LOOKING INTO ACTORS, OBJECTS AND THEIR INTERACTIONS FOR VIDEO UNDERSTANDING

by

Effrosyni Mavroudi

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
November 2022

Abstract

Automatic video understanding is critical for enabling new applications in video surveillance, augmented reality, and beyond. Powered by deep networks that learn holistic representations of video clips, and large-scale annotated datasets, modern systems are capable of accurately recognizing hundreds of human activity classes. However, their performance significantly degrades as the number of actors in the scene or the complexity of the activities increases. Therefore, most of the research thus far has focused on videos that are short and/or contain a few activities performed only by adults. Furthermore, most current systems require expensive, spatio-temporal annotations for training. These limitations prevent the deployment of such systems in real-life applications, such as detecting activities of people and vehicles in an extended surveillance videos.

To address these limitations, this thesis focuses on developing data-driven, compositional, region-based video understanding models motivated by the observation that actors, objects and their spatio-temporal interactions are the building blocks of activities and the main content of video descriptions provided by humans. This thesis makes three main contributions. First, we propose a novel Graph Neural Network for representation learning on heterogeneous graphs that encode spatio-temporal interactions between actor and object regions in videos. This model can learn context-aware representations for detected actors and objects, which we leverage for detecting complex activities. Second, we propose an attention-based deep conditional generative model of sentences, whose latent variables correspond to alignments between words in textual descriptions of videos and object regions. Building upon the framework of Conditional Variational Autoencoders, we train this model using only textual descriptions without bounding box annotations, and leverage its latent variables for localizing the actors and objects that are mentioned in generated or ground-truth descriptions of videos. Finally, we propose an actor-centric framework for real-time activity detection in videos that are extended both in space and time. Our framework leverages object detections and tracking to generate actor-centric tubelets, capturing all relevant spatio-temporal context for a single actor, and detects activities per tubelet based on contextual region embeddings. The models described have demonstrably improved the ability to temporally detect activities, as well as ground words in visual inputs.

Thesis Committee

René Vidal (Advisor & Primary Reader)
Herschel Seder Professor
Department of Biomedical Engineering
Johns Hopkins University

Donald Geman
Professor
Department of Applied Mathematics and Statistics
Johns Hopkins University

Rama Chellappa
Bloomberg Distinguished Professor
Department of Electrical and Computer Engineering, and
Department of Biomedical Engineering
Johns Hopkins University

Alan Yuille
Bloomberg Distinguished Professor
Department of Cognitive Science and Department of Computer Science
Johns Hopkins University

Acknowledgments

First of all, I would like to express my gratitude to my advisor Professor René Vidal for his continued guidance throughout my PhD. His extensive knowledge and genuine interest in diverse topics beyond computer vision has been inspiring and has helped shape my research. His generous feedback in our weekly meetings has taught me how to communicate my research effectively and with rigor, and how to conduct high-quality research by paying attention to details. I am particularly grateful for the freedom he gave me to explore my own research ideas and for the opportunities to collaborate in multiple interdisciplinary projects.

I would also like to thank the other members of my committee, Prof. Donald Geman, Prof. Rama Chellappa, and Prof. Alan Yuille, for their mentorship and invaluable feedback on my work over the past years. I am honored to have them on my dissertation committee. I would also like to thank Prof. Sridevi Sarma, Prof. Gregory Hager, Prof. Daniel Robinson and Prof. Jonathan Perry for serving in my Doctoral Oral Board examination committee.

I am grateful to all the researchers I had the luck to collaborate with during my PhD journey. This thesis would not have been in its current state without the generous guidance, continued support, and constant encouragement of

Dr. Benjamin Béjar Haro. The core ideas of every chapter in this thesis have been shaped and refined in our numerous fruitful conversations. I would like to thank Dr. Joshua Gleason and Dr. Carlos Castillo from the University of Maryland for our collaboration on the DIVA project and for generously sharing their data and code, and Prashast Bindal who assisted in deploying our DIVA system during the pandemic. I was also lucky to collaborate with researchers Prof. Bert Tanner and Prof. Elena Kokkoni in the early stages of my PhD in a rewarding, interdisciplinary project that applied activity recognition for pediatric motor rehabilitation.

I would also like to thank Jesse Schettler, Hao Jiang, Vamsi Krishna Ithapu and Lorenzo Torresani for turning my remote internships at Google and Meta into wonderful learning experiences with their mentorship and support. I will always be grateful to my undergraduate advisor Petros Maragos. He has been a role model and with his guidance I have been able to discover a research field I am truly passionate about.

Thanks to all past and present VisionLab members for the support, enlightening discussions and all the fun memories: Dr. Lingling Tao, Dr. Colin Lea, Dr. Shahin Sefati, Dr. Siddharth Mahendran, Dr. Flori Yellin, Dr. Paris Giampouras, Dr. Chong You, Dr. Manolis Tsakiris, Dr. Evan Schwab, Dr. Giann Gorospe, Dr. Guilherme Franca, Dr. Zhihui Zhu, Carolina Pacheco, Ambar Pal, Connor Lane, Pouria Tohidi, Yutao Tang, Kaleab kinfu, Aditya Chattopadhyay, Salma Tarmoun, Darshan Taker, Tianjiao Ding, Ryan Chan, Liangzu Peng, Steven Kan, and many others. My sincere thanks to the members of the LCSR and CCVL vision groups, Jonathan Jones, Tae Soo Kim, Huiyu Wang and many others, for welcoming me in their group meetings and reading groups, making Hopkins such an intellectually stimulating environment.

I would like to thank the staff at CIS and the BME department for making everyday life easier: Anthony Kolasny, Christina Bailey, Missy Kirby, Kimberly Biasucci, Heather Lockard-Wheeler, Hong Lan and Kay Beard. I am also grateful for the funding that enabled my PhD research: NIH grant R01HD87133-01 and IARPA grant D17PC00345.

There are so many friends who supported me during the ups and downs of the PhD journey and I could not have reached this milestone without them. Costas, Mpampis, Nikos, Moschoula, Giorgos, Antonis and Iasonas - thanks for your long-time friendship, for always believing in me, and for your continuous encouragement despite being miles away. Angelos - thanks for not only being one of my first research collaborators, but also a mentor and a constant beacon of positive energy and support. Zeinab, Stephanie, Morgan, Ashley, Elana, Eric, Chin Fu, Leandros, Christos, Kyriaki, Marina, Haris and Bill - thanks for making Baltimore a second home, I am so lucky to have met all of you and I will truly miss you.

Finally, I am more than grateful to my family for their unconditional love, support, and patience during these years. To my parents and aunt for their daily support and patient listening, to my brother for always cheering me up and to my grand-parents for being proud of me always. I dedicate this thesis to them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The amount of unconstrained video data gathered daily by consumer devices and surveillance cameras is exploding. Every 60 seconds, on average, people upload 500 hours of video to YouTube worldwide [13]. Also, more than 70 million surveillance cameras had been installed in the US by 2018 [14]. Therefore, it has become infeasible for humans to analyze the content of such videos.

