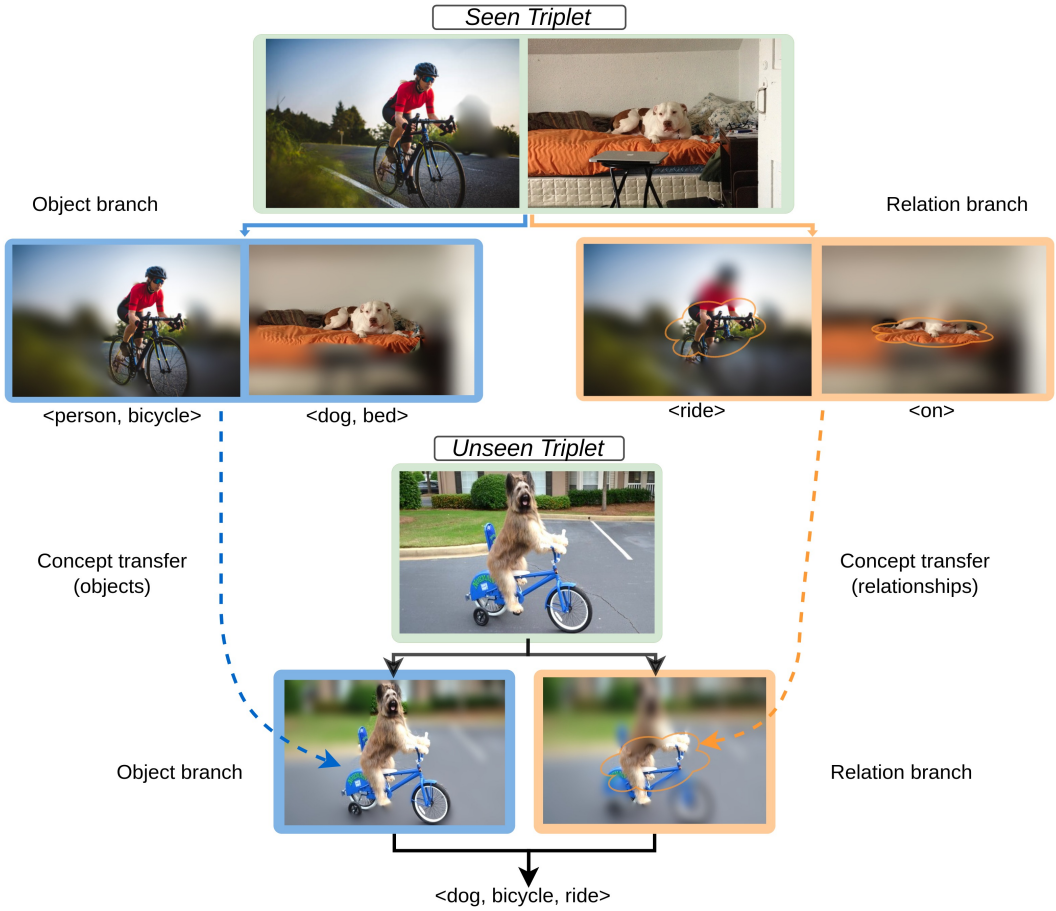

Figure 4.1: *Diagram to show the concept learning and transferring in DDS. By focusing on different spatial regions, DDS learns the concept of relationships (ride, on) and objects (person, bicycle, bed) independently. In the lower section of the diagram, we show how these learned concepts are transferred and utilized to detect the unseen triplet* $\langle dog, bicycle, ride \rangle$.

relationships and objects. As shown in the figure, DDS learns the concept of 'ride', 'on', 'person', 'bicycle', and 'bed' from the training examples of a 'person riding a bike' and a 'dog on the bed'. The decoupled design makes DDS look into different spatial regions for relationships and objects. These learned concepts are transferred to successfully detect the unseen triplet $\langle dog, bicycle, ride \rangle$, see Section 4.5.3 for more details.

Existing works [12, 90, 91] for generating DSGs follow a two-stage process. First, an object detector localizes objects irrespective of their relationship status. In the second

stage, different attention-based [90,91] and graph-based [12] methods utilize features extracted from the previously localized objects to predict relationships. In this process, the extracted features of the objects are fed throughout the network for relationship detection. As a result, these methods learn to associate a relationship only with a particular combination of objects and hence perform poorly to predict unseen triplets.

In contrast, DDS ensures the learning of discriminative spatio-temporal cues for relationships and objects. Fig. 3.2 shows the overview of our architecture. It consists of two separate branches: the relation and the object branch. We chose a Transformer based encoder-decoder [74] architecture for these branches with two different sets of queries. Moreover, a novel temporal decoder is added to embed temporal information into the queries. These separate sets of queries focus on learning generalized representations for relationships and objects from differently encoded feature maps in both temporal and spatial domains. This is significantly better than the existing works, where the same object features are used for both object and relationship detection. Also, DDS does not have the dependence on the off-the-shelf object detectors like previous works.

Our proposed model is thoroughly evaluated on the Action-Genome [92] dataset for DSG generation, where it achieves significant performance gains compared to the SOTA models. Additionally, we evaluate DDS on the task of static scene-graph (SSG) generation on the HICO-DET [6] dataset and unusual SSG generation on the UnRel [2] dataset, where DDS outperforms all the existing models in both datasets. Finally, the proposed design choices are evaluated in an extensive ablation study.

4.2 Related Works

DDS is built on the previously developed works in SSG and DSG generation. This section is used to review the literature in the mentioned areas along with additional

relevant publications on scene-graph generation under the compositional setting.

4.2.1 Static Scene-Graph (SSG) Generation

SSG generation is proposed by [93] for the task of image retrieval. An extensive literature exists [4,34,35,53–57,59–62,64,94–102] in this area. The initial works [4,34,53–56] rely heavily on two-stage (object detection and then scene-graph generation) structures. Few of these works utilize different recurrent neural network (RNN) variants [18, 103] while other prominent researches focus on graph structure [4,35,53] with attention mechanisms. Also, many authors utilize prior knowledge [53,59,104] (e.g. semantic knowledge, statistical heuristics) for SSG generation. Despite recent improvements in SSG generation, these methods are heavily constrained by their reliance on the object detection quality as noted in [105].

Modern works [68–71,105–111] in SSG generation focus on utilizing a one-stage Transformer based architecture to deal with the aforementioned issues. These approaches mainly focus on human-object interaction (HOI) detection. HOI detection is a subtask of SSG generation where the relationships among objects are limited to interaction verb [90], such as hold, work, talk. These methods employ one-stage Transformer based encoder-decoder architectures following the architecture from DETR [74]. These architectures rely on set-based predictions to generate SSG. Among these works, Qpic [68] uses a single encoder-decoder model while CDN [107] extends Qpic by using sequential decoding of objects and relationships. Additionally, MSTR [106] enables the use of multi-scale feature maps to these networks. Another concurrent work, SSRT [105] refines the overall architecture with spatial and semantic support features. Moreover, a recent line of research heavily exploits the usage of very large-scale semantic knowledge engines (e.g. CLIP [112]) [105, 108, 111]. Also, few works propose to utilize different

types of post-processing steps [111] for the task. Apart from the obvious limitation of these works being unable to utilize temporal dependencies, they perform poorly while detecting unseen triplets. With the decoupled multi-branch design, we significantly differ from these works by using separate sets of queries for relationship and object detection.

4.2.2 Dynamic Scene-Graph (DSG) Generation

DSG is an extension of the SSG where the scene-graph is created for videos. This process is harder since temporal cues need to be utilized [12, 90, 91]. Current works in this area have two-stage architectures following the initial works on SSG. Among these works, STTran [12] utilizes a temporal decoder based on the self-attention mechanism. DSGAP [91] expands STTran with an anticipatory pre-training paradigm. On the other hand, HORT [90] utilizes a multi-branch design with different types of Transformers.

Both STTran and HORT use similar features for relationship and object detection. These features come from the object bounding boxes predicted by off-the-shelf object detectors. STTran concatenates pair-wise object features to use as relationship features whereas HORT pools relationship features from a joint box of object pairs. However, using similar features for relationship and object detection forces the learning of relationships and objects to be dependent on each other. Therefore, to ensure generalized learning, we focus on learning the features independently.

4.2.3 Compositionality in Scene-Graph Generation

Creating new compositions from base known concepts during inference is known as compositional zero-shot learning (CZSL) under the compositional setting [9, 13, 28, 113, 114]. In this paper, we utilize this setting to evaluate our model. Kato et al. [115] introduce CZSL in SSG generation with an embedding-based model. Many following

works [67, 116, 117] adapt different object-affordance ideas. These works assume there exist common relationships between the subjects and the objects. Our work does not have such limited assumptions, and as a result, is able to generate scene graphs even when the relationships are very unusual (See Table 4.7).

4.3 Method

This section describes the developed model for DSG generation. Here, we propose a multi-branch network that learns distinct feature representations for relationships and objects.

4.3.1 Problem Formulation

Given an input video, $\mathbf{V} = \{\mathbf{I}_1, \mathbf{I}_2, .., \mathbf{I}_t, .., \mathbf{I}_T\}$ with T frames, the task in DSG generation is to predict a set of relationship triplets, $\{R_1, R_2, .., R_t, .., R_M\}$ at every frame of the video. Every frame has N_M number of relationship triplets. Each relationship triplet can be presented by $\langle s, o, r_{so} \rangle$. Here, s, o refers to subject, object and are represented by bounding boxes and category labels. r_{so} is the relationship between s and o. In a single frame $\mathbf{I}_t$, s and o can have multiple relations, as shown in the sample input-output pair in Figure 4.2.

In this paper, the main goal is to predict relationship triplets under the compositional setting. In this setting the test set contains triplets that are not present in the training set. Consider having in total N_o number of objects and N_r number of relations. The training set has N_s number of relationship triplets composed from the mentioned N_o objects and N_r relationships. On the other hand, the test set has N_u number of unseen triplets not present in the training set in addition to the N_s number of seen triplets, where all the unseen triplets are composed of the same N_o objects and N_r relationships.

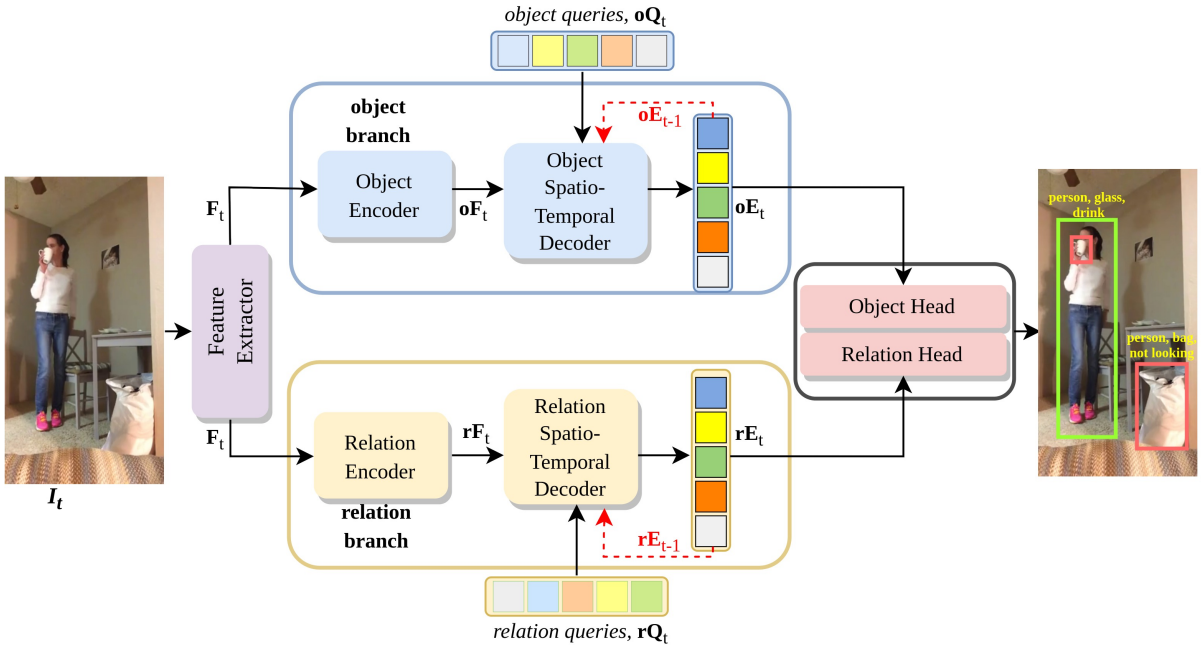

Figure 4.2: *Overview of DDS's architecture. Given an input frame* $\mathbf{I}_t$*, features are extracted by the backbone. These features are fed to the object and the relation branch. These decoupled branches consist of an encoder and a spatio-temporal decoder. Encoders from both branches encode the feature maps differently and send them to the decoders. Each spatio-temporal decoder takes a set of queries (object/relation) along with the previous frame's embeddings (shown by the red arrow). The output of the spatio-temporal decoders are learned embeddings. These learned embeddings are fed to the object and the relation heads to predict relationship triplets.*

4.3.2 Technical Overview

The proposed work adopts a one-stage approach for DSG generation compared to the current two-stage methods [12, 90, 91] as the former [68–71] have shown impressive performance in creating SSG. However, these image-based works present poor generalization capabilities. Therefore, we propose a network that uses a different set of queries with two branches: the relation branch and the object branch. Each branch follows a Transformer-like encoder-decoder architecture. Fig. 4.2 shows a diagram of the model, where a convolutional neural network (CNN) extracts features from the input frame, and those are encoded differently by the object and the relation encoders. Each spatio-

temporal decoder takes encoded features from their respective encoder in addition to two types of inputs: queries for the current frame ($\mathbf{oQ}_t$, $\mathbf{rQ}_t$) and the embeddings ($\mathbf{oE}_{t-1}$, $\mathbf{rE}_{t-1}$) propagated from the previous frame. As the encoded features differ for each branch, the queries learn decoupled features for relationships and objects. The decoder outputs are the learned object and relation spatio-temporal embeddings. These embeddings are sent to the relation and the object heads for final predictions. Moreover, these embeddings are propagated to the next frame of the video.

4.3.3 Feature Extraction & Encoders

Consider a frame $\mathbf{I}_t \in \mathbb{R}^{N_C \times H \times W}$ at time t of the input video V. Here, N_C, H, W are the number of channels, height, and width of the frame, $\mathbf{I}_t$. DDS uses a CNN network as backbone, $\mathbf{B}$ (e.g. resnet-50 [45]) to extract features $\mathbf{B}(\mathbf{I}_t) \in \mathbb{R}^{N_{C'} \times H' \times W'}$ from the input frame. Then, 1×1 convolution is used to reduce the channel dimension from $N_{C'}$ to d. After that, a flattening operation is performed and a fixed positional embedding is added to $\mathbf{B}(\mathbf{I}_t)$ like existing works [68–71] to get the feature map $\mathbf{F}_t \in \mathbb{R}^{(H'W') \times d'}$. These embeddings express each spatial position of the feature map in high dimensions [74]. DDS uses $\mathbf{F}_t$ as a common feature for both the relation and the object branches.

Both of the network's branches have an encoder comprising of stacked multi-head self-attention layers [26] with a feed-forward network (FFN). The output of the encoders are two separate feature maps,

$$\mathbf{rF}_t = \text{Relation Encoder}(\mathbf{F}_t) \tag{4.1}$$

$$\mathbf{oF}_t = \text{Object Encoder}(\mathbf{F}_t) \tag{4.2}$$

Here, $\mathbf{rF}_t$, $\mathbf{oF}_t$ refer differently encoded versions of the feature map $\mathbf{F}_t$. Our spatio-temporal decoders will utilize these feature maps for decoupled learning.

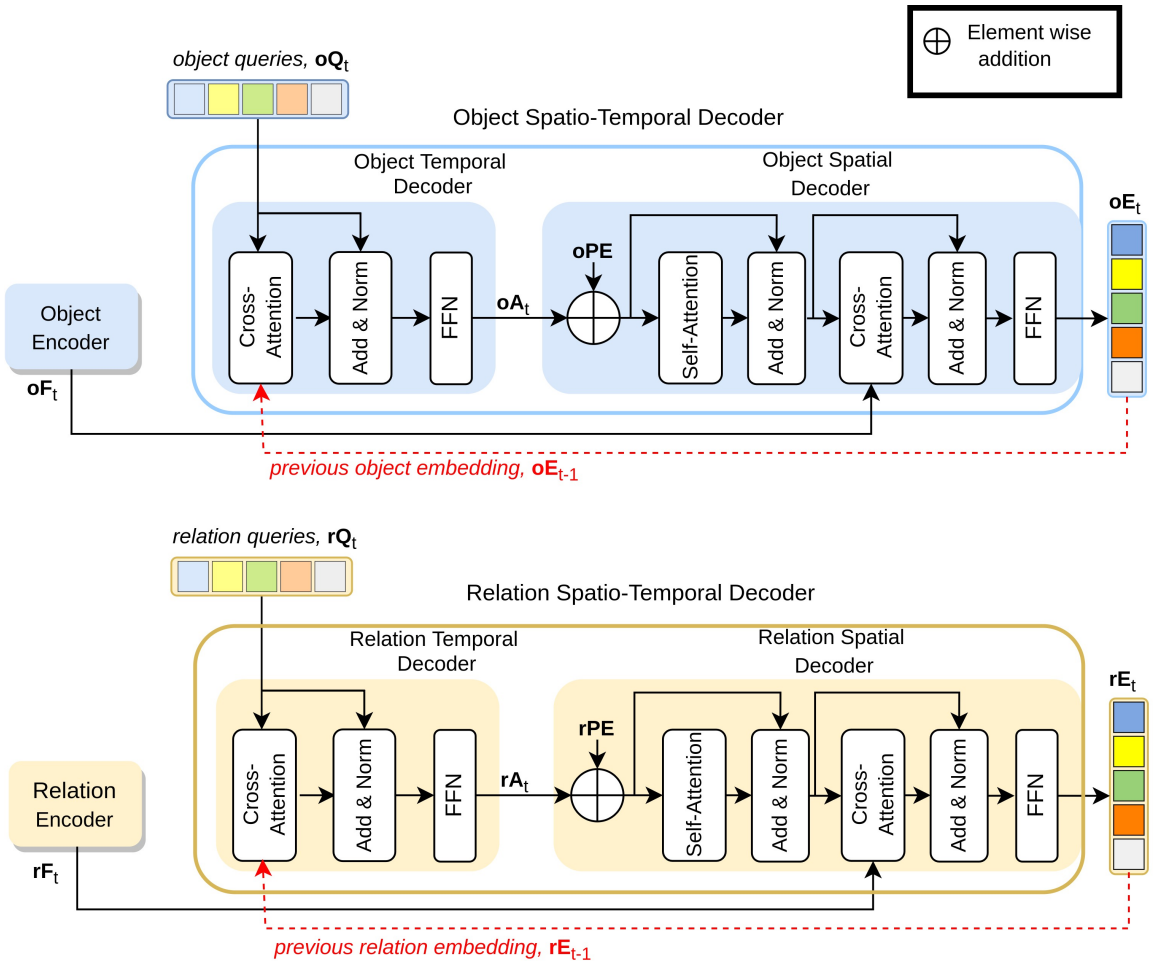

Figure 4.3: *Design of the spatio-temporal decoders. Every spatio-temporal decoder is composed of a temporal and a spatial decoder. Each decoder converts a different set of queries into learned embeddings while making sure decoupled learning in each branch.*

4.3.4 Spatio-Temporal Decoders

DDS's spatio-temporal decoders convert a set of learnable queries into output embeddings. This transformation occurs in two stages. In the first stage, the current frame's queries attend to the previous frame's output embeddings to aggregate temporal information. In the second stage, these aggregated queries gather information from the encoded feature maps of the current frame. The proposed multi-branch design ensures discriminative feature learning for the queries of each branch. Each decoder consists of two small

components: temporal and spatial decoder. Please see Fig. 4.3 for reference.