Monitoring public safety video is particularly challenging for humans, since suspicious activities might occur (1) after long periods of no activities, (2) in multiple regions of the video simultaneously, or (3) in the background far from the video sensor. For example, Figure 1.1 shows example frames from a surveillance video, where in different spatial regions people are talking to each other, a person opens a vehicle trunk, while another vehicle is dropping off a person. Considering that the attention span of a human camera operator has been estimated at only 20 minutes [15], it becomes evident that human camera operators are typically overwhelmed with the amount of video data that they need to monitor. Thus, the need for automated systems that can

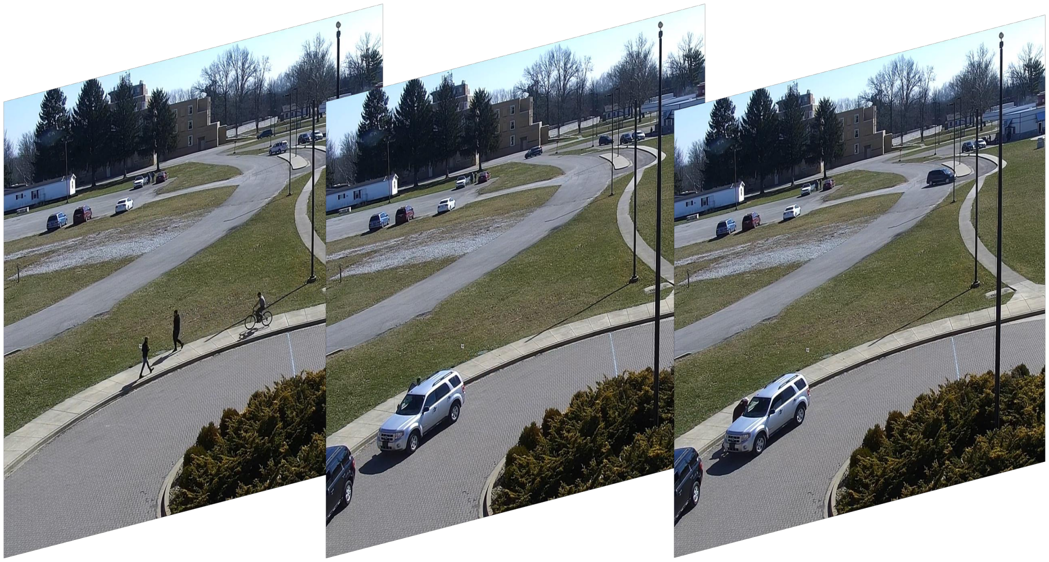
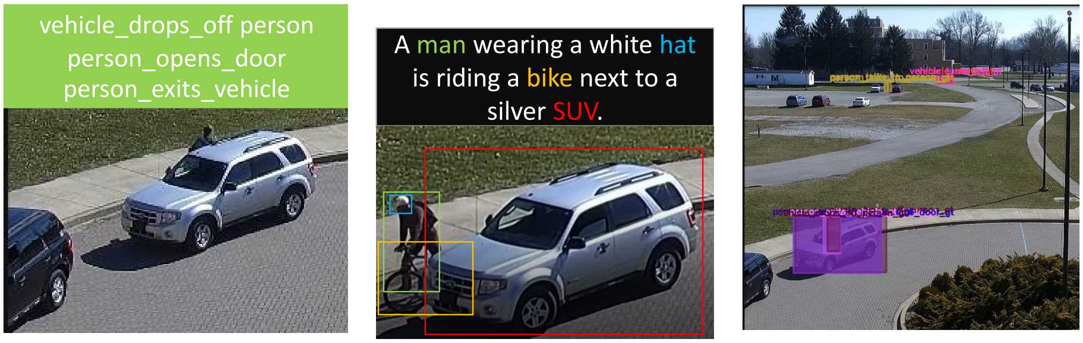


Figure 1.1: Sample frames from an extended surveillance video of the MEVA dataset [10]. Extended videos capture atomic activities (such as *vehicle turning left*), group activities (such as *people talking*), and actor-object interactions (such as *vehicle dropping off person* or *person riding bike*). Activities might occur simultaneously in different spatio-temporal volumes of the video, both in the background and foreground.

understand the rich visual content of *extended surveillance videos* is growing.

Moreover, the increase in life expectancy and population aging has motivated the design of assistive robots that can help elderly individuals [16, 1, 17]. The automatic understanding of the underlying scene and events captured in a video is crucial for enabling these robotic agents to recognize the activities of humans and comprehend their instructions, so as to effectively monitor and assist them. For example, an assistive robot needs to be able to determine where the bottle and the table is in the video recorded by its sensors, in order to "fetch the *bottle* on top of the *table*". It also needs to recognize the activities performed by humans in order to respond appropriately to them (e.g., clean the table after the person finishes *eating a sandwich*).


vehicle_drops_off person
person_opens_door
person_exits_vehicle
A man wearing a white hat
is riding a bike next to a
silver SUV.
person_talks_to_person_gt

(a) Temporal Activity Detection **(b)** Grounded Visual Description **(c)** Activity Detection in Extended Videos

Figure 1.2: An overview of different video understanding tasks tackled in this thesis. (a) Given a video that is spatially-centered around a few actors, the task of temporal activity detection aims to answer *what* activities occur in it and *when*? (b) The task of grounded visual description aims to describe the event in the video with a natural language *sentence* and also localize (*ground*) the semantic entities mentioned in the sentence. (c) Given a video that is *extended both in space and time*, the task of activity detection in extended videos aims to detect what activities are happening and when, as well as localize the interacting entities.

1.1 Video Understanding Tasks

The goal of *video understanding* is to create algorithms that are able to understand the events captured in videos and convey that information to humans.

Traditionally, computer vision researchers have focused on the problem of predicting a human activity label for a short, trimmed video capturing a person performing a single activity. This is a video understanding task known as *Video Classification*. However, simply classifying activities performed only by humans in short videos falls short of matching the way humans see and perceive the world around them, and is not enough for building automated systems that can be deployed in real-life applications such as smart surveillance and assistive robotics. For example, consider the "snapshots" from the surveillance video shown in Figure 1.1. To effectively assist security personnel, an automated video understanding system would ideally be able to automatically and effectively analyze such a long, multi-actor video and (1) *detect* which activities occur and when, (2) convey that information to humans by means of *activity labels* or *sentences*, and (3) *localize* in space and time all the interacting entities (people, vehicles and interacting objects). This requires tackling more complex video understanding tasks, that move towards increasing levels of detail and increasing complexity of the input video data. In this thesis, we focus on the following tasks, summarized in Figure 1.2:

- **Temporal Activity Detection:** Moving beyond temporally trimmed videos, this task aims to detect *which activities* from a predefined set of activities of interest are happening in a long, untrimmed video that is spatially-centered around a few actors and *when*. In other words, this task involves predicting which activities of interest occur at each frame.

- **Grounded Visual Description:** Given a video segment that is spatially-centered around a few actors and shows an event, i.e., a set of related activities, this task aims to *describe* it with a sentence, and *localize* where all the interacting entities (actors and objects) of the described event are. For example, given the video description "a woman is unloading a suitcase from a blue SUV", the goal is to localize each mentioned semantic entity (*woman*, *suitcase*, *SUV*), i.e., draw a bounding box around each mentioned semantic entity at each video frame. This task not only describes videos, but also connects linguistic symbols to external physical objects (which is known as *grounding* [18]), hence the name

Grounded Visual Description.

- **Activity Detection in Extended Videos:** Given an extended (untrimmed, multi-actor) surveillance video, the goal is to *detect* when activities are happening and *localize* the interacting entities of each activity instance.

The rest of this chapter is organized as follows. First, in Section 1.2, we review approaches for representing videos and tackling the aforementioned tasks, in order to place this thesis in context. Then, we discuss the main open challenges in Section 1.3. Last, we conclude by describing our contributions to address these challenges and provide an outline of the thesis in Section 1.4.

1.2 Related Work

In this section, we present an overview of related work structured around: (a) video representations, and (b) approaches for temporal activity detection and for visual grounding. The goal of this global overview is to facilitate our discussion of open challenges that follows and to place this thesis in context. Additional related work that is specific to the tasks we address in later chapters will be presented in the respective chapters.

1.2.1 Video Representations

Video understanding requires capturing high-level semantic concepts and reasoning about their spatio-temporal interactions. This is very challenging for computer vision systems, whose inputs are just sequences of arrays of brightness values. The system then needs to convert this sequence to a video representations that captures spatio-temporal visual cues, such as the appearance of actors, objects and the scene, their motion and their interactions.

Early approaches. Early works on activity recognition relied on *explicit 3D models* of the human body and recognized activities based on the movement patterns of structures such as legs and arms [19, 20, 21]. Since estimating and tracking the 3D pose of non-rigid objects such as the human body is a very challenging problem, *view-based* approaches observed that different activities of a human could be characterized by different spatio-temporal patterns, and leveraged 2D appearance (e.g., 2D silhouettes [22, 23]) or motion (e.g., frame differences or optical flow [24, 25, 26]) instead of 3D structural elements. For instance, Bobick et al. [25] represented human movement in terms of Motion History Images, where pixels that have recently moved are brighter, and then used a *template-matching* approach to match a test video of a human to a database of known movements.

Holistic hand-crafted representations. Although global motion templates might be sufficient for recognizing atomic actions, such as waving hands, that a detected person is performing in front of a homogeneous, static background, they do not generalize to natural videos, which often depict multiple movements and background clutter. This limitation motivated the development of approaches that focus on interest points in the video and compactly describe the appearance and motion in small spatio-temporal volumes around each interest point with *local descriptors*. For example, interest points might correspond to corners or points with non-constant motion [27], and an appearance descriptor could be the histogram of the distribution of edge orientations (HOG), the histogram of optical flow (HOF) or the histogram of oriented optical flow (HOOF) in a spatio-temporal neighborhood of an interest point [28, 29, 30]. Although these descriptors capture local neighborhood characteristics, they are eventually aggregated to represent a video with a holistic Bag-of-Visual-Words [29] or a Bag-of-Dynamical-Systems representation [30, 31, 32]. At a high-level, a dictionary of codewords is constructed, for example by clustering the descriptors and keeping the cluster centers, and then histograms are computed that describe the statistics of the descriptors of a video, e.g., by assigning each descriptor to its closest codeword and counting how many times each codeword appears in the video. This line of work culminated in the seminal Dense Trajectories (DT) [33] and Improved Dense Trajectories (iDT) [34] approaches, which densely sample feature points in each frame, and track them in the video based on optical flow. Multiple descriptors are then computed along the trajectories of feature points to capture shape, appearance and motion information, local features are quantized, and the video is represented by a histogram. Given labeled training videos, classifiers, such as Support Vector Machines, can be trained to predict video labels based on this compact histogram representation.

Holistic deep representations. In 2012, Krizhevsky et al. [35] demonstrated impressive improvement over hand-crafted representations in image classification by training AlexNet, a deep Convolutional Neural Network (CNN) that consists of stacks of trainable convolutional filters. These filters compute spatial feature maps, capturing appearance cues, and can be trained end-to-end, jointly with classifiers. Motivated by the success of CNNs in the image domain, researchers started exploring deep architectures for video classification. Initially, Karpathy et al. [36] directly apply 2D CNNs on frames and design frame-level fusion methods to aggregate features over time. To better model motion cues, Simonyan and Zisserman [37] propose a two-stream approach, where the first stream consists of a 2D CNN applied on each RGB frame, while the second stream applies a 2D CNN on stacked Optical Flow frames. The action predictions of these two separately-trained streams are combined via late fusion. Instead of relying on the additional flow stream to model motion cues, another group of methods introduces spatio-temporal convolutional filters [38, 39, 40], that are applied on short video clips and can learn to encode discriminative spatio-temporal patterns directly from raw pixels. To capture more long-range temporal context, Lea et al. [41, 42, 43] propose to use temporal convolutional networks that capture both fine-grained motion patterns as well as long-term temporal interactions. However, up until 2016, hand-crafted video representations, such as the iDT, were still the video representation of choice and were combined with deep features to yield state-of-the-art results. The I3D model proposed by Carreira et al. [44] led to a significant breakthrough in video understanding performance. It not only combines insights about (a) 3D convolutional filters, (b) two-stream architectures, and (c) effective initialization from 2D networks pretrained on image datasets, but it is also trained on the Kinetics large-scale annotated video dataset. In this thesis, we use pretrained I3D models to compute rich spatio-temporal feature maps of videos. All the aforementioned deep architectures, including more recent Convolutional Neural Networks, such as the SlowFast network [45], or Transformer-based networks that employ self-attention on patches, such as the Multiscale Vision Transformer [46], represent frames or short clips as a whole (similar to the statistical bag-of-words representations). Hence, they cannot scale easily to long, extended videos due to GPU memory constraints and are not designed for recognizing complex, fine-grained activities.

Part-based and region-based video representations. To better model fine-grained activities, e.g., activities that differ in subtle motions or involve human-object interactions, mid-level video representation approaches [47, 48, 49, 50, 51] discover or specify a set of mid-level semantic units (e.g., parts, regions and attributes) and represent a video utilizing them. The semantic units can be 2D patches [52, 47, 53], 3D spatio-temporal patches [54, 48, 55], convolutional action primitives [41], trajectory clusters [56], body-part tubelets [49], or based on object proposals [57, 50, 51]. For instance, Maji et al. [47] represent a video in terms of activated *poselets*, where each poselet is a linear classifier that encodes a part of human pose under a given viewpoint. Zhu et al. [54], generalize that to *actons*, which capture patterns within spatio-temporal volumes, and Mavroudi et al. [49] learn deep *moving poselets*, i.e., mid-level classifiers that capture spatio-temporal configurations of a body part during different phases of an action. Lan et al. [57] represent a video as hierarchy of discriminative spatio-temporal segments, with segments grouped in different levels of granularity, ranging from actors and interacting objects to fine-grained body part movements and individual objects. Interactions between actor and object regions are modeled in [58, 59, 50]. Prest et al. [59] use hand-crafted features for representing human-object interactions. After tracking human and object detections, they model the interaction between a human track and an object track in terms of relative position and motion features. Gkioxari et al. [50] represent actor and object regions with deep features and when predicting an activity they select the most informative secondary, object region, and add its score to the actor region. Baradel et al. [51] compute pairwise relational features between object detections with learnable functions (Relational Networks [60]) and use them to predict activites. However, all these methods represent each semantic unit either with local hand-crafted/deep features, or with features extracted from pairwise interactions, without taking into account the global spatio-temporal interactions among them. Addressing this limitation of region-based approaches will be one of the main contributions of this thesis.