Temporal Decoders: These decoders allow queries to leverage temporal dependencies. Each temporal decoder takes two sets as inputs: the current frame's queries and the embeddings from the previous frames. For frame $\mathbf{I}_t$, the current frame's relation and object queries sets are defined as $\mathbf{rQ}_t \in \mathbb{R}^{N_q \times d}$ and $\mathbf{oQ}_t \in \mathbb{R}^{N_q \times d}$. Every query is a d dimensional vector, and every branch has N_q number of queries. Embeddings from the previous frames for the relation and the object branches are presented by $\mathbf{rE}_{t-1} \in \mathbb{R}^{N_q \times d}$ and $\mathbf{oE}_{t-1} \in \mathbb{R}^{N_q \times d}$, and are marked with red arrows in Fig.4.3.

Temporal decoders are made of stacked multi-head cross-attention layers [74] with an FFN network. The cross-attention in the temporal decoders allows the current frame's queries to select what to learn from the previous frame's embeddings. The outputs of the temporal decoders are the temporally aggregated queries $\mathbf{rA}_t$, $\mathbf{oA}_t$. They are fed to their respective spatial decoders. In the case of the first frame of a video, the temporal decoders directly output $\mathbf{rQ}_t$ and $\mathbf{oQ}_t$ as $\mathbf{rA}_t$ and $\mathbf{oA}_t$ without passing them through the cross-attention and FFN blocks as there is no previous frame in this case.

Spatial Decoders: The spatial decoders architecture is similar to the standard Transformer decoder [74]. These decoders consist of both self-attention and cross-attention layers along with FFN networks. Each decoder takes encoded feature maps ($\mathbf{rF}_t$ or $\mathbf{oF}_t$) along with the aggregated queries of the temporal decoders ($\mathbf{rA}_t$ or $\mathbf{oA}_t$) from their respective branch as inputs. Also, these decoders take learnable positional embeddings. These embeddings for the relation and the object branch are defined as $\mathbf{rPE} \in \mathbb{R}^{N_q \times d}$, $\mathbf{oPE} \in \mathbb{R}^{N_q \times d}$.

$$\mathbf{rE}_t = \text{Relation Decoder}(\mathbf{rN}_t, \mathbf{rPE}, \mathbf{rQ}_t) \tag{4.3}$$

$$\mathbf{oE}_t = \text{Object Decoder}(\mathbf{oN}_t, \mathbf{oPE}, \mathbf{oQ}_t) \tag{4.4}$$

The outputs of the decoders are the learned spatio-temporal embeddings. They are used in the object and the relation heads to make the final relationship triplet predictions. Also, each spatial decoder's output is fed to the next frame's temporal decoder as previous embeddings.

Both spatial and temporal decoders keep the separation in feature learning for relationships and objects using different encoded features, set of queries, and previous embeddings. As a result, the output embeddings from the decoders are decoupled generalized representations.

4.3.5 Object Heads

The output embeddings from the object spatio-temporal decoder, $\mathbf{oE}_t$ are fed to four different FFNs. For input frame $\mathbf{I}_t$, these FFNs predict subject bounding boxes, $\mathbf{sB}_t \in [0,1]^{N_q \times 4}$, object bounding boxes, $\mathbf{oB}_t \in [0,1]^{N_q \times 4}$, subject prediction vectors, $\mathbf{sP}_t \in [0,1]^{N_q \times N_o}$, and object prediction vectors, $\mathbf{oP}_t \in [0,1]^{N_q \times N_o}$. Here, N_q is the number of queries, and N_o is the total number of objects.

4.3.6 Relation Heads

Like the object heads, the learned output embeddings, $\mathbf{rE}_t$, are fed to two FFNs that produce as output the relation prediction vectors, $\mathbf{rP}_t[0,1]^{N_q \times N_r}$ and relation region bounding boxes $\mathbf{rB}_t \in [0,1]^{N_q \times 4}$. Here, N_q is the number of queries, and N_r is the total number of relationships under consideration. Notice that the relation region bounding box is defined as the union between the subject and object bounding boxes.

4.3.7 Inference

We compose N_q relationship pairs by one-to-one matching of $\mathbf{sB}_t$ and $\mathbf{oB}_t$. One-to-one matching refers to matching the q-th prediction from $\mathbf{sB}_t$ with the q-th prediction from $\mathbf{oB}_t$. Moreover, for every prediction vector in $\mathbf{sP}_t$ and $\mathbf{oP}_t$, the maximum confidence score is used to create $\mathbf{sP}_{tmax} \in [0,1]^{N_q}$ and $\mathbf{oP}_{tmax} \in [0,1]^{N_q}$ and the corresponding index is used to determine the category label for each of the bounding boxes. For every composed relationship pair, the final relation score prediction vectors are calculated as:

$$\mathbf{rP}_{tfinal} = \mathbf{rP}_t * \mathbf{sP}_{tmax} * \mathbf{oP}_{tmax} \tag{4.5}$$

4.3.8 Training

For training DDS, we utilize losses similar to Qpic [68]. This loss calculation implicitly binds the two sets of queries from the relation and the object branch. The loss calculation happens in two stages:

In the first stage, we find the bipartite matching between the predictions and the ground truths. First, the total prediction set for the input frame $\mathbf{I}_t$ is generated as,

$$\mathbf{P}_t = \{\mathbf{sB}_t, \mathbf{oB}_t, \mathbf{sP}_t, \mathbf{oP}_t, \mathbf{rB}_t, \mathbf{rP}_t\} \tag{4.6}$$

This yields N_q number (equal to the number of queries in each branch) of predictions. N_q is chosen in such a way that it is always greater than the number of ground truths per frame. We pad ground truths with ϕ (no relationship triplet) so that it is possible to have the ground truth set $\mathbf{G}_t$ with N_q number of elements. One important detail to note here is that there are three kinds of ground truth bounding boxes: subject bounding boxes, object bounding boxes, and relation regions. Ground truth relation regions refer

to the union bounding boxes between the subject and the object bounding boxes that have relations, and are only used during the training phase. Next, each element in $\mathbf{P}_t$ is matched with an element from the ground truth set, $\mathbf{G}_t$. The matching cost metrics is, $\mathbf{C} \in \mathbb{R}^{[N_q \times N_q]}$. Any element (i, j) in this metrics refers to the cost to match i^{th} element from $\mathbf{P}_t$ with j^{th} element from $\mathbf{G}_t$ and defined as,

$$\mathbf{C}^{(i,j)} = \eta_b(\mathbf{C}_{sb}^{(i,j)} + \mathbf{C}_{ob}^{(i,j)} + \mathbf{C}_{rb}^{(i,j)}) + \eta_s\mathbf{C}_{s}^{(i,j)} + \eta_o\mathbf{C}_{o}^{(i,j)} + \eta_r\mathbf{C}_{r}^{(i,j)} \tag{4.7}$$

$\mathbf{C}_{sb}^{(i,j)}, \mathbf{C}_{ob}^{(i,j)}, \mathbf{C}_{rb}^{(i,j)}$ are the subject bounding box, the object bounding box and the relation region matching costs, $\mathbf{C}_{s}^{(i,j)}$ is the subject label matching cost, $\mathbf{C}_{o}^{(i,j)}$ is the object label matching cost and $\mathbf{C}_{r}^{(i,j)}$ is the relation label matching cost between i^{th} element from $\mathbf{P}_t$ and j^{th} element from $\mathbf{G}_t$. These costs are calculated following [68]. $\eta_b, \eta_s, \eta_o, \eta_r$ are fixed hyper-parameters. The Hungarian matching algorithm [74] is used to find the optimal matching between the predictions and the ground truths by using these cost metrics. After this matching, every prediction is associated with a ground truth. Next, the following loss is calculated for training the network:

$$\mathcal{L} = \lambda_g\mathcal{L}_{GIOU} + \lambda_l\mathcal{L}_{L1} + \lambda_s\mathcal{L}_{sub} + \lambda_o\mathcal{L}_{obj} + \lambda_r\mathcal{L}_{rel} \tag{4.8}$$

Here, $\mathcal{L}_{GIOU}$ and $\mathcal{L}_{L1}$ are the generalized intersection over union (gIOU) and L1 box regression losses for the predicted subject bounding boxes, object bounding boxes, and relation regions. $\mathcal{L}_{sub}$ and $\mathcal{L}_{obj}$ are the cross-entropy losses for subject and object label predictions. $\mathcal{L}_{rel}$ is the binary cross-entropy loss for the relationship label predictions. λ_s, λ_o, λ_g, λ_l, and λ_r are the corresponding hyper-parameters.

Notice that a portion of the datasets [6, 92] fixes the subject as humans. In this case, the subject prediction vectors and subject bounding boxes are not used for loss

Attention	**Spatial**	**Contact**
Looking at	On the side of	Standing on
Not looking at	Above	Touching
Unsure	Beneath	Wearing
	In	Writing on
	In front of	Carrying
	Behind	Covered by
		Have it on the Back
		Lying on
		Eating
		Holding
		Wiping
		Twisting
		Not contacting
		Leaning on
		Drinking from
		Sitting on

Table 4.1: Different types of relationships in the AG dataset. In total, it has 25 relationships divided into three types.

Dataset Name	Number of Seen Triplets	Number of Unseen Triplets	Number of Training frames	Number of Test frames
Action genome (AG) [92]	499	0	166,785	54,371
Action genome* (AG*) [92]	421	80	146, 517	54, 371
HICO-Det [6]	480	120	37,328	9,552
UnRel [2]	4015	65	4000	1000

Table 4.2: A summary of the datasets used for evaluating DDS. * refers to the new training split created by us to test DSG generation models in the compositional setting.

calculation.

4.4 Experiments

4.4.1 Experimental Setup

We evaluate DDS's performance in Action Genome (AG) [92] dataset. Moreover, we show our model's performance in SSG generation datasets: HICO-DET [6] and UnRel [2]. All the datasets provide relationship triplet annotations with subject and object bounding

boxes. SSG generation datasets only contain images, therefore, each sample is treated as a single-frame video. Next, the used datasets will be presented in more detail:

Action Genome (AG) [92]: This dataset is built on top of the Charades [118] dataset, provides frame-level annotations, and is extensively used in the literature for DSG generation. It has 36 distinct object classes and 25 relationship classes. The object classes are common household items such as doors, windows, and cups, and have a total (train and test set) of $476,229$ bounding boxes. The relationship classes are divided into 3 distinct sub-types:

- *Attention* relationships indicate how the subjects are looking at the objects.
- *Spatial* relationships present the spatial layout of the different interacting objects.
- *Contacting* relationships report the manipulation of the objects by the subjects.

Table 4.1 is showing different types of relationships in the AG dataset. In total, AG provides $1,715,568$ instances of the mentioned relationships contained in $135,484$ subject-object pairs. Every subject-object pair can have multiple relations. Also, on AG the subject class is always human.

Originally, the AG dataset provided $7,464$ videos with $166,785$ frames in the training set and $1,737$ videos with $54,371$ frames in the test set. The original training set contains 530 relationship triplets. All these relationship triplets are present in the test set. We refer to this setting as the fully-supervised setting. The main interest in this work is to evaluate DDS's performance in the compositional setting. To this end, we have generated a new training data split. While this split retains all individual relationships and objects, it has fewer relationship-object triplets compared to the original training set. In this way, our test set now has relationship triplets that are not present in the training set.

To create this split, we calculate the occurrence score D_i for each object-relationship

pair, P_i. D_i is defined as,

$$\mathbf{D}_i = \frac{NF_i}{TF_i} \tag{4.9}$$

Here, NF_i is the number of frames in the training set containing P_i and TF_i is the number of total frames in the training set.

Two object-relationship pairs (P_i and P_j) are defined as connected if they appear together in any of the video. We select the pair P_i with the maximum D_i value in the training set and subsequently include all the pairs connected to it. This process continues until our training split encompasses all distinct objects and relationships.

This new proposed training set contains $6,784$ videos with $146,517$ frames containing 421 relationship triplets. The original test set is not changed. It contains 499 object-relationship, where 80 of them are not present in our new training set.

HICO-Det [6]: This dataset has 80 objects and 117 relationship classes. The relationships are limited to interactions such as holding, working, etc. In the literature, this dataset is used for HOI detection (described in the chapters 2 and 3). Recently, a RF (Rare First) protocol is proposed by [102] for this dataset to evaluate SSG generation performance under compositional setting. Under this protocol, a new training split has been created by not selecting rare interaction classes. An interaction class is called rare if it has less than 10 samples in the HICO-Det's original training set. This protocol has $37,328$ images in the training set with 480 relationship triplets. The test set has $9,552$ images with 480 seen relationship triplets and 120 unseen relationship triplets.

UnRel [2]: This dataset provides extremely unusual SSG triplets, for example: ($\langle elephant, bike, riding \rangle$). It has 4000 training and 1000 test images with 100 objects and 70 relationships. The original train/test split provided by the authors already provides a compositional setting where the test set has 65 unseen relationship triplets. The training set contains 6672 seen relationship triplets. Every image in the training set

contains multiple relationship triplets. On average, every image has 7.5 relationships.

A summary of all the datasets is shown in Table 4.2.

4.4.2 Evaluation Metrics

Following existing works [12, 90–92], we report our performances in AG dataset with Recall@K metric. Here, $K = [20, 50]$. We utilize the most challenging SGDet [92] protocol to report our performances. In this protocol, the network needs to detect relationship triplets along with subject and object bounding boxes. There can be multiple relationship triplets between a subject-object pair. Moreover, mAP (mean average precision) was selected to report the performances in UnRel and HICO-Det datasets similar to current works [2, 67, 73, 116]. Here, performances are reported in three categories: unseen (only unseen relationship triplets), seen (only seen relationship triplets), and full (all relationship triplets) [102].

For all datasets, a prediction triplet from DDS is considered correct if subject and object bounding boxes have at least 0.5 Intersection over Union (IoU) with ground truth bounding boxes and subject, object, and relationship labels match with ground truth labels.

4.4.3 Implementation details

ResNet-50 [45] is used as the CNN backbone. Both temporal decoders inside the spatio-temporal decoders have a single layer. We follow Qpic's [68] setup for the encoders. Each encoder has 6 layers. We select 6 layers for the object decoder with 3 layers for the relation decoder. All loss coefficients in equation 4.7 and equation 4.8 are set as [68]. The number of queries in each branch is 64. Each query is a vector of size 256. The model is trained with AdamW [119] optimizer. We initialize the parameters of DDS from

DETR [74] trained on COCO [47] object detection dataset. The initial learning rate for the backbone network is 10^{-6} and for the other part of the network is 10^{-5}.

When training in the AG dataset, we drop the learning rate by 10 times at every 40 epochs and utilize a batch size of 128. DDS processes each video frame from a single video sequentially. We utilize scale augmentation like [74]. Input frames are resized with the shortest side being at least 480 and at most 800 pixels, and longest side is at most 800.

In the other datasets [2, 6], the learning rate is dropped by a factor of 10 at every 60 epochs with a batch size of 32. We use a scale augmentation scheme similar to the one used for AG, except that the longest side of the resized image is chosen as 1333. In AG dataset, the model is trained for 50 epochs and in the other datasets models are trained for 100 epochs. We resize the frames from the AG dataset to dimensions smaller than the resized frames from the SSG generation datasets. This adjustment ensures a more manageable training time for the AG dataset.

4.4.4 Data Processing and Training

DDS is trained in 8 A100 Nvidia GPUs using pytorch's distributed data parallel (DDP) framework [120]. AG dataset has different length of videos. While doing sequential processing of the input data this characteristic of the AG dataset creates a unique challenge for utilizing the DDP framework. We propose a novel data loading strategy to solve the challenge.