1.2.2 Task-specific Video Understanding Frameworks

Temporal activity detection. While most work on activity recognition focuses on activity classification, by assuming that each input video is manually temporally trimmed and spatially-cropped to capture a single activity, there exist works that attempt to temporally detect activities in long, untrimmed videos that are spatially-centered around a few actors. Recent approaches can be grouped into three families: (a) *frame-wise or clip-wise* classification approaches, (b) two-stage, *proposal-based* approaches, and (c) *end-to-end* approaches. Frame-wise or clip-wise classification approaches [61, 62, 63, 64, 65] follow a bottom-up approach, where they apply classifiers to each individual frame or short clip to detect the presence or absence of each activity class. Then, a post-processing step, involving smoothing and merging, is required to obtain temporal activity detections. These approaches typically incorporate long-term temporal context in the representation of each frame or short clip, by applying deep temporal models (such as Recurrent Neural Networks or Temporal Convolutional Networks). For example, the TGM [64] method constructs a frame representation while focusing at different neighboring temporal segments, by applying a Temporal Convolutional layer whose kernels are constructed based on a mixture of temporal Gaussian distributions. In this thesis, we will also employ a frame-wise classification approach, due to its simplicity, efficiency, and superior performance in the case of temporally-overlapping activities. We will focus on representing frames in terms of contextual region embeddings that capture actor-object spatio-temporal interactions.

In proposal-based approaches [66, 67, 68, 69, 70, 71], a large number of candidate temporal segments (temporal proposals) which are likely to contain an activity is extracted, and then each proposal is separately classified (assigned to one activity class or rejected as background) and temporally refined. Essentially, these approaches are more efficient and sophisticated versions of the sliding window paradigm, where every possible window with different temporal scales needs to be classified. Temporal proposals of pre-defined temporal durations and intervals can be extracted based on sparse dictionaries [66], or by adapting the Region Proposal module of object detector networks [72] to the time domain [69], while variable duration proposals can be extracted based on start/end frame probabilities and segment confidence scores (dense proposals) [73, 71].

Last, end-to-end approaches either integrate proposal generation and classification into a single, end-to-end architecture [74], or approach temporal activity detection as a structured output prediction problem [75].

Grounded visual description. Developing models that can both generate a sentence and link the generated words to their corresponding visual regions is a nascent research area, motivated by a need for more trustworthy and interpretable captioning models [76, 77, 78]. Such models can be seen as an evolution of early image auto-annotation methods [79], and methods for generating either visually grounded storylines [80] or descriptions with grounded and co-referenced people [81]. Zhou et al. [4] proposed using attention-based captioning models for generating sentences. Their GVD model then grounds words based on region attention coefficients.

Visual object grounding. Grounding words (rather than whole sentences [82] or phrases [83, 84]) in images and videos is an active research field in the intersection of vision and language. Early attempts for weakly-supervised visual grounding given textual descriptions of images and videos relied on graphical models [85, 86]. Powered by advances in region proposal generation, a large group of recent methods [87, 88] cast the task as a *Multiple Instance Learning* (MIL) problem. These methods define an image-sentence matching score determined by word-to-region alignments and learn how to correctly match images to sentences using ranking losses. Such methods have also been extended to videos [89, 90, 9] with frame-sentence matching scores and mechanisms to account for missing objects. However, these MIL-based methods cannot both generate sentences and ground objects. This limitation is lifted by the *captioning-based* GVD-Grd method [4], which grounds each word based on region attention coefficients, computed with the previous words as query, combined with region-to-class similarity coefficients. These are obtained by transferring object class knowledge from external datasets.

1.3 Challenges

Despite the significant advances in video understanding presented in the previous section, many challenges still remain. In this thesis we focus on three challenges, described next.

Temporal detection of fine-grained, co-occurring activities. Powered by deep convolutional networks that process whole video frames or short clips, and large datasets with rich human annotations, modern systems are capable of accurately recognizing hundreds of activity classes in benchmark datasets. However, studies have suggested that these holistic representations often focus on the scene context and the objects that are present [91, 92], rather than the fine-grained human movements and human-object interactions. Although leveraging such holistic context is sufficient for recognizing short, simple activities such as *playing the piano*, *swimming*, or *riding a bike*, it might not be sufficient to distinguish between more complex activities with similar appearance and motion patterns. This is the case for a large number of daily activities: they are performed in the same scene context (e.g., in a kitchen or in a parking lot), and involve interactions with the same objects (e.g., opening a vehicle door/exiting a vehicle, opening a door/closing a door/entering through doorway).

One way of augmenting holistic representations is by developing more structured, mid-level representations of videos in terms of semantic parts, such as body parts or detected objects [56, 59, 49, 50]. Since in the past the technology to identify such semantic parts was not robust enough to serve as basis for high-level video understanding, these approaches were mostly applied to small datasets and restricted domains. However, recently deep learning models [93] have made impressive progress in efficiently detecting objects in unconstrained scenes. Building upon these recent advances in object detection, we can utilize the detected actors and objects in a video as the semantic parts. Hence, we need to develop algorithms that can represent videos in terms of the detected semantic regions and their fine-grained spatio-temporal interactions.

Limited localization supervision for visual grounding. Since we aim to design region-based frameworks that can tackle diverse tasks in real-world videos with the ultimate goal of deploying them in real-world applications, their training must be as computationally- and data- efficient as possible. However, the performance of fully-supervised grounded visual description models often depends on the quantity and quality of annotated data. In particular, training a system that can localize the semantic entities that are mentioned in the sentence in the video typically requires per-frame bounding box annotations of the these entities (actors and objects), which are costly and time-consuming to obtain. Due to this reason, many datasets only provide images/videos with textual descriptions without bounding box annotations. Thus, there is a need for approaches that can be trained with *weak, textual supervision*. A line of work for weakly-supervised grounding in videos leverages a set of candidate regions extracted from the video and uses video-to-sentence matching as a downstream training task, where the matching is determined by the compatibility of words and regions. However, these methods cannot both generate descriptions and ground the referred semantic entities of the generated description.

Real-time activity detection in extended surveillance videos. The majority of real-world videos, such as surveillance or sports videos, are extended in time and space, i.e., they are long untrimmed videos that capture multiple actors of various types (people, vehicles, animals) performing multiple activities in various regions of indoor or outdoor scenes. Therefore, efficiently detecting activities of interest in extended videos is a key step towards real-world video understanding.

Activity detection in long, untrimmed videos with a large variation in the number and scale of actors is an under-explored and challenging area. Current systems' performance is indeed shown to degrade with an increasing number of actors and decreasing actor scale [94]. Hence, they are expected to struggle with activity detection in extended videos containing an average of around 30 actors (up to hundreds) of varying scales, including tiny actors, performing multiple activities of varying lengths [10].

Moreover, most activity recognition methods assume that actors are adult humans, although this assumption does not hold in many critical applications. For example, in surveillance videos activities of interest might be performed by both humans and vehicles. Based on the aforementioned assumption, most state-of-the-art methods first detect adults in the video and then process these single person regions and/or tracks, in order to detect activitiess [95, 96, 94, 97, 98, 99, 100, 101, 102]. Such regions of interest cannot handle multiple types of actors and, more importantly, it is not trivial to combine them to obtain the relevant visual context for detecting various types of activities, such as activities involving a single actor, interactions between actors or actor-object interactions. Last, a system that processes every single region or volume associated with a potential actor will be very slow for multi-actor videos.

1.4 Thesis Outline and Contributions

Motivated by the fact that events in natural videos typically arise from spatio-temporal interactions between actors and objects, in this dissertation we aim to tackle the challenges described in the previous section by learning data-driven, task-driven, interaction-aware actor and object representations. We also aim to leverage these representations in novel region-based, compositional frameworks that tackle multiple diverse video understanding tasks, such as temporal activity detection, and grounded visual description. We argue that region-based video understanding frameworks can facilitate the recognition of complex, fine-grained activities by explicitly modeling interactions, can more tightly connect vision and language by facilitating grounding, and can scale to extended videos with multiple actors performing multiple activities in different spatio-temporal regions.

In Chapter 2, we will start by exploring the use of interactions among detected actor and object regions together with the idea of graph-based representation learning for temporal activity detection. Our goal is to develop a representation learning framework to learn task-driven, discriminative, and context-aware region representations by performing representation learning on graphs, which model spatio-temporal interactions among detected actors and objects in a video and interactions among symbolic concepts in a common semantic space. For this purpose, we propose a novel graph neural network for graph-based representation learning on spatio-temporal graphs, called Visual-Symbolic Spatio-Temporal Message Passing Neural Network (VS-ST-MPNN). Given a heterogeneous graph, with multiple types of nodes (such as actors and objects), and multiple types of edges (such as spatial or symbolic), our model computes a context-aware embedding for each region by aggregating context from neighboring, interacting regions, while capturing the nuances of different region and interaction types. We also design region-based video understanding frameworks that leverage the context-aware region representations for two applications, including temporal activity detection in long, untrimmed videos. We then present an experimental evaluation on the CAD-120 and Charades video datasets. Our results demonstrate that our contextual region embeddings lead to improved activity detection performance when compared to local region features or embeddings computed by state-of-the-art Graph Neural Networks, and help us establish the new state of the art in activity detection on both datasets. This proposed graph-based region representation learning method was first presented as a conference paper in [103]. In this thesis, we explore an alternative formulation for heterogeneous visual message passing that employs multi-head attention, and include new experimental results that evaluate different graph design choices and different variants of our model.

In Chapter 3, we study approaches for aligning words in sentences with visual regions, which is the key for tackling the Grounded Visual Description (GVD) and Visual Object Grounding (VOG) tasks. We start in the fully-supervised setting, where we are given training videos with ground-truth textual descriptions and bounding boxes for each referred semantic entity. We introduce the AO-GVD model that extends a popular soft-attention-based grounded video description model [4] by adopting the heterogeneous message passing modules from Chapter 2 to enrich region embeddings with contextual cues about the interactions among actors. We present an experimental evaluation of different variants of this model on the ActivityNet Entities video dataset. The rest of the Chapter focuses on the more challenging training setup, where without any bounding boxes, the goal is to design and train a model that can tackle both GVD and VOG tasks. We account for the lack of ground-truth grounding annotations by introducing a sequence of discrete latent variables, each one of which models a word-to-region alignment. Our model, called Grounded Visual Description - Conditional Variational Autoencoder (GVD-CVAE) allows us to both generate sentences and also infer the latent word-to-region alignments given ground-truth or generated sentences, by taking into account the whole sentence, including the word to be grounded.

To learn the parameters of the latent variable model, we leverage Amortized Variational Inference (AVI) and design a training objective that encourages the model to learn latent variables that capture meaningful word-to-region alignments. Experiments on three benchmark datasets, Flickr30k Entities, ActivityNet Entities and YouCook2, demonstrate both our learned alignment distributions improve upon soft attention in grounding and set the state of the art in two of these datasets. This model is published in [104]. In this thesis, we also propose an extension of our latent variable model, which models the temporal dependency between word-to-region alignments of words in a sentence, and provide a more extensive evaluation of our approach, that includes comparison with additional baselines, additional ablation studies and qualitative results.

In Chapter 4 we move beyond videos that are spatially-centered around an event, and instead focus on activity detection in surveillance videos that are extended both in space and time. For this purpose, we propose an actor-centric framework, that decomposes an extended video into a collection of action-agnostic actor-centric tubelets of interest, where each tubelet is a sub-video that is spatially-centered around an actor. Our proposed approach for tubelet extraction is based on object detection, tracking and region grouping, and not only helps localize activities in space on an actor-level, but also also reduces the number of video regions that need to be processed, reducing the overall processing time. Given a tubelet, one can then detect the activities of the primary actor using the region-based framework from Chapter 2. To better handle activities of low resolution occurring at the background of extended videos, we adapt the region-based activity detection framework to leverage local motion cues in the form of optical flow. We also present an experimental evaluation of our system on the MEVA dataset, and compare with state-of-the-art methods. This system was originally presented in [105]. Finally, we conclude the thesis in Chapter 5.

The contributions of this thesis were motivated by a number of preliminary studies. In particular, we began the exploration of discriminative, region-based video representations for video understanding by using deep appearance and motion features extracted from spatio-temporal volumes defined along body-part trajectories to learn mid-level classifiers called deep moving poselets [49]. We demonstrated that learning such discriminative body part representations jointly with action classifiers and sharing them among action classes enhances full body representations for classification of atomic activities in benchmark datasets. However, these representations are not suitable for recognizing non-atomic, complex activities, such as group activities or human-object interactions. Overcoming this limitation stimulated Contribution 1 of this dissertation that builds video representations based on regions of semantic entities and their interactions. We also conducted preliminary studies on video understanding for extended videos, focusing on (a) action classification in trimmed videos extended in space with a single actor, using videos captured during sessions of rehabilitation therapy for infants and (b) untrimmed time-series of kinematic data, using robotic kinematic data captured during performing surgical tasks with a surgical robot. For the former, we adapted Multiple Instance

Learning techniques for training multi-view classifiers based on weak single-view frame-level labels instead of per-view actor-level labels [106], and we also experimented with a actor detection-based approach [107]. For the latter, we proposed a novel framework that combines a temporal Conditional Random Field model with a frame-level representation based on discriminative sparse coding [108]. On the one hand, our work in infant action classification demonstrated the need for training models without spatio- temporal bounding boxes, which are costly to obtain especially in the medical domain, and the benefits of first localizing the actor and then recognizing the activities. On the other hand, our work on surgical gesture segmentation demonstrated the benefits of properly capturing long-term temporal context and further motivated learning task-driven, discriminative features.

Chapter 2

Graph-based Representation Learning for Region-based Activity Detection in Videos

In this chapter, we focus on the problem of learning structured video representations based on the interactions among actors and contextual objects. We describe how such interactions can be modeled with heterogeneous, spatio-temporal graphs and introduce our novel graph-based region representation learning framework.