Consider a dataset of 9 videos processed by 2 GPUs with batch size of 2, as depicted in the Figure 4.4. In DDP, each GPU receives an independent model copy and a data subset. As illustrated, GPU 1 and 2 obtain videos of lengths 2 and 6 in one mini-batch, respectively. After 2 frames, GPU 1 has no more frames to process because DDP doesn't

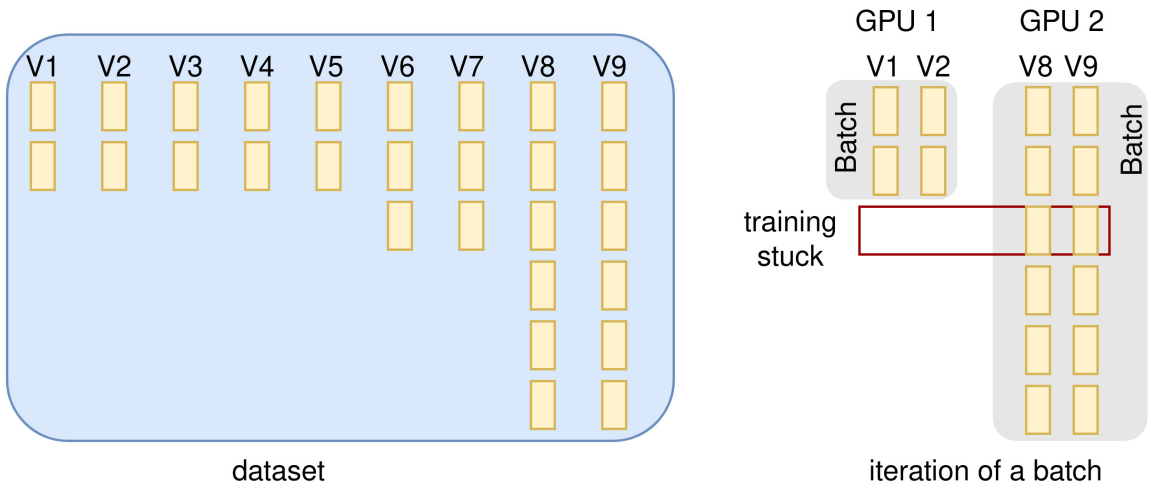

Figure 4.4: *Using Pytorch's distributed data parallel (DDP) for processing dataset with videos of varying lengths presents a challenge. A situation like this is depicted here. Gradients are accumulated after each iteration across the GPUs. If a mini-batch contains videos of different lengths, one GPU, as shown with GPU 1 in the figure, might have no frames to process. This causes the training to stall; pytorch waits indefinitely for the gradient from the idle GPU.*

Dataset Name	Per epoch training time in closest minutes
AG [92]	29
HICO-Det [6]	7
UNRel [2]	1

Table 4.3: Training time of DDS in different datasets.

load new data until the mini-batch is complete. Consequently, GPU 1 won't return any gradient during the accumulation step, causing the program to stall without an error message. We can solve the described stalling in training by using common techniques like padding data or sampling data. We define these techniques below,

- **Padding data:** We pad each video in the dataset to match the frame count of the longest video. Padding can be done using a black frame or by repeating the last frame. This method is extremely wasteful specially if the difference between the smallest and largest size videos in the dataset is large.

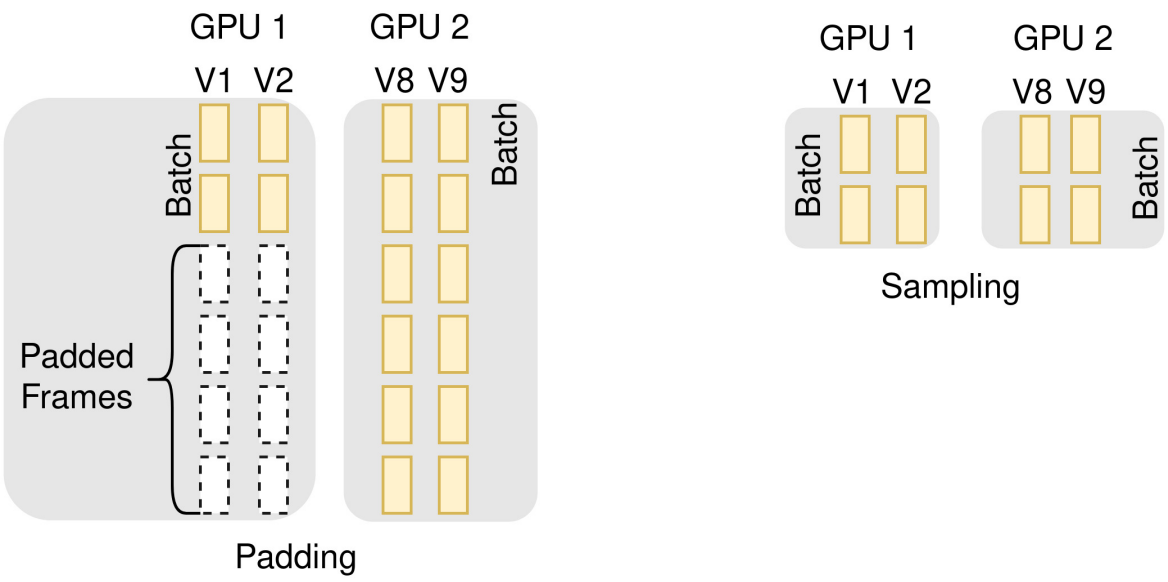

Figure 4.5: *Padding and sampling dataloading for dataset with videos of different lengths. Each yellow block represents a video frame. Each white block is a padded frame.*

- **Sampling data:** We sample each video in the dataset to have the same frame count as the average length video. We remove the videos with less frame count than the average length from the dataset. This may degrade the performance of the model as huge amount of data is being discarded.

These techniques are visualized in Figure 4.5. However, our analysis in section 4.5.2 indicates that these methods are either inefficient (padding) or imprecise (sampling). Therefore, we propose a novel method for dataloading called **Block Load - BLoad**. Figure 4.6 shows our method.

BLoad innovatively extends the traditional padding strategy. Instead of simply padding a video sequence with zeros or repeating its last frame, we introduce a more dynamic approach. For reference, please see our algorithm 4.1. We create blocks sized to the largest video, T_{max}. A block, B, might contain multiple videos, but its total frame count will always match T_{max}. We iteratively fill B with videos. If a video of the needed

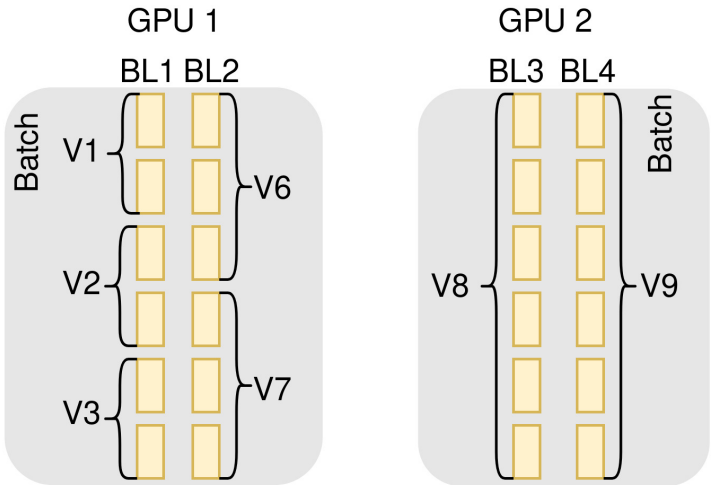

Figure 4.6: *Proposed BLoad strategy. We create block of videos. Each block has number of frames equal to the largest video in the dataset. Each yellow block represents a video frame.*

length is absent, we use padding.

To ensure accuracy in processing, we maintain a record of the starting index for each video within the block, referred as reset_table in algorithm 4.1. This allows us to correctly relay previous frame embeddings to our spatio-temporal decoders. BLoad strategy is only used for training in the AG dataset as other datasets used to evaluate DDS are SSG generation dataset.

Training time for different datasets can be seen in Table 4.3. As expected DDS requires most time to train in the AG dataset.

4.5 Results & Analysis

In this section, we first compare DDS's performance with the SOTA models in Section 4.5.1. Next, a detailed study on the impact of different components of our network is presented in Section 4.5.2. Finally, in Section 4.5.3 qualitative results from our model are provided.

Algorithm 4.1: Our dataloading algorithm - BLoad.

```
// Initialize a dictionary L_dict with video lengths as keys and
   video ids as values
L_dict ← Dictionary();
new_dataset ← [];
reset_table ← [];
while L_dict is not empty do
    remaining_frames ← T_max;
    block ← [];
    block_reset ← [];
    while remaining_frames ≥ min(keys(L_dict)) do
        sampled_sequence ← Random(L_dict);
        block.append(sampled_sequence);
        remaining_frames −= len(sampled_sequence);
        block_reset.append(T_max − remaining_frames);
    end
    Pad(block, remaining_frames);
    new_dataset.append(block);
    reset_table.append(block_reset);
end
```

4.5.1 Comparison with the SOTA models

In the AG [92] dataset, we report DDS's performances in Table 4.4 under the compositional setting. In this setting, there are $\sim 12\%$ less training data with 80 unseen relationship triplets. We retrain the SOTA model STTRan [12] in the mentioned setting for comparison. It is important to note here, among the three recent DSG generation models [12, 90, 91], only STTran's code is publicly available, therefore limiting the capacity to evaluate other models. DDS outperforms STTran in all recall levels in both unseen and seen relationship triplet detection. Especially, for detecting unseen triplets DDS achieves $4 - 24$ times improvement over the SOTA model. It shows the generalization power of DDS.

Additionally, for a fair comparison with other models, we train DDS in the full training

	Seen		Unseen	
	R@20	R@50	R@20	R@50
STTran [12]*	33.7	36.6	0.3	4.4
DDS (Ours)	**41.8**	**48.8**	**7.4**	**18.2**

Table 4.4: DDS's performance comparison in AG test set under the compositional setting. Both reported models are trained on the proposed small-size training set under the compositional setting. * means the model was trained using publicly available code. Among recent DSG generation models, only STTran's [12] code is publicly available. The best results are shown in **bold**.

Method	Backbone	R@20	R@50
GPNN [43]	ResNet-101	33.3	42.6
PPDM [85]	Hourglass-104	34.1	43.5
VRD [121]	ResNet-101	22.0	32.7
IMP [18]	ResNet-101	34.4	43.7
MSDN [122]	ResNet-101	34.7	43.8
Graph RCNN [123]	ResNet-101	35.0	44.1
RelDN [95, 124]	ResNet-101	35.1	44.9
VCTree [95]	ResNet-101	35.0	44.3
STTran [12]	ResNet-101	36.2	48.8
HORT [90]	ResNet-101	36.5	46.7
DSGAP [91]	ResNet-101	37.9	50.1
DDS (Ours)	ResNet-50	43.3	51.5
DDS (Ours)	ResNet-101	**46.3**	**54.4**

Table 4.5: DDS's performance comparison in AG test set. Here, like other models, DDS is trained in the full training set of AG dataset. The best results are shown in **bold**. For the other models, all the reported numbers are taken from the original publications.

set of AG dataset and report performance in Table 4.5. Here, all methods up to VCTree [95] are SSG generation methods implemented by [12]. As expected, with similar training data like other models in Table 4.5, DDS achieves SOTA performance on all metrics.

In Table 4.6 and Table 4.7, DDS outperforms all existing methods in HICO-Det [6] and UnRel [2] datasets. Among these methods, [7] and [117] follow a Transformer based encoder-decoder architecture. The proposed network shows superior performance than both of the models. In summary, DDS outperforms existing works in both seen and unseen SSG generation (full category) in HICO-Det by 5% and in UnRel by 33%.

Method	Unseen (mAP)	Seen (mAP)	Full (mAP)
VCL [102]	10.1	24.3	21.4
ATL [116]	9.2	24.7	21.6
FCL [67]	13.2	24.2	22.0
THID [7]	15.5	24.3	23.0
SCL [117]	19.1	30.4	28.1
DDS (Ours)	**21.1**	**31.7**	**29.6**

Table 4.6: DDS's performance comparison in HICO-Det test set under RF (Rare–First) compositional setting. The best results are shown in **bold**.

Method	Unseen (mAP)	Seen (mAP)	Full (mAP)
VRD [121]	-	-	7.2
WSL [2]	-	-	9.9
DUV [73]	-	-	13.4
DDS (Ours)	**16.3**	**27.4**	**17.9**

Table 4.7: DDS's performance comparison in UnRel test set. The best results are shown in **bold**. Not reported results are marked with -.

4.5.2 Ablation Studies

We perform ablations for different design choices of our network in this section. For these ablations, we report our performances in the HICO-Det [6] dataset, except for data loading ablation, where we report the performances in the AG [92] dataset.

Multi-branch Design: We first validate our decoupled multi-branch design. The performances are reported in Table 4.8. The base network has one single branch where a single set of queries is used for both relationship and object detection. In row 1 of Table 4.8 the performance is pretty poor for the single branch base network, especially in the unseen category. Next, a multi-branch network is created by using two spatio-temporal decoders. Both decoders get the same encoded features. However, the relation branch in this case doesn't do relation region prediction. A performance improvement is observed compared to the single branch for this design. Similarly, with the gradual introduction of two encoders and relation region prediction, the performance keeps increasing. In particular, all these components yield a significant improvement in the unseen category

Type	Relation Region	Separate Encoders	Separate Spatio-temporal Decoders	Unseen (mAP)	Seen (mAP)
Single branch (base network)	✗	✗	✗	17.9	29.9
Multi branch	✗	✗	✓	18.7	30.5
	✗	✓	✓	19.7	31.6
	✓	✓	✓	**21.1**	**31.7**

Table 4.8: Impact of different components on our decoupled multi-branch design.

	Unseen (mAP)	Seen (mAP)
DDS with o to r	19.6	30.6
DDS with r to o	19.4	28.8
DDS with separate queries	**21.1**	**31.7**

Table 4.9: Different kinds of query sharing strategy. o to r refers to object to relation branch query sharing. r to o refers to relation to object branch query sharing.

compared to the seen category and thus show the impact of decoupled learning on unseen relationship triplet predictions. We also do qualitative analysis between DDS and the base network described in the section 4.5.3.

Relation Region Ground Truths: As noted in Section 4.3.6 the relation branch only produces relation region prediction during training. Ground truth relation region is required for the training. However, ground truth relation regions are not strictly defined by the provided annotations such as subject and object bounding boxes. Therefore, we experiment with different settings to generate ground truth relation regions for subject and object pairs that have relationships:

- Mixture: In this case, the intersection of the subject, and object bounding boxes is selected as the relation region if the boxes have an IoU greater than the θ. For any other case, the relation region is defined as their union boxes. Different values of θ are tested.

- Union box: Here, the union of the subject and object bounding boxes are defined as the relation regions.

Relation Region	Sub-Obj IoU, θ	Unseen (mAP)	Seen (mAP)
Mixture	0	19.2	30.3
	0.1	18.0	30.4
	0.5	19.4	30.8
Union Box	-	**21.1**	**31.7**

Table 4.10: Different kinds of relation region ground truths. Sub-Obj IoU refers to the IoU between the subject and the object bounding boxes.

Object Spatial Decoder	Relation Spatial Decoder	Unseen (mAP)	Seen (mAP)
3	3	20.2	31.4
3	6	20.8	**31.8**
6	3	**21.1**	31.7
6	6	18.0	31.3

Table 4.11: Different number of layers in the object spatial decoder and the relation spatial decoder.

The performances of the network with different relation region ground truths are shown in Table 4.10. With union boxes, DDS performs best (last row). This may be due to the fact that the union box guarantees the inclusion of the spatial location of the relation and therefore it can be very helpful in detecting non-contact relationship triplets (e.g. subject looks at object etc.).

Share Queries: DDS utilizes two different sets of queries for the relation and the object branches. We also test DDS's performance by sharing the queries between the branches with two different strategies:

- DDS with o to r: In this case, the output of the object spatio-temporal decoder is fed as input relation queries to the relation spatio-temporal decoder.
- DDS with r to o: In this case, the output of the relation spatio-temporal decoder is fed as input object queries to the object spatio-temporal decoder.

The performances are reported in Table 4.9. Without sharing the queries DDS performs

Number of Queries, N_q	Unseen (mAP)	Seen (mAP)
32	19.8	30.5
64	**21.1**	**31.7**
100	19.7	31.3

Table 4.12: DDS's performance with different number of queries, N_q.

	Padding	Sampling	BLoad
Number of frames padded	534,831	0	3695
Number of frames discarded	0	92,271	0
Time per epoch in minutes	170	18	29
Performance (recall @20)	-	39.2	43.3

Table 4.13: Comparison of various data processing schemes in the AG dataset. - indicates that performance was not assessed due to excessive computational costs.

the best, especially in unseen categories. This matches our hypothesis on the importance of decoupled learning by utilizing two different sets of queries. We also notice a significant performance drop even in seen classes when we share queries from the relation to the object branch (2nd row). This shows object queries have more generalization ability than the relation queries.

Spatial Decoders: We test with different numbers of layers for spatial decoders. The result is shown in Table 4.11. DDS performs poorly in extreme cases (first and last row). This is expected as DDS is getting underfitted due to the decrease in the number of layers (first row) and overfitted as the number of layers increases (last row) in the relation branch. Also, the best performance in the unseen category arises when the object decoder has more layers than the relation decoder. Given that the object spatial decoder decodes two entities (subject, object), it is reasonable to require more layers than the relation spatial decoder.

Number Of Queries: We also test with different number of queries, N_q. Our design ensures equal number of queries in both our branches. For example, if the object branch has $N_q = 32$ queries then the relation branch will also have $N_q = 32$ queries. The

performance comparison can be seen in Table 4.12. In both seen, unseen category the best performance is reached with 64 number of queries. From that point, we can see the performance decreases in both direction.

Data Loading: As described in section 4.4.4, loading data efficiently from a dataset (e.g. AG) with varying length of videos poses a significant challenge. Common solutions such as padding, sampling suffer from inefficiencies and inaccuracies. To solve the problem, we propose our own dataloading strategy BLoad where we create blocks of videos equal to the largest length video in the dataset. Please see 4.4.4 for reference.

To emphasise the differences and potential advantages of BLoad, we present a comparative analysis in Table 4.13. As expected when every video is padded to match the length of the dataset's largest video, it leads to an excessively redundant training set. In particular, the AG dataset has a total of $166,785$ frames. The padding strategy augmented this count by almost fivefold, primarily due to the predominance of shorter videos within AG. The sheer computational intensity required by padding made it impractical to even complete the training.

The sampling strategy makes each video in the dataset to have the average video length. In this scheme, videos shorter than the average are discarded. While this expedited the training process, it compromised performance.

Our BLoad method strikes an elegant balance by neither discarding any frames nor resorting to excessive padding. This inclusive approach, which ensures every frame is taken into account, has shown better performance compared to the more exclusionary sampling method.

In summary, BLoad has shown superior performance in terms of efficiency and accuracy compared to other data loading schemes.