We first provide motivation and discuss prior work for modeling spatio-temporal visual interaction cues and semantic interaction cues for tackling complex video understanding tasks. Then, we present our framework for learning task-driven, context-aware actor and object embeddings by leveraging the rich information encoded in heterogeneous graphs that model the interactions among actor and object regions in a video.

Finally, we show how our framework can be applied to tackle fine-grained video understanding tasks that require spatial and temporal reasoning, such as sub-activity and object affordance detection, and temporal activity detection. The learned, graph-structure-aware region embeddings lead to state-of-the-art performance in activity recognition when evaluated on benchmark datasets with natural videos of daily indoors activities performed by humans. In Chapter 4 we will extend our framework to handle extended videos that capture a large number of actors of multiple types (humans and vehicles) in indoor and outdoor settings.

2.1 Motivation and Overview

The field of video understanding has been moving towards increasing levels of complexity beyond simple classification of a single action performed by a single person in a short, trimmed video. A plethora of real-world applications require video understanding systems that can automatically detect multiple complex activities performed by multiple actors interacting with objects in untrimmed videos, or systems that can describe videos with natural language sentences.

Modeling contextual interactions is crucial for addressing such video understanding tasks. As a motivating example, consider the video frames in Fig. 2.1). Recognizing the activity *watching tv* requires capturing the spatial interaction between the actor and the object, while *taking a cup* requires temporal reasoning as well, considering the change of the pose of the actor and the change of the position of the object. In addition to such local visual spatio-temporal interactions, video understanding can be aided by interactions in a

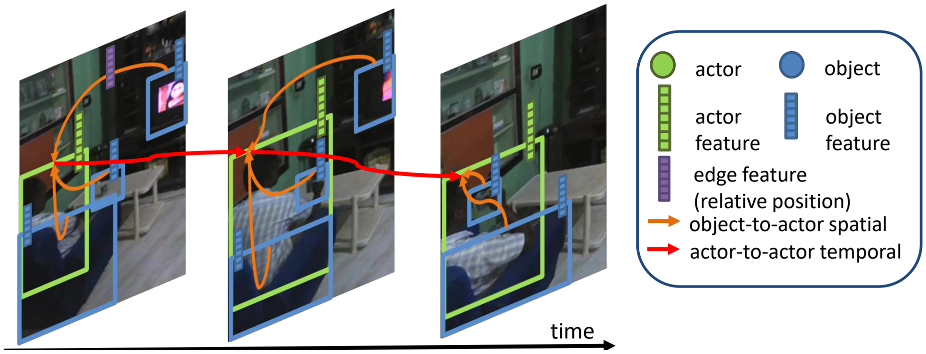


Figure 2.1: Interactions among video regions are powerful video understanding cues. Video regions can correspond to actors (e.g., humans) or contextual objects (e.g., tv or sofa). Each region type is associated with not only different semantics but also features of potentially different dimensionality. Furthermore, interactions between regions can occur in the same time frame (*spatial interactions*) or between different frames (*temporal interactions*). All these potential interactions between regions in a video can be encoded using heterogeneous graphs. (Sample frames from the Charades [2] dataset.)

global semantic space, such as prior knowledge about co-occurring or semantically similar activities or objects in videos. For example, the activity *sitting in a chair* frequently co-occurs with the activity *watching tv*. Therefore, there is a need to develop algorithms that can effectively model spatio-temporal visual and semantic context.

One way of capturing such context is to use graphs whose nodes represent scene entities and whose edges represent relationships between scene entities. Such graph-based models have a rich history in computer vision, including probabilistic graphical models, stochastic grammars, and compositional systems. However, traditional graph-based approaches to video understanding based on probabilistic graphical models [109, 1, 110, 111] focused mainly on modeling context at the level of symbols rather than visual signals or representations. Moreover, while stochastic grammars [112] used appearance models to bridge the gap between signals to symbols and modeled rich hierarchical relationships, they used hand-crafted features to represent regions and only modeled simple, pairwise spatial and functional relationships between entities at the same representation level. Hence, their feature representations and horizontal relationships remained fairly limited.

Recent advances in deep learning have enabled *representation learning on graph-structured data* using deep architectures called Graph Neural Networks (GNNs). The central idea behind GNNs is to learn graph-structure-aware node embeddings via the repeated aggregation of information from local node neighborhoods using non-linear transformations [3]. GNNs have recently been designed for refining the local region features of semantic entities in videos based on the spatio-temporal context captured by visual st-graphs [82, 113, 114, 115].

Although representation learning on visual st-graphs has lead to significant advances in video understanding there are four key limitations of state-of-the-art approaches that prevent them from fully exploiting the rich structure of these graphs. First, the visual st-graph is a *heterogeneous* graph that has distinct node types (*actor*, *object*, etc.) and distinct edge types (*object-to-actor spatial*, *actor-to-actor temporal*, etc.), with each type being associated with a feature of potentially different dimensionality and semantics, as shown in the example of Fig. 2.1. However, most GNNs assume that all nodes/edges are of the same type. Therefore, recent attempts at explicitly modeling actors and objects have resorted to applying separate GNNs for each node type [116] or edge type [82, 117]. Second, most methods operate on a graph of fixed edge weights [113] with dense connectivity. In practice, only a few of the edges capture meaningful interactions. Third, current approaches do not incorporate edge features, such as *geometric relations between regions*, for updating the node representations. Finally, despite modeling local visual context, existing approaches do not reason at a global video level or exploit semantic interactions, which have been shown to be beneficial in the image recognition domain [118, 119].

In this thesis, in an effort to address these limitations, we propose a novel GNN model, called Visual-Symbolic Spatio-Temporal Message Passing Neural Network (VS-ST-MPNN), that performs representation learning on heterogeneous visual-symbolic graphs to obtain context-aware representations of actors and objects. Our model handles heterogeneous graphs by employing *learnable message functions that are specialized for each edge type*. We also adapt the visual edge weights with an *attention mechanism* that is specialized for each type of interaction. For example, an actor node will separately attend to actor nodes at the previous frame and object nodes at the current frame. Furthermore, we employ *edge features* to refine the actor and object representations, as well as to compute the attention coefficients that determine the connection strength between regions. Intuitively, nodes which are close to each other or are interacting should be strongly connected. Finally, one of our key contributions is incorporating an attributed *symbolic graph* whose nodes correspond to semantic concepts, such as actions described by word embeddings, and whose edges capture semantic concept relationships, such as co-occurrence. We represent a video as a graph consisting of two major subgraphs: (a) a video-specific visual spatio-temporal subgraph that encodes potential local spatio-temporal visual interactions among actor and object regions in a video, and (b) a symbolic subgraph that encodes global interactions in a semantic interaction space and is shared among all videos of a dataset. We fuse the information of the two subgraphs with learnable connections between their nodes, and we employ graph convolutions to learn global semantic interaction-aware features. Importantly, we do not require ground truth annotations of objects, tracks, scene graphs, or semantic labels for each visual node.

Preliminary results or the VS-ST-MPNN framework were first published in [103]. In this thesis, we extend the framework by proposing an alternative message passing mechanism for refining the visual node embeddings inspired by Transformers [120]. We also provide a more in-depth discussion of the algorithm with more details about the region-based video understanding framework, the construction of the input heterogeneous graph and training objectives. We also conduct additional, extensive ablation studies to thoroughly assess the effectiveness of each graph design choice and each message passing component of our model. Importantly, we expand the types of input graphs that we consider, including visual graphs with temporal edges connecting regions in both past and futures frames and symbolic graphs with latent semantic concepts initialized with random node attributes.

Consequently, our contributions are as follows:

1. **Modeling region interactions with heterogeneous graphs**: We model

contextual cues for video understanding by combining a symbolic subgraph, capturing semantic interactions, with a visual spatio-temporal subgraph, encoding interactions between actors and objects (Section 2.3.1).

2. **Graph-based representation learning on heterogeneous spatio-temporal graphs**: We introduce a Message Passing Neural Network that is tailored to our input heterogeneous graphs with edge-type-specific message functions and node-type-specific node update functions, both of which take into account edge features. We also describe how the VS-ST-MPNN can be trained jointly with task-specific recognition networks to learn task-driven, context-aware region embeddings (Section 2.3.2).

3. **Experimental evaluation**: To demonstrate the effectiveness and generality of our method, we evaluate it on tasks such as temporal activity detection, and sub-activity and object affordance detection on two challenging, benchmark datasets and show that it achieves state-of-the-art performance (Section 2.5).

2.2 Related Work

In Section 1.2, we provided a review of holistic video representations, including a discussion of convolutional networks for representation learning on 2D or 3D grids, and deep temporal models for representation learning on time-series. Here, we provide additional discussion of approaches that model actor-object interactions and semantic interactions with a focus on methods that perform representation learning on graphs, and draw connections between our graph neural network and related models in other research areas.

2.2.1 Modeling Human-Object Interaction and Semantic Interaction Cues

Context and its role in computer vision has been studied for a long time [121, 122]. Our proposed model is related to approaches that capture context in terms of contextual objects, human-object interactions and semantic interactions. These approaches can be grouped into four families.

The first family includes *probabilistic graphical models* of human activities and videos, where graphs are used to model statistical dependencies among region labels [109, 123, 1, 110, 111]), or among the activity label and latent part labels [124]. Gupta et al. [123] aim to simultaneously recognize objects, estimate manipulation movements and object reactions by taking advantage of the contextual information provided by each element to the others via a Bayesian Network. Koppula et al. [1] use a graph to represent the joint probability of sub-activity and object affordance labels in a video with a Conditional Random Field (CRF). The energy of a particular assignment of sub-activity and object affordance labels to humans and objects is a function of the labels and hand-crafted region and interaction features. Other approaches move beyond Bayesian Networks and CRFs, and utilize deep hierarchical context models [111]. Another related line of work designs representations of visual scenes based on stochastic grammars and their associated And-Or graphs [112, 125, 126, 127], which capture hierarchical decompositions (e.g., a scene is composed of foreground and background, the foreground is composed of objects, each object can be composed of parts etc.), as well as functional and spatial relationships among structures at each level of the hierarchy. Our work is more related to the horizontal links of And-Or graphs at the terminal node level (region proposal level). While prior work has focused on rich, vertical compositional structures, capturing the hierarchy of labels, their horizontal links capture simple spatial, geometric relations (such as occlusions) or functional relations (such as *carry*, *support*, etc.) between pairs of nodes. In addition, each object node is represented by a detector that can recognize the object based on primitives or hand-crafted local region features. Instead, our regions are represented with rich, discriminative features extracted from deep networks, which we aim to further refine based on long-range spatio-temporal visual and semantic interactions captured in heterogeneous visual-symbolic graphs.

The second group of approaches treats the activity recognition task as a *graph-matching* problem, wherein to recognize the activity for a video, they extract a graph from it and find its closest match from a set of graph exemplars per activity class [128, 129]. For example, Brendel et al. [128] represent an activity as a spatio-temporal graph whose nodes correspond to multi-scale spatio-temporal video segments. Then, given the graphs of training videos, they learn an archetype graph for each activity via least-squares optimization. A new video is then parsed by matching its extracted graph with the closest activity archetype graph.

The third family encodes semantic dependencies among activity labels with *symbolic graphs*, whose nodes correspond to activity labels. These approaches build upon a long line of work on exploiting external knowledge encoded in label relation graphs for visual recognition tasks. Semantic label hierarchies, such as co-occurrence, have been leveraged for improving object recognition [130, 131, 132, 133], multi-label zero-shot learning [134] and other image-based visual tasks [135, 136]. Much fewer papers utilize knowledge graphs for video understanding [137, 138, 139], possibly due to the limited number of semantic classes in traditional video datasets. For example, the SINN [139] performs graph-based inference in a hierarchical label space for action recognition. Although we also leverage such symbolic graphs, we use them to define a semantic space that facilitates the computation of semantic context-aware region embedding, rather than directly performing inference on them.

Most related to our method is the last group of approaches that compute interaction-aware mid-level representations of videos that are then mapped to activity labels. These representations capture pairwise interactions of actors with other regions [58, 59] and/or scene context [58, 96]. Prest et al. [59] use hand-crafted features for representing human-object interactions. After tracking human and object detections, they model the interaction between a human track and an object track in terms of relative position and motion features. Baradel et al. [51] compute pairwise relational features between object detections with learnable functions (Relational Networks [60]) and use them to predict activites. However, these methods only model pairwise interactions, while our method builds upon a recent line of methods that encode visual interactions with graphs and learn graph-structure aware features via end-to-end training.

2.2.2 Graph-based Representation Learning for Video Understanding

Representation learning on visual spatio-temporal graphs. A growing line of work aims to extract structured representations of videos by capturing interactions among visual units via graphs and performing representation learning on these graphs. One of the first approaches applying a deep network on a visual graph for video understanding is the Structured Inference Machine [140], which refines actor features with message passing, and filters out spurious interactions with trainable gating functions, but only captures spatial relationships between actors. Another early approach is the S-RNN [141], which introduces the concept of weight-sharing between nodes or edges of the same type, but does not iteratively refine node representations.

With the advent of Graph Neural Networks (GNNs), many researchers have explored modeling tracklets [82], feature map columns [96, 142, 143], or object proposals [144, 51, 113, 114, 115, 117] as graph nodes and using off-the-shelf GNNs, such as MPNNs [145] and GCNs [3] to refine the node or edge representations, obtaining significant performance gains. However, since most these existing GNNs are not designed to handle distinct node and edge types, applying them to visual spatio-temporal graphs requires treating every node and edge in the same way [113, 51, 146, 114, 142], or focusing only on one edge type, such as actor-actor [144] and actor-object [96], or using separate GNNs for each node type [116] and edge type [113, 82]. In doing so, they completely ignore or sub-optimally handle the rich, heterogeneous visual graph structure. Recently, Arnab et al. [117] propose a variant of MPNNs that has spatial and temporal message functions and apply it to spatio-temporal action detection and scene graph generation. Similarly, Bertasius et al. [147] propose a Vision Transformer which is applied on top of video patches and utilizes spatial and temporal attention mechanisms. In contrast, our proposed method is more general and can be directly applied to any heterogeneous spatio-temporal graphs with an arbitrary number of node and edge types. The benefit of such fine-grained modeling has already been established in fields such as computational pharmacology and relational databases [148, 149, 150, 151], but remains relatively unexplored in computer vision.