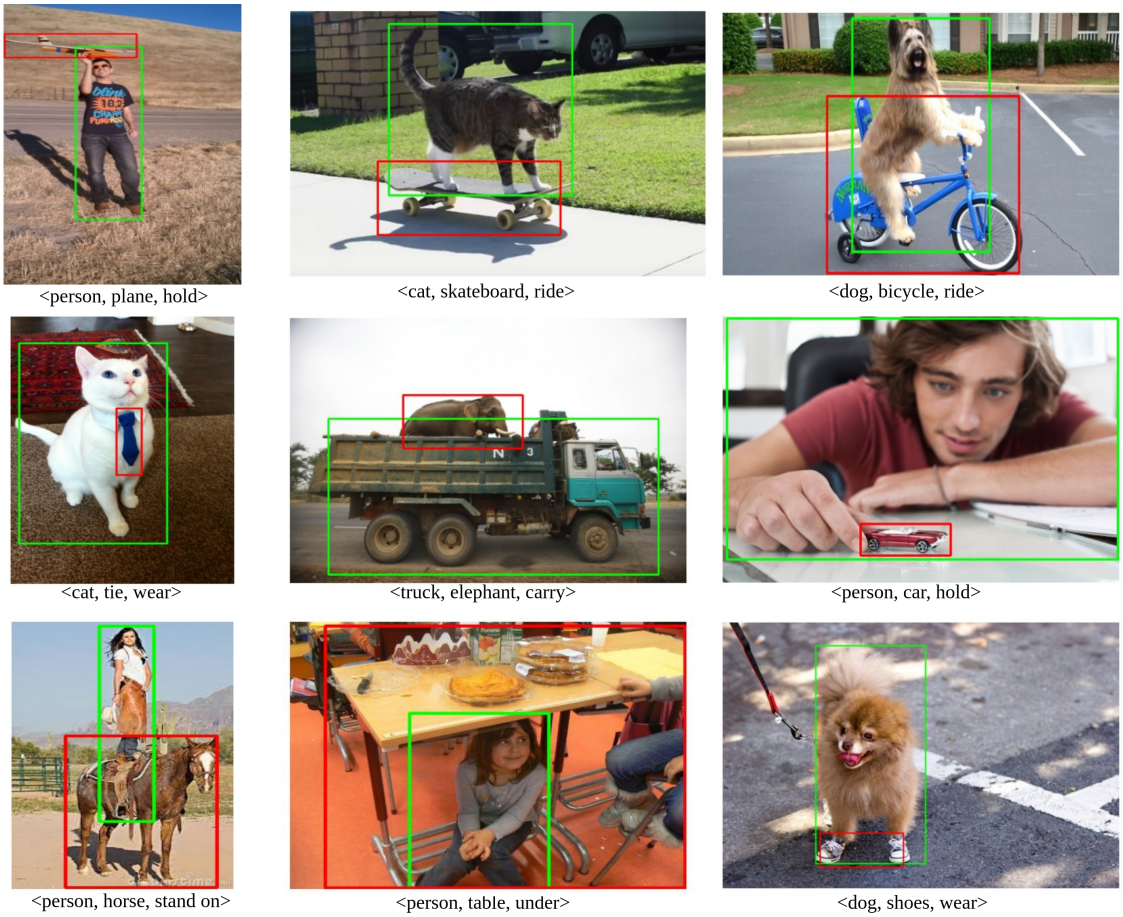

Figure 4.7: Qualitative results of DDS for predicting unusual relationship triplets in UnRel [2] dataset. The subject bounding box is green and the object bounding box is red. *Our base network (single branch) fails to detect these marked triplets.* For both networks, we utilize top-20 predictions per sample.

4.5.3 Qualitative Results

This section qualitatively analyze DDS's performance.

We first show DDS's performance in the UnRel dataset. This dataset has very unusual unseen relationship triplets. In this dataset, we compare DDS with our base network, a single branch network with one encoder, and one spatio-temporal decoder (details in section 4.5.2). Fig. 4.7 illustrates the samples where DDS is successful; however, our base network completely misses the relationship triplets (predicted bounding boxes do

Figure 4.8: Qualitative results of DDS for predicting unseen relationship triplets in HICO-Det [6] dataset. The subject bounding box is green and the object bounding box is red. *SOTA network THID [7] fails to detect these marked triplets.* For both networks, we utilize top-20 predictions per sample.

not match with ground truth). We utilize the most confident 20 predictions for each sample from both networks for this visualization.

Next, we visualize DDS's performance in the HICO-Det [6] dataset in Figure 4.8. Here, we show examples where DDS's prediction matches with the ground truth, however, SOTA network THID [7] does not make correct predictions. THID [7] is an end-to-end Transformer based one stage network. It utilizes large-scale pretrained vision-langage model CLIP [112] to distill knowledge for unseen relationship triplet detection. DDS clearly outperforms THID for detecting unseen relationship triplets.

We also evaluate DDS on custom samples in Fig. 4.9. These samples contain common

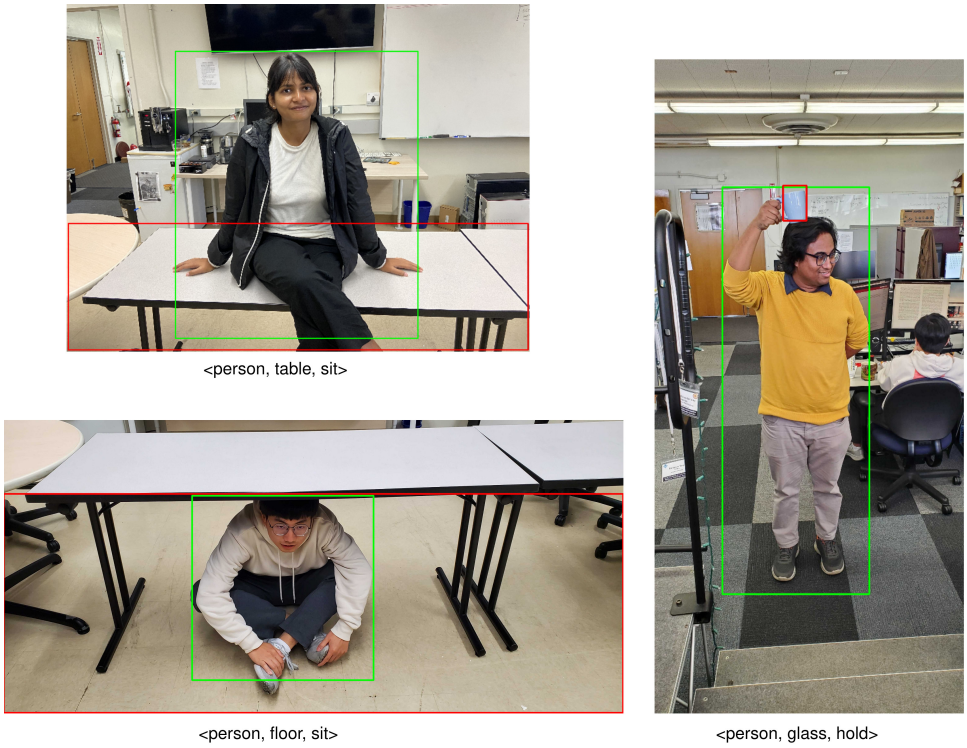

Figure 4.9: Performance of DDS on custom samples. The subject bounding box is green and the object bounding box is red. For these samples we only utilize the top most prediction of DDS.

relationships with unusual contexts. DDS predicts the correct relationship triplets in all the samples.

We compare the attention maps from DDS and the base network to further analyze our improved performances. Fig. 4.10 presents attention maps for the samples where both DDS and the base network have correct predictions. The attention maps are of the queries that predict the marked subject and object bounding boxes from the last layer of the spatio-temporal decoders. We overlap attention maps from both our spatio-temporal decoders to get the final attention map. As can be seen, although both networks have correct predictions, DDS's attention maps cover the correct spatial region. In contrast, the base network with only one spatio-temporal decoder has attention on a very random

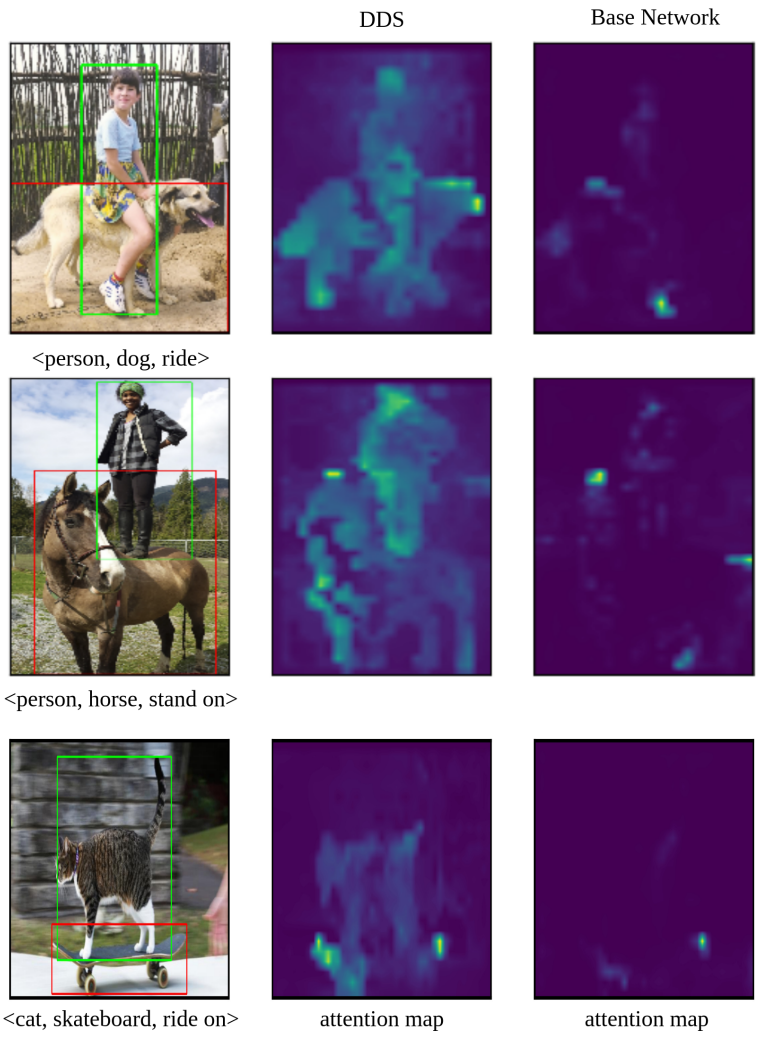

Figure 4.10: Performance analysis of DDS over the base network. The subject bounding box is green and the object bounding box is red. The attention maps are visualized from the last layer of the spatio-temporal decoder.

portion of the object and the subject.

4.6 Discussion

This work proposes a multi-branch decoupled network for DSG generation. The DDS network is comprised of two encoder-decoder based Transformer branches. This design enables independent learning of objects and relationships, thus enabling DDS to detect unseen relationship triplets. The effectiveness of DDS is demonstrated through extensive experiments with DDS achieving SOTA performance on three benchmark datasets. Moreover, the conducted ablation studies have provided the motivation and significance for different components of DDS. However, while successful compared to the existing works, the quantitative results (Table 4.5, 4.6, 4.7) show room for improvement in detecting unseen relationship triplets. Future research will focus on improving DDS for a better generalized DSG generation.

Chapter 5

Learning Object-Attribute Compositions Using Localization

5.1 Introduction

Human visual reasoning allows us to leverage prior visual experience to recognize previously unseen Object-Attribute (O-A) compositions. O-A compositions can play crucial role in detecting actions as exaplined in the previous chapter 1. In this Chapter, we focus on correctly associating attributes and objects specially in clutter scenes.

Predicting novel O-A compositions – referred to as Composition Zero Shot Learning (CZSL) [8,9,13,16,28,113,125,126]–is an active area of research. There has been significant progress on CZSL methods in recent years, however, as our experiments demonstrate, their performance degrades in natural cluttered scenes, as illustrated in Fig. 5.1. The main reason in these cases is the interference from the other potential confusing elements. For example, in Fig. 5.1(B.1), the SOTA methods are not able to detect the object of interest given its size relative to image; and while the bird is the object of interest in Fig. 5.1(B.2), the surrounding context dominated by the green leaves results in an incorrect

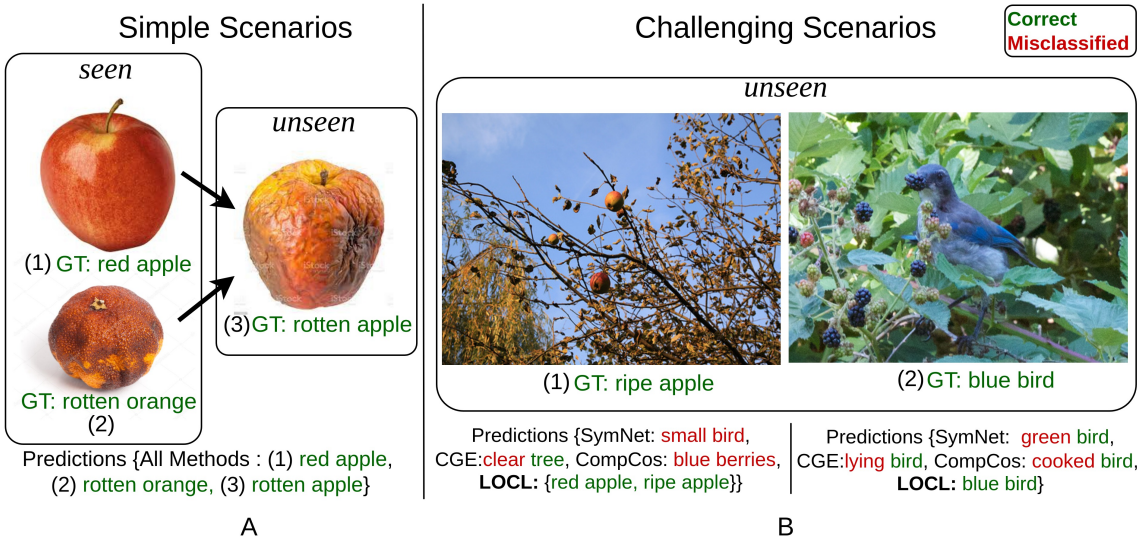

Figure 5.1: The object of interest shown in images A.1, A.2, and A.3 presents simple scenarios where all SOTA (SymNet [8], CGE [9], CompCos [10]) methods make correct O-A associations. However, for the same object (*apple*) in a more cluttered scene in image B.1, these methods fail. Even in cases where there is a dominant object of interest, such as a *bird* in (B.2), where there is significant background clutter, most of the SOTA methods have incorrect O-A associations.

association of the color attribute to the object.

The poor performance of the SOTA methods can be attributed to the dominant confounding elements thereby impeding the right O-A composition prediction. This in turn is due to the bias towards seen O-A composition during training time. Generalization to more realistic cases as seen in Fig. 5.1(B) is crucial for robust systems.

Inspired by these limitations, we propose Learning Object-Attribute Composition using Localization (LOCL). Our model (LOCL) leverages spatially-localized learning, which is not present in the existing CZSL networks. It is reasonable to ask *Why not first localize the objects and then associate the attributes?* In principle, this can be done, however, the SOTA methods for object detection and localization use extensive datasets for their training. Hence it will not be possible to meaningfully test the CZSL with pre-trained detectors. The images shown in Fig. 5.1 are from existing datasets for CZSL

methods [9,14,15]. *We note that all the experiments reported in this work use the datasets that are created for evaluating CZSL approaches.*

Existing SOTA O-A detection approaches do not take into account the possibility of scene attributes confounding with correct O-A composition prediction [8, 9, 125]. These methods are designed to work with wholistic image features [13, 16, 28, 113, 126]. Some recent work address this issue by partitioning the image into regions [58, 127] or equal-size grid cells [128–130], but they are not very effective in capturing distinctive object features.

Our approach towards better generalization of CZSL to more challenging images (Fig. 5.1.B) with background clutter is to leverage localized feature extraction in O-A composition. Specifically, we adopt a two step approach. First, a Localized Feature Extractor (LFE) associates an object with its attribute by reducing the interference arising from additional attribute-related visual cues occurring elsewhere in the image. The CZSL benchmark datasets *do not* contain any localization information. As noted before, off-the-shelf object detectors can be inadvertently exposed [25] to unseen OA compositions. Therefore, we developed a weakly supervised method for localized feature extraction. Second, the composition classifier uses the localized distinctive visual features to predict an O-A pair.

The proposed LOCL outperforms competing CZSL methods on *all* existing datasets – including the more challenging CGQA – providing a strong evidence in favor of its applicability to more realistic scenes. Further, the performance of all existing methods improve when our localized feature extractor is included as a pre-processing module – although LOCL still outperforms these methods.

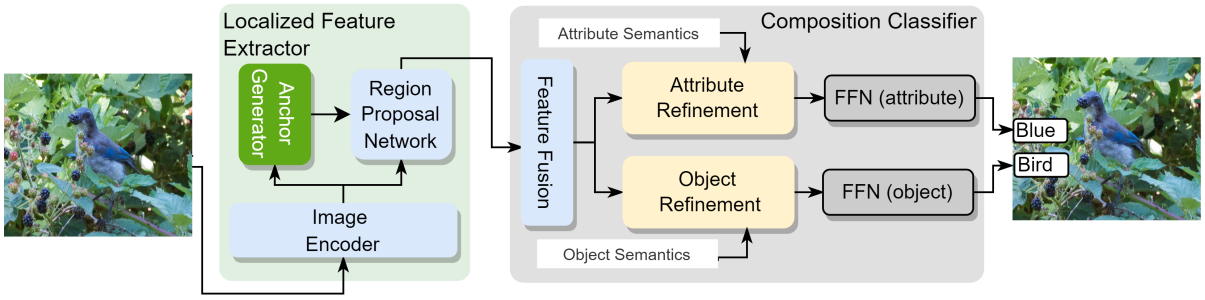

Figure 5.2: LOCL architecture. The Localized Feature Extractor (Section: 5.3.1) generates proposals that are likely to contain objects. These proposals are refined with the object and attribute semantics using Composition Classifier (Section: 5.3.2).

5.2 Related Work

Existing works in object attribute (O-A) CZSL task typically assume that the images of the object of interest present in uncluttered context. This assumption is also true for the initial CZSL datasets [14, 15]. As a result, most of the CZSL methods [9, 13, 16, 28, 29, 113, 125, 126, 131, 132] perform quite well on uncluttered scenes. As noted before, some of the methods reduce the interference from confounding elements by partitioning the image. Among these works, [129] is designed for datasets with a dominant object with a clear background. On the other hand, [127] partitions the image to equal-sized grid cells and relies on aligning the attribute semantic and visual features. However, our analysis has shown these works are not fully effective in the cluttered scenes.

In many other recent works, the O-A problem is considered as a matching problem in a latent space [13, 28, 113, 131]. For this matching task, [113] proposes a modular network with a dynamic gating whereas [13] defines objects and attributes using a support vector machine (SVM). On the other hand, [8, 132] consider attributes as functional operation over objects. More recently [9, 16, 29, 125, 126] focus on the relationships among attributes and objects. [29] disentangle attributes and objects with metric based learning. [9]

learns attribute-object dependence in a semi-supervised graph structure where unseen combinations are considered connected during training. This is extended in [16, 26, 125] where all possible combination of objects and attributes are considered during inference.

In general, the performance of current methods drop significantly on images with background clutter. Taking inspiration from the generic pipeline of weakly supervised object detection (WSOD) [133–139], LOCL consists of a region proposal network with pseudo-label generation module that leverages supervision from linguistic embeddings in a novel contrastive framework. This feature extraction can be utilized to improve the existing network's performance in images with background clutter as shown in Table 2.4.

5.3 Approach

The primary issue to be addressed is the scene complexity, that require that the methods are able to make the correct associations during the training phase and predict the unseen configuration during testing. The proposed method is intuitive and straightforward in creating a weakly supervised framework that is modular and generalizes well. The LOCL extracts localized features of object regions in the image, which allows it to learn useful OA relationships and also suppress spurious O-A relations from the background clutter.

First, we pre-train a Localized Feature Extractor $LFE(.)$ (Sec. 5.3.1) network to extract features from multiple regions of the image. Second, the pre-trained $LFE(.)$ along with a Composition Classifier $CC(.)$ (Sec. 5.3.2) network learns to detect the O-A composition. *The key insight is to leverage the features from regions containing the object of interest to learn accurate O-A associations.* Fig. 5.2 summarizes the overall LOCL architecture.

Problem Setting: Let $\{I, T_o, T_a, (a, o)\}$ be the training dataset with N samples, where I is the input image. T_o, T_a are the list of all object and attribute labels, respectively, and (a, o) is the tuple of attribute-object pairs in the image. The O-A pairs labels used during training are categorised as seen pairs. The goal of CZSL trained model is to take in an input image I and predict $(\hat{a}, \hat{o})$. The O-A pairs labels used during inference are novel and unseen. H*ere the seen and unseen object-attribute pairs are mutually exclusive.*

Proposed Network: The proposed LOCL is as $(\hat{a}, \hat{o}) = CC\ (\ LFE\ (I),\ T_a,\ T_o)$, where $LFE(.)$ and $CC(.)$ are trainable networks. LOCL is trained in two stages. In the first stage, given an image I, pre-training of $LFE(.)$ is done to generate multiple localized features. The details of the $LFE(.)$ module are discussed in Sec. 5.3.1. The output of the trained network $LFE(.)$ is a list of n features of object regions identified in the image.

In the second stage, out of these n features, r features ($r < n$) are input to the composition classifier (Sec. 5.3.2) to make the final prediction of attribute and object present in the image.

5.3.1 Pre-training Localized Feature Extractor($LFE(.)$)

Our Localized Feature Extractor network $LFE(.)$ is a combination of an image encoder (ResNet-50 [45]), text encoder, Region Proposal Network (RPN), and a pseudo label generator, as shown in Fig. 5.3. The RPN is inspired from F-RCNN [25]. It is trained from scratch using a contrastive learning framework. It generates proposals features for regions in the image that has high likelihood of object presence. The pseudo label generator creates labels to supervise the visual space that have high semantic similarity with ground truth O-A pair.

Given the input image I, the image encoder generates feature map $\mathbf{F} \in \mathbb{R}^{H' \times W' \times C}$

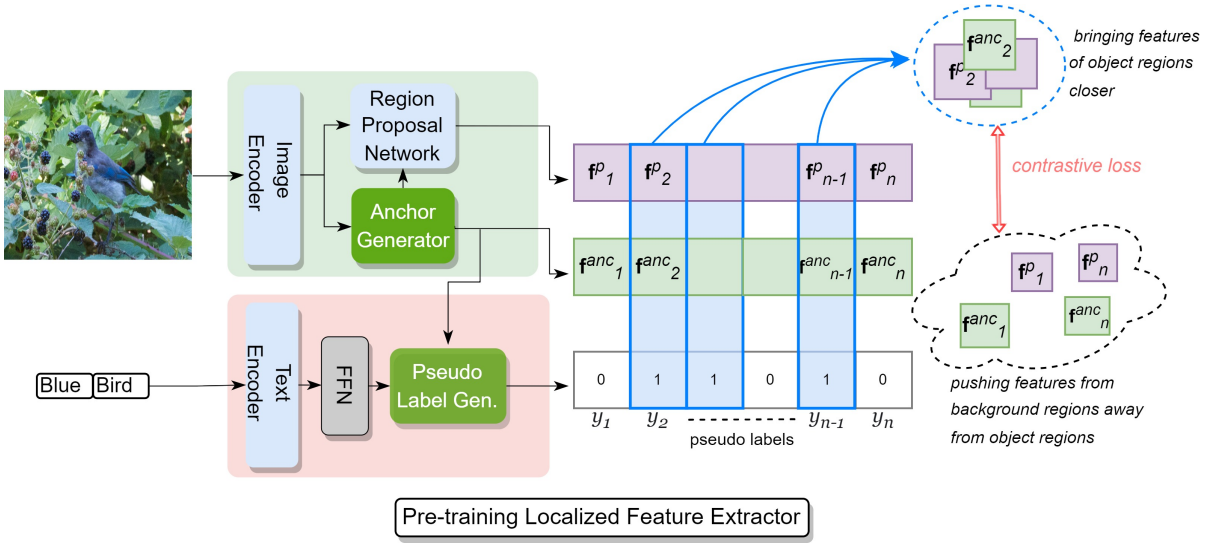

Figure 5.3: Summary of pre-training the localized feature extractor.The image encoder and region proposal are jointly trained to generate features of object of interest. During training time, we use text embeddings to generate pseudo labels to train the image encoder and region proposal using contrastive learning. At the test time, the learned image encoder and region proposal network are used to generate features from object regions.

where H', W' and C are the height, width and channel dimensions. Then n valid anchors (based on input image size, in our case for an image of 256×256, $n = 576$,) are generated on the input image [25]. Anchors are a set of rectangular boxes with different aspect ratio and scale generated at **each pixel of the input image** [25]. Corresponding to each anchor, a list of features is pooled from $\mathbf{F}$. The pooled anchor features are $[\mathbf{f}_1^{anc}, \mathbf{f}_2^{anc}, ..., \mathbf{f}_n^{anc}] \in \mathbb{R}^C$ shown as output of "Anchor Generator" in Fig. 5.3

The text encoder generates semantic pair embedding from the input text label (a : *Blue*, o : *Bird*) pair. With the help of these semantic pair embedding, we generate pseudo ground truth labels to train $LFE(.)$ network with weak supervision [140]. In the following, we refer to "pseudo ground truth labels" as "pseudo labels" for simplicity.

Pseudo Label Generator: The ground truth O-A semantic pair embeddings generated from text encoder are projected through fully connected layers (FFN) into a common subspace as visual embeddings. The output of FFN is denoted by $\mathbf{f}_{ao}^{text} \in \mathbb{R}^C$, where "$ao$" index is the ground truth (in our case one per image). Here the length C of semantic embedding equal to channel dimension C of visual feature vector $\mathbf{F}$. Now to generate pseudo labels, a cosine similarity score is computed between each $\mathbf{f}_k^{anc}$ and $\mathbf{f}_{ao}^{text}$.

$$\phi_k = \frac{\mathbf{f}_{ao}^{text} \cdot \mathbf{f}_k^{anc}}{||\mathbf{f}_{ao}^{text}||\ ||\mathbf{f}_k^{anc}||} \ \forall\ \mathbf{f}_k^{anc},\ where\ \ (\phi = [\phi_1, \phi_2, .., \phi_k, .., \phi_n]) \tag{5.1}$$

$$\mathbf{y} = \begin{cases} 1 & argsort(\phi)[0:l] \\ 0 & for\ all\ other\ indexes \end{cases} \tag{5.2}$$

where $\mathbf{y} = [y_1, y_2, .., y_k, .., y_n]$, l top anchors are selected out of n based on cosine similarity score ϕ. They are assigned with label 1 in $\mathbf{y}$ and rest are assigned 0 as shown above with Eq. 5.2. Here each y_k represents the presence/absence of object of interest regions in the image. Intuition here is that $\mathbf{f}_k^{anc}$'s which contains the object will lie closer to $\mathbf{f}_{ao}^{text}$ in feature space.

Region proposal Network (RPN): The RPN branch shown in Fig. 5.3 is inspired from FasterRCNN [25]. The RPN generated proposals are used to pool a list of features from the feature map $\mathbf{F}$. The pooled features are $[\mathbf{f}_1^p, \mathbf{f}_2^p, ..., \mathbf{f}_n^p] \in \mathbb{R}^C$. Now the anchor features $[\mathbf{f}_1^{anc}, \mathbf{f}_2^{anc}, ..., \mathbf{f}_n^{anc}]$, proposal features $[\mathbf{f}_1^p, \mathbf{f}_2^p, ..., \mathbf{f}_n^p]$ and pseudo label $y = [y_1, y_2, .., y_n]$ are used to train the function $LFE(.)$ using contrastive learning as explained in next section.

Contrastive Pre-training: Recall that the current benchmark CZSL datasets [10, 14, 15] do not have ground truth bounding boxes for objects of interest. For this reason,

we use both the anchor features and proposal features to localize the objects of interest. This is different from regular object detection networks [25]. The pseudo label $\mathbf{y}$ informs the network which anchor features are likely to represent object(s) of interest. Using contrastive learning, we train the regional proposal network branch to localize the object with weak supervision. The goal of contrastive learning is to maximize the similarity between similar feature vectors and minimize the similarity between the dissimilar feature vectors. Here, the objective is to maximize the cosine similarity between $< \mathbf{f}_k^{anc}, \mathbf{f}_k^p >$ features where there is a possibility of object being present and minimize in all other cases. The contrastive objective function is:

$$\mathcal{L}_{CON} = \sum_{k=1}^{n} y_k * d_k^2 + (1 - y_k) * \max(0,\ 1 - d_k^2), \tag{5.3}$$

where $*$ is element wise multiplication, y_k tells us which features have the possibility of having an object (Eq. 5.2), d_k is the cosine distance between $< \mathbf{f}_k^{anc}, \mathbf{f}_k^p >$.

$$d_k = \frac{\mathbf{f}_k^p \cdot \mathbf{f}_k^{anc}}{||\mathbf{f}_k^p||\ \ ||\mathbf{f}_k^{anc}||} \ \forall\ k = [1, 2, ..n] \tag{5.4}$$

Along with contrastive loss, we optimize binary cross entropy over the objectness score predicted by region proposal network. The overall loss function is:

$$\mathcal{L}_{total} = \alpha * \mathcal{L}_{CON} + \beta * \mathcal{L}_{BCE}(o, \phi), \tag{5.5}$$

where, α and β are empirically-determined scaling parameters, o is the objectness score from RPN and ϕ is the cosine distance from Eq. 5.1. Once the $LFE(.)$ network is trained, the output of trained model are the proposal feature vectors $[\hat{\mathbf{f}_1^p}, \hat{\mathbf{f}_2^p}, ..., \hat{\mathbf{f}_n^p}]$ along with objectness score $\hat{o} = [\hat{o_1}, \hat{o_2}, .., \hat{o_n}]$. This learnt parameter objectness score ensures selection of features with object information, thereby minimizing the interference from

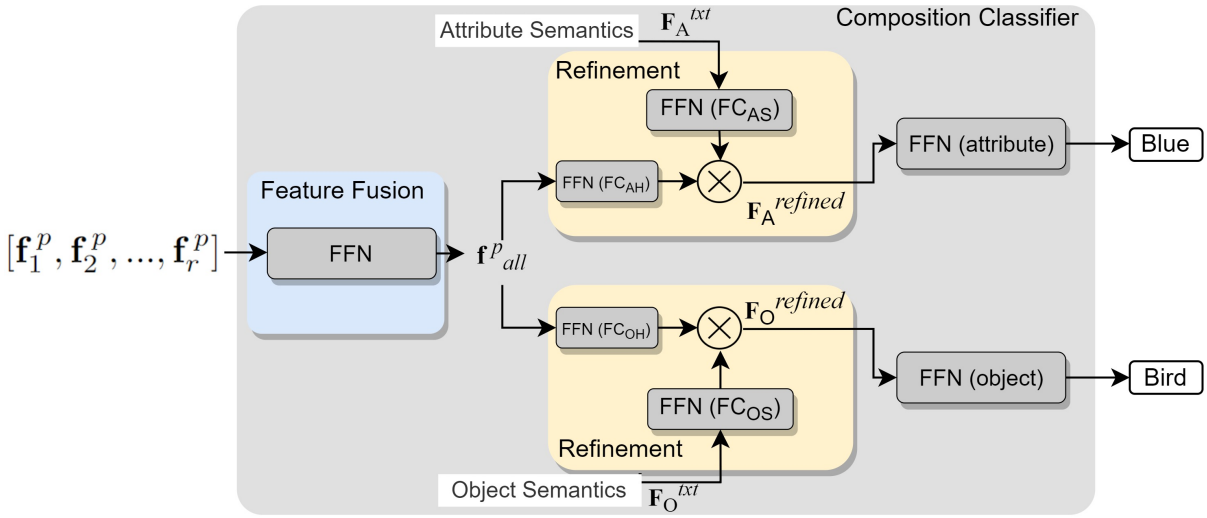

Figure 5.4: Composition Classifier $CC(.)$ architecture. The proposal features $[\hat{\mathbf{f}}_1^p, \hat{\mathbf{f}}_2^p, ..., \hat{\mathbf{f}}_r^p]$ are the outputs from $LFE(.)$ which are combined into a single representation $\mathbf{f}_{all}^p$. The attribute and object semantics are the semantic encoding of all attributes and objects under consideration. Two branches predict attribute and object from semantically refined $\mathbf{f}_{all}^p$.

potential confusing elements as shown in Fig. 5.1

5.3.2 Composition Classifier $CC(.)$

The ability to learn individual representation of O-A in visual domain is crucial for transferring knowledge from seen to unseen O-A associations. Existing SOTA works [9, 10, 125, 126, 141] use homogeneous features from whole image as without localizing the object, they ignore the discriminative visual features of object and its attributes. Our Composition Classifier network $CC(.)$ leverages the distinctive features extracted by $LFE(.)$ to predict the object and its corresponding attribute as shown in Fig. 5.4. It is challenging to associate right attribute with the object by using homogeneous features, as there can be interference from prominent confounding elements like examples shown in Fig5.1B.

The input to $CC(.)$ is a set of top r $(r < n)$ proposal feature vectors from pre-

trained $LFE(.)$ $[\mathbf{f}_1^p, \mathbf{f}_2^p, ..., \mathbf{f}_r^p] \in \mathbb{R}^{r \times C}$ sorted in descending order based on objectness score $o = [o_1, o_2, .., o_r]$. The proposal feature vectors are fused into a single visual feature $\mathbf{f}_{all}^p \in \mathbb{R}^{1 \times C}$ using weighted average with learnable parameters [142] as shown in Fig. 5.4. Input to the learnable parameters has $r \times C$ dimension. r is the number of proposals and C is the number of channels. We swap the dimensions at the input to fuse different proposals together. The output of it is a single feature vector of length C. The fusion operation is a learnable weighted average operation, that learns to create joint representation from features of object regions. Next, $\mathbf{f}_{all}^p$ is projected to two different representation via two feed forwards networks FC_{AH} and FC_{OH} layers as shown in Fig. 5.4. Similarly, the semantic embeddings of all attributes $\mathbf{F}_A^{text} = [\mathbf{f}_{a1}^{text}, \mathbf{f}_{a2}^{text}.., \mathbf{f}_{ai}^{text}]$ and all objects $\mathbf{F}_O^{text} = [\mathbf{f}_{o1}^{text}, \mathbf{f}_{o2}^{text}.., \mathbf{f}_{oj}^{text}]$ (generated from T_a and T_o) are projected via feed forward network FC_{AS} and FC_{OS}. Here T_a and T_o are the list of all attributes and objects in the dataset respectively, and $i = len(T_a)$, $j = len(T_o)$. $len(.)$ is length. Inspired by [8, 26, 53, 104, 105], the visual feature projections are refined as shown below. The particular choice of refinement by one-to-one multiplication is an empirical choice. We explore different refinement techniques in Table 5.6. This all is done by *Refinement* block as shown in Fig. 5.4.

$$\mathbf{F}_O^{refined} = FC_{OH}(\mathbf{f}_p^{all}) \circ FC_{OS}(\mathbf{F}_O^{txt}) \;\; ; \;\; \mathbf{F}_A^{refined} = FC_{AH}(\mathbf{f}_p^{all}) \circ FC_{AS}(\mathbf{F}_A^{txt}) \tag{5.6}$$

where, "$\circ$" represents element-wise multiplication and the feed forward network are two fully connected layers with ReLu activation. This refinement aggregates semantic information with the visual features. These refined features are passed through another feed forward network (Fig. 5.4) and softmax layers to make the final decision on the object $\hat{o}$ and attribute $\hat{a}$ present in the image I. To train the composition classifier function $CC(.)$, we optimize binary cross entropy function over the O-A prediction.

5.4 Experiments

5.4.1 Datasets

Table 5.1 summarizes the datasets used. In the MIT-states dataset the images are of natural objects collected using an older search engine with limited human annotation causing a significant label noise [29]. For UT-Zappos [14] the simplicity of the images (one object with white background) makes it unsuitable to work in natural surroundings. We performed experiments on very recently released Compositional GQA (CGQA) dataset [10,143]. This dataset is proposed to evaluate existing CZSL models in a more realistic challenging scenarios with background clutter. The splits used on all the datasets are as follows.

MIT-States [15] has a total of 53,000 images with 245 objects and 115 attributes. The splits for MIT-States dataset have 1262 object-attribute pairs (34,000 images) for the training set, 600 object-attribute pairs (10,000 images) as the validation set and 800 pairs (12,000 images) as test set.

UT-Zappos [14] has 29,000 images of shoes catalogue. The splits used are of 83 object-attribute pairs (23,000 images) for the training set, 30 object-attribute pairs (3,000 images) for the validation set and 36 pairs (3,000 images) for test set. These splits are selected following previous works [10,113]. The images in UT-Zappos [14] dataset are not really entirely a compositional dataset as the attributes like *Faux Leather vs Leather* are material differences but not specifically any visual difference [10]. Also, the simplicity of the images (one object with white background) makes it unsuitable to work in natural surroundings where the object of interest has interference confounding elements in the scene.

The third dataset used is Compositional-GQA (CGQA) dataset [9, 143]. It has 453 attributes and 870 objects. The splits for CGQA have 5592 object-attribute pairs (26,000

images) for training set, 2292 pairs (7,000 images) for validation set and 1811 pairs (5,000 images) for testing set. These splits are as proposed by [10]. The CGQA dataset have images curated from visual genome dataset [144] which comprises of images from natural and realistic settings. Most of the images in CGQA have an object of interest with confounding elements in the background, that makes it an extremely challenging dataset to evaluate CGQA models.

5.4.2 Implementation Details

Both the localized feature extractor $LFE(.)$ and Composition Classifier $CC(.)$ are trained on all the datasets. To train $LFE(.)$, an efficient contrastive pre-training framework is used with a margin distance of 1. As IE, we use ResNet-50 [45] pre-trained on [112]. For text encoding we utilize text encoder similar to [112]. The Anchor Generator generates 576 valid anchor boxes. Corresponding to each of these anchor boxes, features $[\mathbf{f}_1^{anc}, \mathbf{f}_2^{anc}, ..., \mathbf{f}_{576}^{anc}]$ are pooled from $\mathbf{F}$. To generate ϕ according to Eq. 5.7, each $\mathbf{f}_j^{anc}$ is matched with semantic word embedding vector, and top 20 scores are labeled as 1 and rest as 0 as shown in Eq. 5.8 to create the pseudo label y. This is an empirically selected value, it covers almost all the object regions in the image. The **Region Proposal Network** generates proposal boxes and an objectness score corresponding to each proposal box. The number of proposal boxes are equal to the number of anchor boxes. Corresponding to these proposals boxes, features $[\mathbf{f}_1^{p}, \mathbf{f}_2^{p}, ..., \mathbf{f}_{576}^{p}]$ are pooled from feature map $\mathbf{F}$.

Contrastive loss is used for pre-training. The cosine distance d_k is computed between each anchor feature $\mathbf{f}_k^{anc}$ and $\mathbf{f}_k^{p}$, the total number of features are 576 for anchors and 576 proposals. The pseudo label y is of length 576. y_k is equal to 1 where the $\mathbf{f}_k^{anc}$ feature have potential object. The scaling parameters for contrastive loss α and clas-

sification loss β are set to $0.6, 0.4$ respectively. The network is trained for 100 epochs, convergence is observed around 45 epochs, based on that, early stopping is done at 50 epochs. The learning rate starts with $1e^{-5}$ with decay of 0.1 after every 10 epochs. The batch size is set at 24. The optimizer used is *Adam* optimizer. The region proposal branch of the network learns to select features from regions where the objects are present. During training, we restrict the learning rate of linear projection layer of f_{ao}^{txt} to a low value to stabilize the region proposal branch.

Compositional Classifier $CC(.)$: The ability to learn individual representation of O-A in visual domain is crucial for transferring knowledge from seen to unseen O-A associations. Existing SOTA works [9, 10, 125, 126, 141] use homogeneous features from whole image as without localizing the object, they ignore the discriminative visual features of object and its attributes. Our Composition Classifier network $CC(.)$ leverages the distinctive features extracted by $LFE(.)$ to predict the object and its corresponding attribute. It is challenging to associate right attribute with the object by using homogeneous features, as there can be interference from prominent confounding elements. $CC(.)$ takes as input, the top 10 pooled features $[\mathbf{f}_1^p, \mathbf{f}_2^p, ..., \mathbf{f}_{10}^p]$ from pre-trained $LFE(.)$ sorted in descending order based on objectness score $\hat{o} = [\hat{o}_1, \hat{o}_2, .., \hat{o}_{10}]$. Each block in $CC(.)$ consists of two fully connected layer with ReLU activation. The initial learning rate for $CC(.)$ network is set to $1e^{-3}$ with a decay of 0.1 after every 7 epochs. We observed that fine tuning $LFE(.)$ with a lower learning rate of $1e^{-6}$ while training $CC(.)$ performed better than freezing it. The batch size used is 32. All the experiments are done on a single nvidia V100 Tesla.

	# Images			# Objects	#Attributes	# OA Pairs
	Train	Val	Test			
MIT-States [13]	30k	10k	13k	245	115	1962
UT-Zappos [14]	23k	3k	3k	12	16	116
CGQA [9]	26k	7k	5k	870	453	9378

Table 5.1: Comparison of different CZSL datasets [9, 13, 14]

5.4.3 Evaluation Metrics

Following current methods [8, 9, 113], we evaluate our network's performance in Generalized Compositional Zero Shot Learning Protocol (GCZSL). Under this protocol, we draw seen class accuracy vs unseen class accuracy curve at different operating points of the network. These operating points are selected from a single calibration scaler that is added to our network's predictions of the unseen classes [8, 9, 113]. We report area under "seen class accuracy vs unseen class accuracy curve" (AUC) as our performance metrics. Additionally, we report our network's performance on Top-1 accuracy in seen and unseen classes.

5.4.4 Results

LOCL outperforms current methods in the test set of all benchmark datasets in almost all categories as shown in Table 5.2. We evaluate LOCL's performance in terms of AUC under Generalized Compositional Zero Shot Learning (GCZSL) protocol. We also report Top-1 accuracy in seen and unseen classes. In MIT-States [15] LOCL outperforms SOTA method by 8% on unseen class accuracy and 1.7% AUC. In UT-Zappos [14] LOCL's unseen class accuracy is 5.2% better than the SOTA method. Moreover, it almost *doubles* the unseen class accuracy while achieving 1.1% improvement in terms of AUC for the more challenging CGQA [9] dataset.

Current CZSL methods use homogeneous features from the backbone instead of using

distinctive visual features of objects and attributes. While such techniques may work on simpler datasets like UT-Zappos [14], as evidenced by the high performance, the more realistic datasets such as CGQA pose challenges. Table 5.2 shows, LOCL achieves the best results on the challenging CGQA dataset. Bias towards seen O-A compositions is a common issue [113] in current CZSL methods. Recent approaches [9, 16] have utilized a graph structure with message passing/blocking [16] or prior possible O-A knowledge [9] to reduce this bias. However, they still tend to be biased towards seen O-A pairs at inference as pointed out by the authors of [16]. In contrast, LOCL learns distinct object and attribute representations in the two separate branches of the $CC(.)$ and achieves high unseen class accuracy and AUC.

BMPNet [16] achieves state-of-the-art (SOTA) performance in seen classes of MIT-States [13] and UT-Zappos [14]. However, their sub-optimal performance in unseen classes indicates a bias towards seen classes. To further investigate this bias, we evaluate BMPNet on the challenging CGQA dataset [9]. We utilize the official repository provided by the authors for this evaluation and report performance in the same matrices used for other datasets. In Table 5.3, we can observe LOCL outperforms BMPNet in all category. Especially in unseen classes, LOCL achieves more than double accuracy than BMPNet. This poor performance indicates a seen class bias of BMPNet. Moreover, LOCL is very efficient and utilizes only $\sim$ 5GB memory for training in the large-scale dataset CGQA. Current graph-based SOTA networks CGE [9] ($\sim$ 10GB), BMPNet [16] ($\sim$ 40GB) utilize much higher GPU memory for the same batch size in CGQA dataset. Therefore, LOCL is suitable for training on large scale challenging CZSL datasets.

Methods	MIT-States [15]			UT-Zappos [14]			CGQA [9]		
	Seen	Unseen	AUC	Seen	Unseen	AUC	Seen	Unseen	AUC
Attop [132]	14.3	17.4	1.6	59.8	54.2	25.9	11.8	3.9	0.3
LabelEmbed [13]	15	20.1	2.0	53.0	61.9	25.7	16.1	5	0.6
TMN [113]	20.2	20.1	2.9	58.7	60.0	29.3	21.6	6.3	1.1
SymNet [8]	24.2	25.2	3.0	49.8	57.4	23.4	25.2	9.2	1.8
CompCos [125]	25.3	24.6	4.5	59.8	62.5	28.1	28.1	11.2	2.6
ProtoProp [126]	-	-	-	62.1	65.5	34.7	26.4	18.1	3.7
BMP-Net [16]	**38.6**	21.7	6.0	**87.3**	64.5	**49.7**	-	-	-
CGE [9]	32.8	28.0	6.5	64.5	71.5	33.5	**31.4**	14	3.6
LOCL (Ours)	35.3	**36.0**	**7.7**	68.0	**76.7**	37.9	29.6	**26.4**	**4.2**

Table 5.2: Performance comparisons on MIT-States [15], UT-Zappos [14], CGQA [9] Datasets. '-' means unreported performance in a particular category. In all three datasets, LOCL significantly outperform current methods. Specially, for the more challenging (significant background clutter) CGQA dataset, the effectiveness of LFE is clearly demonstrated by its performance on the unseen O-A associations.

Methods	CGQA [9]		
	Seen	Unseen	AUC
BMP-Net* [16]	29.1	11.7	2.7
LOCL (Ours)	**29.6**	**26.4**	**4.2**

Table 5.3: Performance comparison on CGQA [9] dataset. LOCL significantly outperform BMP-Net [16] in a challenging (significant background clutter) dataset. The performance of LOCL shows the effectiveness of **LEF** in unseen OA associations. * refers to as our implementation as the authors do not evaluate their model in CGQA dataset.

5.4.5 Ablation Study

5.4.6 Image Encoder and Localized Feature Extractor

Our Localized Feature Extractor $LFE(.)$ is modular and can easily be adapted to other methods. In Table 5.4, we show different SOTA methods' improved performances with our feature extraction. For the sake of fair comparison with SOTA methods: SymNet [8], ComCos [10] and CGE [9], we replace their backbones with our image encoder (IE) pre-trained on a larger dataset [112]. As expected, both our IE and $LFE(.)$ improve the existing networks' performances. This shows that the performance improvement is because of localized features generated from $LFE(.)$. All existing methods use ResNet-18

Methods	Our IE	LFE	CGQA [9]			MIT-States [15]		
			Seen	Unseen	AUC	Seen	Unseen	AUC
SymNet [8]	✗	✗	25.2	9.2	1.8	24.2	25.2	3.0
	✓	✗	25.3	9.3	1.8	26.6	26.1	3.5
	✓	✓	27.7	13.5	2.0	28.7	27.7	3.8
CompCos [10]	✗	✗	28.1	11.2	2.6	25.3	24.6	4.5
	✓	✗	28.4	13.5	2.8	25.6	24.8	4.5
	✓	✓	28.9	16.7	2.9	27.9	26.7	5.1
CGE [9]	✗	✗	31.4	14.0	3.6	32.8	28	6.5
	✓	✗	31.4	19.3	3.8	33.3	28	6.5
	✓	✓	**31.9**	26.1	4.1	**36.3**	29.8	6.6
LOCL	✓	✓	29.6	**26.4**	**4.2**	35.3	**36.0**	**7.7**

Table 5.4: Performance of SOTA methods with our image encoder (IE) and LFE.

as the backbone following the seminal work of [13]. We recommend that CZSL networks should utilize stronger backbones for challenging datasets like CGQA [9]. However, our improved performance is not just coming from a stronger IE. With $LFE(.)$, the performance boost is more significant (specially in terms of unseen class accuracy) than the performance boost with our IE for the CGQA dataset. In particular, $LFE(.)$ increases the unseen class accuracy of CGE [9] in CGQA by *86%*, and other methods also get great improvement with $LFE(.)$. The performance improvement in the MIT-States dataset is less due to the noisy annotations [29] of this dataset. In summary, $LFE(.)$ improves three different architectures thus proving the effectiveness of localized feature extraction.

5.4.7 Number of Proposals

Table 5.5 shows the selection criterion of number of proposal selected from pre-trained $LFE(.)$ the goes as input to $CC(.)$. With $r < 10$, the proposals features miss regions of the object, which leads to poor performance. While when $r > 10$, more background features are picked that suppress the prominent object and lead to drop in prediction quality.

# of proposals	Seen	Unseen	AUC
5	32.1	33.6	7.2
10	**35.3**	**36.0**	**7.7**
15	35.3	35.9	6.9
20	27.6	28.4	6.5

Table 5.5: Performance of LOCL as we select different number of top r proposals from pre-trained LFE in the MIT-States [15]. Best performance is observed with *r=10*. With $r > 10$, more background features are picked that suppress the prominent objects.

5.4.8 Object\Attribute Refinement

Table 5.6 shows refinement operations done on visual features $\mathbf{f}_p^{all}$ as shown in Equation 5.6. The multiplication operations generates more selective information and suppresses the redundant information as compared to concatenation and addition operation [53, 104].

Method	Seen	Unseen	AUC
Addition	28.5	29.6	6.6
Multiplication	**35.3**	**36.0**	**7.7**
Concatenation	32.7	33.1	7.2

Table 5.6: Performance of compositional classifier with different refinement operations in the MIT-States dataset [15].

5.4.9 Pre-training $LEF(.)$ with object embeddings

As discussed in section 5.3.1, we use OA pair name $< Blue, Bird >$ as input during the pre-training of $LEF(.)$. We also test with using only the object names $< Bird >$ as the input. We observe 3% drop in accuracy as compared to OA pair names in the unseen category as shown in Table 5.7. This is expected as the text embeddings generated by the text encoder are more meaningful and have closer representation with the visual features when we provide a complete description of the object in the image i.e. OA pair name.

Names Used	Seen	Unseen	Obj	Attr	*AUC*
Obj-Attr	**35.3**	**36.0**	**42.7**	**53.4**	**7.7**
Obj	32.5	32.8	37.4	41.9	7.1

Table 5.7: Performance of the network in MIT-States [15] with different names used as input to the text encoder while pre-training $LFE(.)$. **Bold** numbers are the best performance setting. The network performs well with $Obj - Attr$ names as input compared to just *Obj* names. The Obj and Attr columns display the top-1 accuracy for detecting objects and attributes, respectively.

5.4.10 Number of Pseudo Labels

For creating pseudo labels y during pre-training, we assign value 1 to top 20 indexes and rest are assigned 0. The equation is:

$$\phi = [\phi_1, \phi_2, .., \phi_k, .., \phi_n] \tag{5.7}$$

$$y = \begin{cases} 1 & argsort(\phi)[0:l] \\ 0 & for\ all\ other\ indexes \end{cases} \tag{5.8}$$

where $y = [y_1, y_2, .., y_k, .., y_n]$, 20 anchors are selected based on cosine similarity score ϕ. They are assigned with label 1 in y and rest are assigned 0 as shown above with Eq. 5.8. Here each y_k represents the presence/absence of object of interest regions in the input image. We experiment with different values for number of potential objects. As shown in Table 5.8, the overall performance of the model drops if we pick a number greater than or less than 20. This is because for smaller value, the $LFE(.)$ is penalized for detecting even the right regions of interests and for larger value than 20, we are learning information from confounding elements from the background where the object may/may not be present.

# Pseudo Labels	Seen	Unseen	AUC	Obj	Attr
10	31.5	27.9	5.2	27.9	31.5
15	33.8	34.1	6.5	28.4	30.8
20	**35.3**	**36.0**	**7.7**	**42.7**	**53.4**
25	29.1	29.6	6.1	31.5	34.6

Table 5.8: Performance of the network in MIT-States [15] with different number of region of interest while pre-training $LFE(.)$. **Bold** numbers are the best performance settings. Here # is "Number of". Obj, Attr columns present the top-1 accuracy in detecting objects and attributes respectively.

5.4.11 Margin distance for contrastive loss

For pre-training $LEF(.)$ with contrastive loss, we use a margin distance of 1. We experimented with different distances for the margin for MIT-states [15] dataset. We achieved best performance at a margin of 1. The experimental evaluation with different margin distance is shown in Table 5.9. Our observations of is that with bigger margin, the network start clustering features from those regions also, which have object of interest along with a significant section of background regions. This leads to drop in attribute detection accuracy.

Margin	Seen	Unseen	AUC	Obj	Attr
0.5	29.6	30.4	5.2	30.2	47.3
1.0	**35.3**	**36.0**	**7.7**	**42.7**	**53.4**
3	34.1	33.9	6.5	41.1	46.8
7	25.3	26.5	4.8	37.3	38.9

Table 5.9: Performance of pre-training the $LFE(.)$ using different margin distance for contrastive learning in MIT-States [15]. We achieve best performance when margin is 1. For higher margin, $LFE(.)$ cluster features of object of interest which have significant region of background/confounding regions also. Leaning to poor performance. Bold numbers are the best performance setting. The Obj and Attr columns display the top-1 accuracy for detecting objects and attributes, respectively.

5.4.12 Scaling parameters of loss function

While pre-training, we combine contrastive loss and binary cross entropy loss using scaling parameters α and β. The equation is:

$$\mathcal{L}_{total} = \alpha * \mathcal{L}_{CON} + \beta * \mathcal{L}_{BCE}(o, \phi), \tag{5.9}$$

where, $\mathcal{L}_{CON}$ is the contrastive loss and $\mathcal{L}_{BCE}$ is the binary cross entropy loss. We test for different values α and β as shown in Table 5.10. It appears giving a bit more weight to the contrastive loss helps $LFE(.)$ to extract better localized features.

α	β	*Seen*	*Unseen*	*AUC*	Obj	Attr
0.3	*0.7*	30.6	32.5	6.9	30.9	33.1
0.4	*0.6*	35.1	35.4	7.5	36.4	39.9
0.5	*0.5*	34.9	35.8	7.7	40.8	49.3
0.6	*0.4*	**35.3**	**36.0**	**7.7**	**42.7**	**53.4**
0.7	*0.3*	32.7	33.7	7.1	33.0	35.2

Table 5.10: Performance of the network with different scaling parameters of the loss function during pre-training in MIT-States [15]. **Bold** numbers are the best performance settings. Obj, Attr columns present the top-1 accuracy in detecting objects and attributes respectively.

5.5 Qualitative Results

We show qualitative results for unseen novel composition with top-1 prediction in Figure 5.5. The examples are presented from datasets : CGQA [9], MIT-States [15], and UT-Zapos [14]. The order of the datasets is in decreasing order of the clutter in the images. As can be seen that in the CGQA dataset, the images contains object of interest with lot of confounding elements creating background clutter. MIT-States [15] is also of natural images. However, most of the images have a dominant object. On the other hand, in UT-Zappos [14] all the images contain a single object with clear white

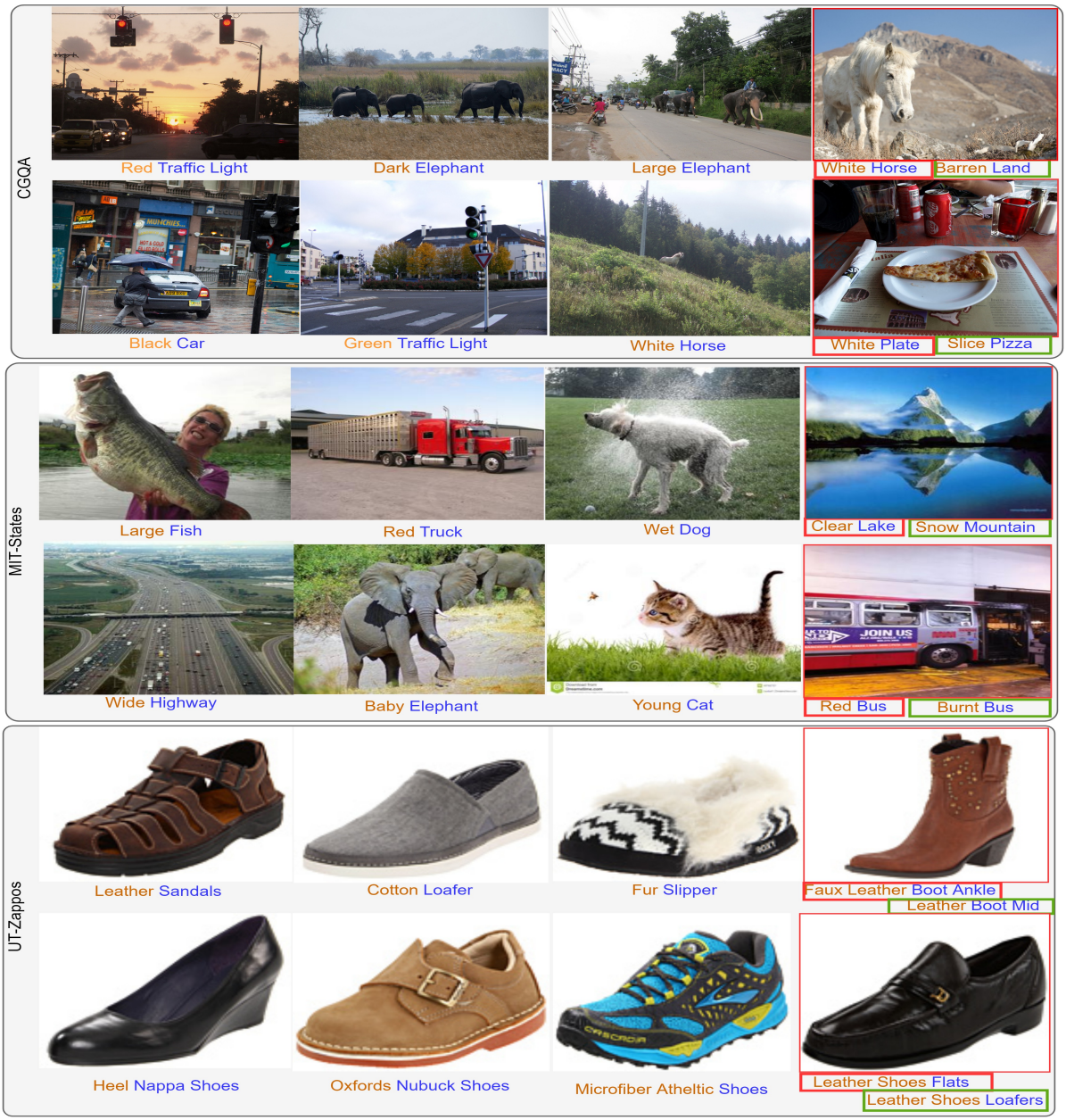

Figure 5.5: Qualitative results of LOCL. Left three columns show correct predictions from our network. Rightmost column shows missed predictions, here, ground truth labels are marked with green box and our predictions are marked in red box. The datasets contain only one OA pair and our predictions though visually correct, do not match with the ground-truth OA in these cases.

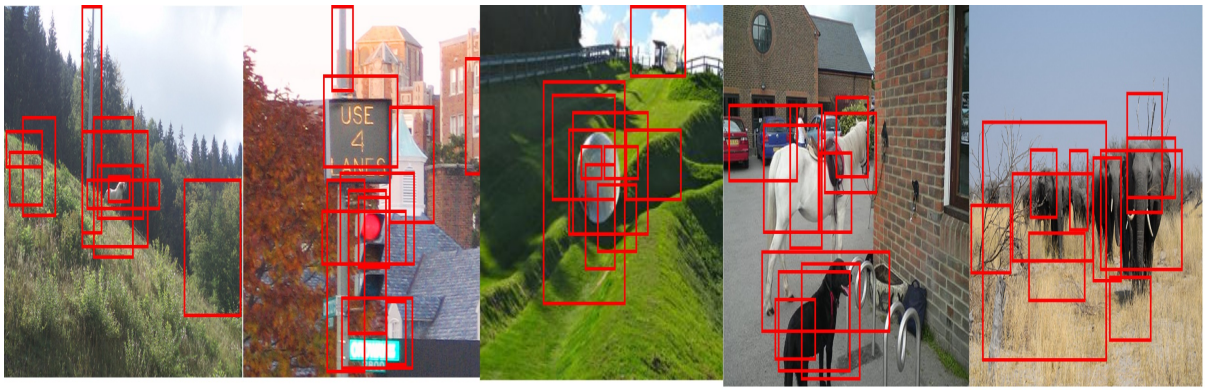

Figure 5.6: Proposals selected based on objectness score. We can see that the proposals are generated on the object of interest. Though LOCL is not designed for multi O-A, but in case of multiple objects, the proposals are distributed over multiple objects.

background. This shows the complexity and the challenges of CGQA dataset compared to the existing ones.

The first three columns represent the examples where our model is making the right predictions. The last column in each dataset shows examples where our model makes the visually correct prediction. However, it does not match with the ground truth label of object and attribute. Our model is selecting object of interest, and it is creating the right attribute-object associations. For example in case of fourth row on the rightmost column, our prediction of the object is right but the image contains multiple attributes, while the ground truth contains only one OA pair. This put an artificial limitation on the evaluation metric even when the predictions are perceiveably correct.

We also show $LFE(.)$ proposals quality in eliminating background in Fig. 5.6. As can be seen $LFE(.)$ only generates proposals for the objects of interest. Even it can generate proposals covering multiple O-As.

5.6 Summary

We present a novel two-step approach, LOCL, for recognizing O-A pairs. Our approach includes a robust local feature extractor followed by a composition classifier. LOCL is evaluated on benchmark datasets. Additionally, our experiments show that the local feature extractor improves the performance of current SOTA CZSL models by a significant margin of 12%.

Chapter 6

Conclusion and Discussion

Action-detection methods are crucial for various security and consumer applications. The primary objective of this thesis is to develop automated methods for detecting and localizing actions. In this chapter, we recap our contributions and explore potential future research directions inspired by this work. Our developed methods focus on two key aspects: atomic action detection and compositional capability, both essential for robust action detection.

In the first half of our thesis, we presented two novel methods for atomic action detection: VSGNet (see chapter 2) and GTNet (see chapter 3). Both of these methods can successfully detect and localize atomic action. In these methods, our main contributions are spatial refinement technique and an attention architecture to identify salient spatial context. Our thorough analysis has clearly shown the positive impact of our innovations on atomic action detection. Spatial refinement helps the the two stage sequential methods to filter unnecessary human-object pairs. On the other hand, our innovative guided attention architecture clearly identifies the spatial regions that are important for understanding atomic actions. These collective efforts have markedly advanced the frontiers of atomic action detection methods. Impressively, both these methods operate in real-time,

processing images at a rate of 26 FPS in a small NVIDIA RTX 2080 Ti GPU.

In the latter part of the thesis, we introduced two methods endowed with compositional capabilities. Compositional ability enables methods to leverage previously learned concepts to recognize new ones. In this regard, we devised DDS (as detailed in chapter 4) and LOCL (explained in chapter 5).

DDS is a multi-branch decoupled network for relationship triplet prediction. We utilize a Transformer like encode-decoder architecture for each of our branch. This design ensures decoupled concept learning. As a result, DDS has been able to compose previously learned concepts to detect out of vocabulary relationship triplets. DDS has achieved SOTA performances in three benchmark datasets. Remarkably, DDS can identify exceptionally rare relationship triplets by drawing upon foundational concepts it has learned earlier.

We also developed LOCL to correctly associate objects with attributes. Attributes play an important role in detecting actions [27]. We created a contrastive learning based pretraining strategy to identify potential object locations with specific attributes. The pretrained feature extractor subsequently generates localized features. These are then utilized by a composition classifier to detect accurate O-A associations. Our assessments indicate a marked enhancement over prior methods, attributable to our innovative pretraining approach. Furthermore, substituting conventional feature extractors with our superior version also resulted in performance boosts in existing models.

6.1 Future Work

Integration of Models: The methods we have developed can act as foundational elements for more intricate methods. Our atomic action detection models, namely VS-GNet and GTNet, are modular in design. However, they face challenges in detecting

Figure 6.1: Consider a complex activity depicted across three consecutive frames illustrating an exchange of objects. In the first frame, person 1 is observed walking out of a door with a red bottle. In the subsequent frame, person 2 is walking out of the door with a coffee mug. At the third frame, the items have switched hands: person 1 now holds the coffee mug, while person 2 possesses the red bottle. Despite the absence of direct visual evidence of the objects being exchanged, one can deduce from the sequence of these three frames that an object exchange has happened.

actions that are scarcely represented in the training set. Our proposed DDS addresses this limitation by leveraging compositional capabilities. A logical next step for future research would be the cohesive integration of these models. Given their modular structure, such integration promises to be relatively straightforward. Additionally, it's feasible to amalgamate these action detection models with our compositional object-attribute (O-A) association model, LOCL. This amalgamation would endow the unified action detection model with enhanced capabilities. Another intriguing avenue for exploration is quantifying the influence of attributes on action detection. This integrated model can be utilized for various applications such as content moderation, autonomous driving. A next research avenue can focus on reducing the size of developed models without hurting the performances. This will ensure the implementation of our models on edge devices.

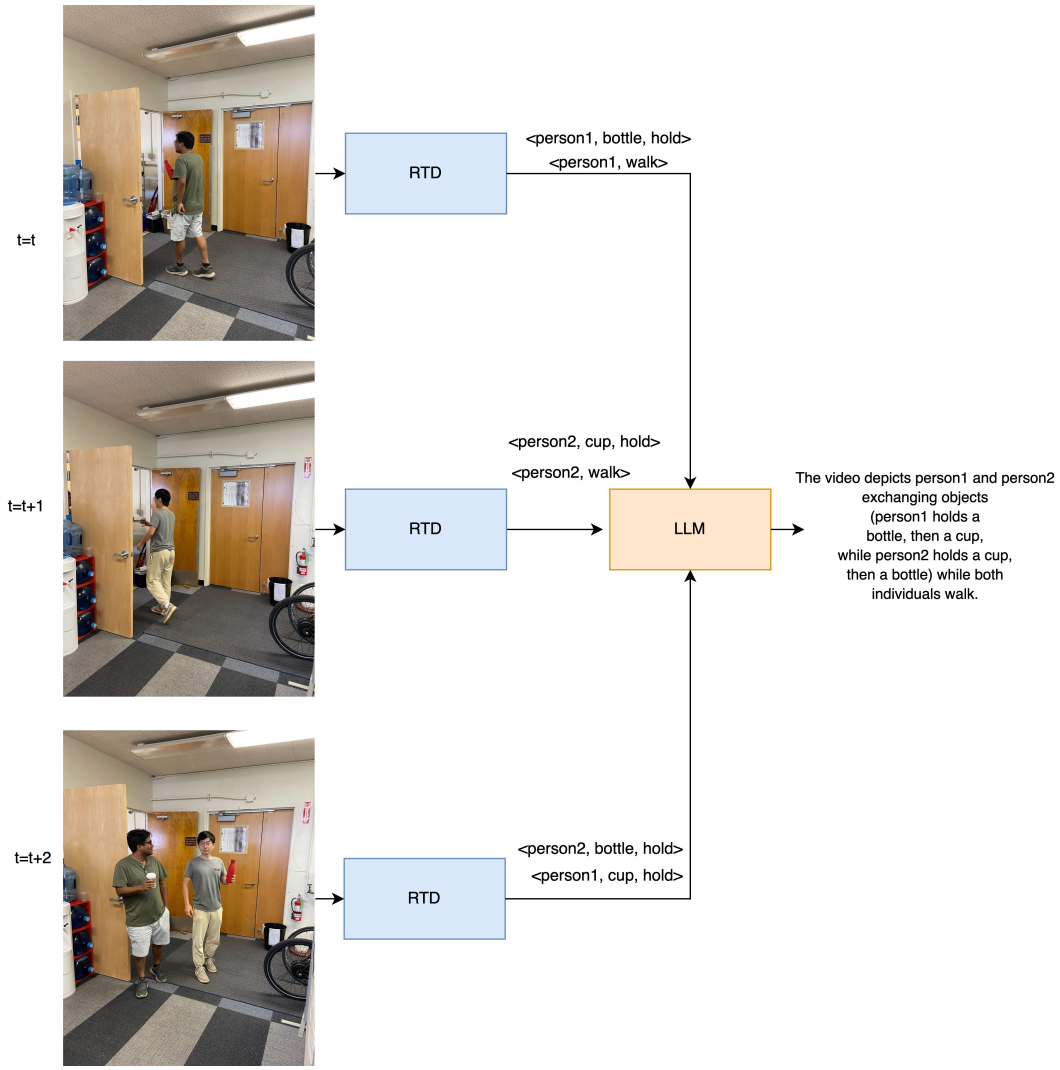

Figure 6.2: Complex activity detection method. RTD refers to relationship triplet detectors. For each frame of an input video we predict relationship triplets by RTDs. These triplets are fed to the LLM to infer complex activity. Our DDS is an example of a RTD. In this experiment, we utilize an oracle RTD. As LLM we utilize ChatGPT [11]. The LLM can perfectly infer the complex activity without using any visual input.

Complex Activity Detection with LLMs: Complex activity refers to the activities that are composed of multiple atomic actions [145]. An example can be seen in Figure 6.1. Although there is no visual evidence of the action: exchange of objects, we can infer it from the images shown here. Detecting this kind of activity automatically is extremely difficult as the method needs to have the power of detection and hierarchical reasoning. With the recent remarkable progress in multi-modal Large Language Models (LLMs) [146–149], we can build method to detect complex activity by integrating our models with the LLM. A method for complex activity detection is shown in Figure 6.2. We can utilize LLM for inferring such complex activity. Here, RTD refers to relationship triplet detectors. Our DDS is such a kind of network. The output of these RTDs are relationship triplets that are fed to the LLM to infer complex activity. It is remarkable to see the performance of LLM without any kind of visual input. This framework is limited by the quality of RTDs. Future research can start from this kind of method and explore the utilization our developed models in this thesis with LLMs to build a robust complex activity detection method.

Bibliography

[1] S. Gupta and J. Malik, *Visual semantic role labeling*, *arXiv preprint arXiv:1505.04474* (2015).

[2] J. Peyre, J. Sivic, I. Laptev, and C. Schmid, *Weakly-supervised learning of visual relations*, in *Proceedings of the ieee international conference on computer vision*, pp. 5179–5188, 2017.

[3] O. Ulutan, S. Rallapalli, M. Srivatsa, C. Torres, and B. Manjunath, *Actor conditioned attention maps for video action detection*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 527–536, 2020.

[4] C. Gao, J. Xu, Y. Zou, and J.-B. Huang, *Drg: Dual relation graph for human-object interaction detection*, in *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[6] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, *Learning to detect human-object interactions*, in *2018 ieee winter conference on applications of computer vision (wacv)*, pp. 381–389, IEEE, 2018.

[7] S. Wang, Y. Duan, H. Ding, Y.-P. Tan, K.-H. Yap, and J. Yuan, *Learning transferable human-object interaction detector with natural language supervision*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 939–948, 2022.

[8] Y.-L. Li, Y. Xu, X. Mao, and C. Lu, *Symmetry and group in attribute-object compositions*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11316–11325, 2020.

[9] M. F. Naeem, Y. Xian, F. Tombari, and Z. Akata, *Learning graph embeddings for compositional zero-shot learning*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 953–962, 2021.

[10] M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata, *Learning graph embeddings for open world compositional zero-shot learning*, *arXiv preprint arXiv:2105.01017* (2021).

[11] OpenAI, *Chatgpt: Interactive and conversational ai*, 2022.

[12] Y. Cong, W. Liao, H. Ackermann, B. Rosenhahn, and M. Y. Yang, *Spatial-temporal transformer for dynamic scene graph generation*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16372–16382, 2021.

[13] I. Misra, A. Gupta, and M. Hebert, *From red wine to red tomato: Composition with context*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1792–1801, 2017.

[14] A. Yu and K. Grauman, *Semantic jitter: Dense supervision for visual comparisons via synthetic images*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5570–5579, 2017.

[15] P. Isola, J. J. Lim, and E. H. Adelson, *Discovering states and transformations in image collections*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1383–1391, 2015.

[16] Z. Xu, G. Wang, Y. Wong, and M. S. Kankanhalli, *Relation-aware compositional zero-shot learning for attribute-object pair recognition*, *IEEE Transactions on Multimedia* (2021).

[17] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, *et. al.*, *Ava: A video dataset of spatio-temporally localized atomic visual actions*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6047–6056, 2018.

[18] 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*, pp. 5410–5419, 2017.

[19] A. I. News, *Surveillance city: Nypd can use more than 15,000 cameras to track people using facial recognition in manhattan, bronx and brooklyn*, 2021.

[20] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton, *et. al.*, *Attend, infer, repeat: Fast scene understanding with generative models*, in *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

[21] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, *Semantic understanding of scenes through the ade20k dataset*, *International Journal of Computer Vision* **127** (2019), no. 3 302–321.

[22] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, *Unified perceptual parsing for scene understanding*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 418–434, 2018.

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

[24] 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*, pp. 4709–4717, 2017.

[25] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, *Advances in neural information processing systems* **28** (2015).

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, *Advances in neural information processing systems* **30** (2017).

[27] N. Saini, B. He, G. Shrivastava, S. S. Rambhatla, and A. Shrivastava, *Recognizing actions using object states*, in *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[28] K. Wei, M. Yang, H. Wang, C. Deng, and X. Liu, *Adversarial fine-grained composition learning for unseen attribute-object recognition*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October, 2019.

[29] Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik, *A causal view of compositional zero-shot recognition*, in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 1462–1473, Curran Associates, Inc., 2020.

[30] 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*, pp. 299–307, 2017.

[31] 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*, pp. 4223–4232, 2018.

[32] X. Liu, P. Ghosh, O. Ulutan, B. Manjunath, K. Chan, and R. Govindan, *Caesar: cross-camera complex activity recognition*, in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pp. 232–244, 2019.

[33] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, *Learning actor relation graphs for group activity recognition*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9964–9974, 2019.

[34] Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu, *Transferable interactiveness knowledge for human-object interaction detection*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3585–3594, 2019.

[35] C. Gao, Y. Zou, and J.-B. Huang, *ican: Instance-centric attention network for human-object interaction detection*, in *British Machine Vision Conference*, 2018.

[36] G. Gkioxari, R. Girshick, P. Dollár, and K. He, *Detecting and recognizing human-object interactions*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8359–8367, 2018.

[37] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[38] 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*, pp. 779–788, 2016.

[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[40] 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*, pp. 21–37, Springer, 2016.

[41] K. Soomro, A. R. Zamir, and M. Shah, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, *arXiv preprint arXiv:1212.0402* (2012).

[42] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, *The thumos challenge on action recognition for videos "in the wild"*, *Computer Vision and Image Understanding* **155** (2017) 1–23.

[43] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, *Learning human-object interactions by graph parsing neural networks*, in *Proceedings of the European conference on computer vision (ECCV)*, pp. 401–417, 2018.

[44] A. Kolesnikov, A. Kuznetsova, C. Lampert, and V. Ferrari, *Detecting visual relationships using box attention*, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[45] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[46] L. Li, Z. Gan, Y. Cheng, and J. Liu, *Relation-aware graph attention network for visual question answering*, *arXiv preprint arXiv:1903.12314* (2019).

[47] 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 *European conference on computer vision*, pp. 740–755, Springer, 2014.

[48] R. Girshick, *Fast r-cnn*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[49] T. Gupta, A. Schwing, and D. Hoiem, *No-frills human-object interaction detection: Factorization, layout encodings, and training techniques*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9677–9685, 2019.

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

[51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[52] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size*, *arXiv preprint arXiv:1602.07360* (2016).

[53] O. Ulutan, A. Iftekhar, and B. S. Manjunath, *Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13617–13626, 2020.

[54] B. Wan, D. Zhou, Y. Liu, R. Li, and X. He, *Pose-aware multi-level feature network for human object interaction detection*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9469–9478, 2019.

[55] Y. Liu, Q. Chen, and A. Zisserman, *Amplifying key cues for human-object-interaction detection*, in *European Conference on Computer Vision*, pp. 248–265, Springer, 2020.

[56] T. Wang, R. M. Anwer, M. H. Khan, F. S. Khan, Y. Pang, L. Shao, and J. Laaksonen, *Deep contextual attention for human-object interaction detection*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5694–5702, 2019.

[57] H. Wang, W.-s. Zheng, and L. Yingbiao, *Contextual heterogeneous graph network for human-object interaction detection*, in *European Conference on Computer Vision*, pp. 248–264, Springer, 2020.

[58] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, *Show, attend and tell: Neural image caption generation with visual attention*, in *International conference on machine learning*, pp. 2048–2057, 2015.

[59] Y. Liu, J. Yuan, and C. W. Chen, *Consnet: Learning consistency graph for zero-shot human-object interaction detection*, in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 4235–4243, 2020.

[60] H.-S. Fang, Y. Xie, D. Shao, and C. Lu, *Dirv: Dense interaction region voting for end-to-end human-object interaction detection*, in *The AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[61] X. Zhong, C. Ding, X. Qu, and D. Tao, *Polysemy deciphering network for human-object interaction detection*, in *Proc. Eur. Conf. Comput. Vis*, 2020.

[62] D.-J. Kim, X. Sun, J. Choi, S. Lin, and I. S. Kweon, *Detecting human-object interactions with action co-occurrence priors*, in *European Conference on Computer Vision*, pp. 718–736, Springer, 2020.

[63] Z. Hou, X. Peng, Y. Qiao, and D. Tao, *Visual compositional learning for human-object interaction detection*, *arXiv preprint arXiv:2007.12407* (2020).

[64] Y.-L. Li, X. Liu, X. Wu, Y. Li, and C. Lu, *Hoi analysis: Integrating and decomposing human-object interaction*, in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 5011–5022, Curran Associates, Inc., 2020.

[65] J. Hu, L. Shen, and G. Sun, *Squeeze-and-excitation networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[66] Z. Hou, B. Yu, Y. Qiao, X. Peng, and D. Tao, *Affordance transfer learning for human-object interaction detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 495–504, 2021.

[67] Z. Hou, B. Yu, Y. Qiao, X. Peng, and D. Tao, *Detecting human-object interaction via fabricated compositional learning*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14646–14655, 2021.

[68] M. Tamura, H. Ohashi, and T. Yoshinaga, *Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10410–10419, 2021.

[69] M. Chen, Y. Liao, S. Liu, Z. Chen, F. Wang, and C. Qian, *Reformulating hoi detection as adaptive set prediction*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9004–9013, 2021.

[70] B. Kim, J. Lee, J. Kang, E.-S. Kim, and H. J. Kim, *Hotr: End-to-end human-object interaction detection with transformers*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 74–83, 2021.

[71] C. Zou, B. Wang, Y. Hu, J. Liu, Q. Wu, Y. Zhao, B. Li, C. Zhang, C. Zhang, Y. Wei, *et. al.*, *End-to-end human object interaction detection with hoi transformer*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11825–11834, 2021.

[72] Y.-L. Li, X. Liu, H. Lu, S. Wang, J. Liu, J. Li, and C. Lu, *Detailed 2d-3d joint representation for human-object interaction*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10166–10175, 2020.

[73] J. Peyre, I. Laptev, C. Schmid, and J. Sivic, *Detecting unseen visual relations using analogies*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1981–1990, 2019.

[74] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[75] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et. al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, *arXiv preprint arXiv:2010.11929* (2020).

[76] R. Girdhar and D. Ramanan, *Attentional pooling for action recognition*, *arXiv preprint arXiv:1711.01467* (2017).

[77] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, *Video action transformer network*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2019.

[78] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, *arXiv preprint arXiv:1905.11946* (2019).

[79] J. Pennington, R. Socher, and C. D. Manning, *Glove: Global vectors for word representation*, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[80] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, *How transferable are features in deep neural networks?*, in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[81] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, *Symmetric cross entropy for robust learning with noisy labels*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 322–330, 2019.

[82] H.-B. Zhang, Y.-Z. Zhou, J.-X. Du, J.-L. Huang, Q. Lei, and L. Yang, *Improved human-object interaction detection through skeleton-object relations*, *Journal of Experimental & Theoretical Artificial Intelligence* (2020) 1–12.

[83] X. Sun, X. Hu, T. Ren, and G. Wu, *Human object interaction detection via multi-level conditioned network*, in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 26–34, 2020.

[84] T. Wang, T. Yang, M. Danelljan, F. S. Khan, X. Zhang, and J. Sun, *Learning human-object interaction detection using interaction points*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4116–4125, 2020.

[85] Y. Liao, S. Liu, F. Wang, Y. Chen, C. Qian, and J. Feng, *Ppdm: Parallel point detection and matching for real-time human-object interaction detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 482–490, 2020.

[86] A. Bansal, S. S. Rambhatla, A. Shrivastava, and R. Chellappa, *Detecting human-object interactions via functional generalization*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 10460–10469, 2020.

[87] M. Popel and O. Bojar, *Training tips for the transformer model*, *The Prague Bulletin of Mathematical Linguistics* **110** (2018), no. 1 43–70.

[88] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *Mobilenetv2: Inverted residuals and linear bottlenecks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[89] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, *Densenet: Implementing efficient convnet descriptor pyramids*, *arXiv preprint arXiv:1404.1869* (2014).

[90] J. Ji, R. Desai, and J. C. Niebles, *Detecting human-object relationships in videos*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8106–8116, 2021.

[91] Y. Li, X. Yang, and C. Xu, *Dynamic scene graph generation via anticipatory pre-training*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13874–13883, 2022.

[92] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, *Action genome: Actions as compositions of spatio-temporal scene graphs*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10236–10247, 2020.

[93] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, *Image retrieval using scene graphs*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

[94] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. W. Taylor, and M. Volkovs, *Context-aware scene graph generation with seq2seq transformers*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15931–15941, 2021.

[95] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, *Learning to compose dynamic tree structures for visual contexts*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6619–6628, 2019.

[96] Y. Cong, H. Ackermann, W. Liao, M. Y. Yang, and B. Rosenhahn, *Nodis: Neural ordinary differential scene understanding*, in *European Conference on Computer Vision*, pp. 636–653, Springer, 2020.

[97] W. Wang, R. Wang, S. Shan, and X. Chen, *Exploring context and visual pattern of relationship for scene graph generation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2019.

[98] J. Shi, Y. Zhong, N. Xu, Y. Li, and C. Xu, *A simple baseline for weakly-supervised scene graph generation*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16393–16402, 2021.

[99] W. Wang, R. Wang, and X. Chen, *Topic scene graph generation by attention distillation from caption*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15900–15910, 2021.

[100] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, *Counterfactual critic multi-agent training for scene graph generation*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4613–4623, 2019.

[101] M.-J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, *Recovering the unbiased scene graphs from the biased ones*, in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1581–1590, 2021.

[102] Z. Hou, X. Peng, Y. Qiao, and D. Tao, *Visual compositional learning for human-object interaction detection*, in *ECCV*, 2020.

[103] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, *Neural motifs: Scene graph parsing with global context*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018.

[104] A. Iftekhar, S. Kumar, R. A. McEver, S. You, and B. Manjunath, *Gtnet: Guided transformer network for detecting human-object interactions*, *arXiv preprint arXiv:2108.00596* (2021).

[105] A. Iftekhar, H. Chen, K. Kundu, X. Li, J. Tighe, and D. Modolo, *What to look at and where: Semantic and spatial refined transformer for detecting human-object interactions*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5353–5363, 2022.

[106] B. Kim, J. Mun, K.-W. On, M. Shin, J. Lee, and E.-S. Kim, *Mstr: Multi-scale transformer for end-to-end human-object interaction detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19578–19587, 2022.

[107] A. Zhang, Y. Liao, S. Liu, M. Lu, Y. Wang, C. Gao, and X. Li, *Mining the benefits of two-stage and one-stage hoi detection*, *Advances in Neural Information Processing Systems* **34** (2021) 17209–17220.

[108] Y. Liao, A. Zhang, M. Lu, Y. Wang, X. Li, and S. Liu, *Gen-vlkt: Simplify association and enhance interaction understanding for hoi detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20123–20132, 2022.

[109] J. Park, S. Lee, H. Heo, H. K. Choi, and H. J. Kim, *Consistency learning via decoding path augmentation for transformers in human object interaction detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1019–1028, 2022.

[110] D. Zhou, Z. Liu, J. Wang, L. Wang, T. Hu, E. Ding, and J. Wang, *Human-object interaction detection via disentangled transformer*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19568–19577, 2022.

[111] X. Qu, C. Ding, X. Li, X. Zhong, and D. Tao, *Distillation using oracle queries for transformer-based human-object interaction detection*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19558–19567, 2022.

[112] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et. al.*, *Learning transferable visual models from natural language supervision*, in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[113] S. Purushwalkam, M. Nickel, A. Gupta, and M. Ranzato, *Task-driven modular networks for zero-shot compositional learning*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3593–3602, 2019.

[114] S. Kumar, A. Iftekhar, E. Prashnani, and B. Manjunath, *Locl: Learning object-attribute composition using localization*, *arXiv preprint arXiv:2210.03780* (2022).

[115] K. Kato, Y. Li, and A. Gupta, *Compositional learning for human object interaction*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 234–251, 2018.

[116] Z. Hou, B. Yu, Y. Qiao, X. Peng, and D. Tao, *Affordance transfer learning for human-object interaction detection*, in *CVPR*, 2021.

[117] Z. Hou, B. Yu, and D. Tao, *Discovering human-object interaction concepts via self-compositional learning*, in *ECCV*, 2022.

[118] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, *Hollywood in homes: Crowdsourcing data collection for activity understanding*, in *European Conference on Computer Vision*, pp. 510–526, Springer, 2016.

[119] L. Ilya, H. Frank, *et. al.*, *Decoupled weight decay regularization*, *Proceedings of ICLR* (2019).

[120] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et. al.*, *Pytorch: An imperative style, high-performance deep learning library*, *Advances in neural information processing systems* **32** (2019).

[121] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, *Visual relationship detection with language priors*, in *European conference on computer vision*, pp. 852–869, Springer, 2016.

[122] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, *Scene graph generation from objects, phrases and region captions*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1261–1270, 2017.

[123] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, *Graph r-cnn for scene graph generation*, in *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

[124] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, *Unbiased scene graph generation from biased training*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020.

[125] M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata, *Open world compositional zero-shot learning*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5222–5230, 2021.

[126] F. Ruis, G. Burghours, and D. Bucur, *Independent prototype propagation for zero-shot compositionality*, *arXiv preprint arXiv:2106.00305* (2021).

[127] D. Huynh and E. Elhamifar, *Compositional zero-shot learning via fine-grained dense feature composition*, *Advances in Neural Information Processing Systems* **33** (2020) 19849–19860.

[128] M. Jaderberg, K. Simonyan, A. Zisserman, *et. al.*, *Spatial transformer networks*, *Advances in neural information processing systems* **28** (2015).

[129] X. Zhao, Y. Yang, F. Zhou, X. Tan, Y. Yuan, Y. Bao, and Y. Wu, *Recognizing part attributes with insufficient data*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 350–360, 2019.

[130] D. Huynh and E. Elhamifar, *Fine-grained generalized zero-shot learning via dense attribute-based attention*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4483–4493, 2020.

[131] Z. Nan, Y. Liu, N. Zheng, and S.-C. Zhu, *Recognizing unseen attribute-object pair with generative model*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8811–8818, 2019.

[132] T. Nagarajan and K. Grauman, *Attributes as operators: factorizing unseen attribute-object compositions*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 169–185, 2018.

[133] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, *Solving the multiple instance problem with axis-parallel rectangles*, *Artificial intelligence* **89** (1997), no. 1-2 31–71.

[134] N. Gonthier, S. Ladjal, and Y. Gousseau, *Multiple instance learning on deep features for weakly supervised object detection with extreme domain shifts*, *arXiv preprint arXiv:2008.01178* (2020).

[135] 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** (2013), no. 2 154–171.

[136] S. Kumar, C. Torres, O. Ulutan, A. Ayasse, D. Roberts, and B. Manjunath, *Deep remote sensing methods for methane detection in overhead hyperspectral imagery*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1776–1785, 2020.

[137] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, *Weakly supervised cascaded convolutional networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 914–922, 2017.

[138] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, *Contextlocnet: Context-aware deep network models for weakly supervised localization*, in *European Conference on Computer Vision*, pp. 350–365, Springer, 2016.

[139] R. A. McEver, B. Zhang, C. Levenson, A. Iftekhar, and B. Manjunath, *Context-driven detection of invertebrate species in deep-sea video*, *arXiv preprint arXiv:2206.00718* (2022).

[140] Y. Tian, D. Krishnan, and P. Isola, *Contrastive multiview coding*, *arXiv preprint arXiv:1906.05849* (2019).

[141] G. Xu, P. Kordjamshidi, and J. Y. Chai, *Zero-shot compositional concept learning*, *arXiv preprint arXiv:2107.05176* (2021).

[142] S. Kumar, A. Iftekhar, M. Goebel, T. Bullock, M. H. MacLean, M. B. Miller, T. Santander, B. Giesbrecht, S. T. Grafton, and B. Manjunath, *Stressnet: Detecting stress in thermal videos*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 999–1009, 2021.

[143] D. A. Hudson and C. D. Manning, *Gqa: A new dataset for real-world visual reasoning and compositional question answering*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

[144] 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** (2017), no. 1 32–73.

[145] O. Ulutan, *Attention Models for Activity Detection*. University of California, Santa Barbara, 2019.

[146] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et. al.*, *Language models are few-shot learners*, *Advances in neural information processing systems* **33** (2020) 1877–1901.

[147] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et. al.*, *Palm-e: An embodied multimodal language model*, *arXiv preprint arXiv:2303.03378* (2023).

[148] J. Manyika, *An overview of bard: an early experiment with generative ai*, .

[149] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et. al.*, *Llama: Open and efficient foundation language models*, *arXiv preprint arXiv:2302.13971* (2023).