Furthermore, most of the graph-based representation learning approaches applied on visual graphs do not use both node and edge features, and cannot handle directed edges. Similar to [142], our method iteratively adapts the visual edge weights, but employs an attention mechanism that is specialized for different edge types and takes edge features into account. The GPNN [11] also utilizes edge features both for refining the scalar edge weights connecting the visual nodes and for refining the node embeddings, but it does not iteratively refine the edge features to capture richer interactions as we do. Our proposed edge-type-specific and edge-feature-conditioned neighborhood aggregation mechanism can also be seen as an extension of edge-conditioned graph convolution [152, 153].

Representation learning on semantic graphs. Our framework is also related to a recent line of work that leverages symbolic graphs that are shared among all images in a dataset by projecting local visual features to the common semantic space defined by the symbolic graph and using graph convolutions to model global semantic interactions. For example, Liang et al. [118] enhance feature maps extracted from images by using a symbolic graph, while [154, 155] use a latent semantic interaction graph. In contrast, we seek to improve the representation of visual subgraph nodes rather than enhance convolutional features on a regular grid, and our model is proposed for the video domain.

Representation learning on hybrid graphs. Fusing information from multiple graphs (such as our visual and symbolic subgraphs) using GNNs is an exciting new research direction [156, 157, 158, 159, 160]. Similar to our approach, Chen et al. [119] combine a visual graph instantiated on objects with a symbolic graph and perform graph representation learning, while [161] enforce the scalar edge weights between visual regions to be consistent with the edges of the symbolic graph. However, they operate on simple spatial graphs and assume access to semantic labels of regions during training. In contrast, we operate on hybrid visual-symbolic spatio-temporal graphs and we do not require known correspondences between visual and symbolic nodes during training.

2.3 Graph-based Actor and Object Representation Learning

We propose a general framework for region-based video understanding that incorporates structured information from an input heterogeneous graph to

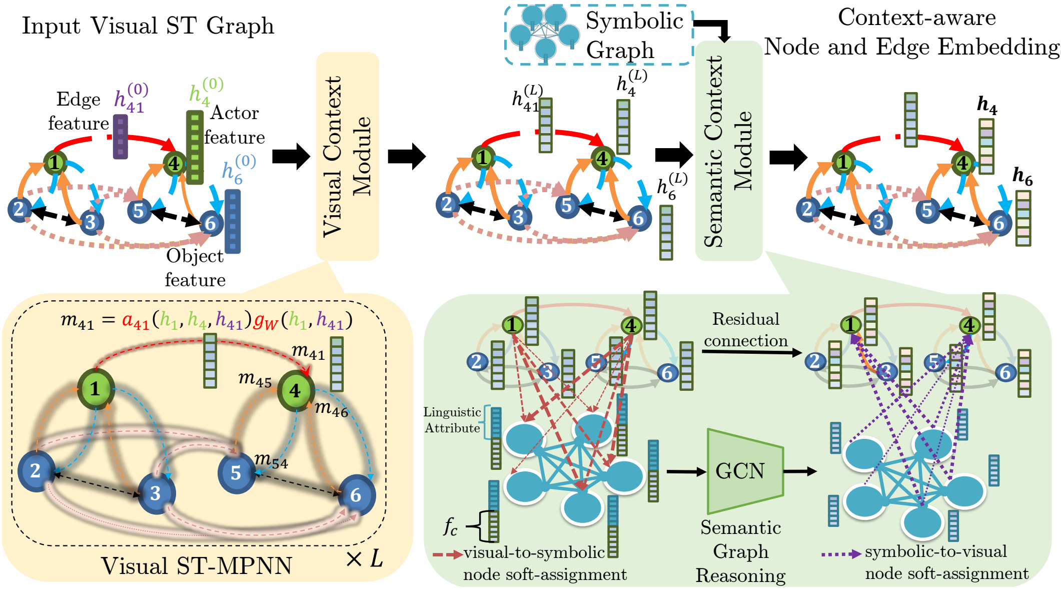


Figure 2.2: Overview of our VS-ST-MPNN model that performs representation learning on heterogeneous, visual-symbolic graphs. Given an input video that is represented as a heterogeneous graph with visual and symbolic subgraphs (as described in Sec. 2.3.1), our framework has two modules that integrate context in the local representations of its nodes and edges: (a) a Visual Context Module (Sec. 2.3.2.1) that performs L^v rounds of node and edge updates on the visual subgraph, with specialized neighborhood aggregation functions that depend on the type of an edge, and (b) a Semantic Context Module (Sec. 2.3.2.2) that integrates visual evidence with semantic concepts encoded in the symbolic graph and learns global semantic interaction-aware features. The final node and edge embeddings can be appropriately pooled and fed to recognition networks for tackling various video understanding tasks.

address a diverse set of video understanding tasks. Given an input RGB video with T_f frames, height H and width W, $I \in \mathbb{R}^{3\times H\times W\times T_f}$, and a set of M regions $R = \{\mathbf{r}_i \in \mathbb{R}^4\}_{i=1}^{M}$ (either manually annotated or extracted with an object detector trained on an external image dataset), we construct a *heterogeneous, attributed* graph $G(R, I)$ that consists of two connected subgraphs: (a) a *visual spatio-temporal subgraph* that encodes potential local spatio-temporal interactions among actor and object regions in the video, and (b) a *symbolic subgraph* that encodes global interactions in a semantic space and is shared among all videos of a dataset. We are interested in leveraging that graph to learn a discriminative model $p(Y|I, R)$, that models the probability of a labeling Y from a task-specific structured label space $\mathcal{Y}$ (e.g., multiple activity labels per frame or a sentence describing each video) given the input video I and regions R. In particular, our primary goal is to refine the local features of actors, objects and their interactions based on the contextual information captured in the input heterogeneous graph. We refine those local features with a novel GNN, called VS-ST-MPNN, shown in Fig. 2.2. Our VS-ST-MPNN is tailored to exploit the rich structure of the heterogeneous input graph by (1) learning *node-* and *edge-type-specific* neighborhood aggregation functions, (2) *adapting the scalar edge weights* with edge-type-specific attention mechanisms, without assuming a fixed, known graph structure such as actor and object tracks or ground-truth actor-object interactions, (3) utilizing geometric region relations and edge attributes both in the node updates and attention mechanism, and (4) incorporating a soft-assignment module for connecting visual and semantic nodes without requiring access to ground-truth semantic labels of regions. Next, we describe our framework in more detail.

2.3.1 Modeling Interactions with Heterogeneous Graphs

Given a set of semantic regions (actors and objects) and their local features, we build a structured representation of a video by representing it as a heterogeneous, attributed, directed graph that encodes interactions among actors and objects. The graph, denoted as $G(R, I) = (\mathcal{V}, \mathcal{E}, \mathcal{H}^{(0)}, \mathcal{Q}^{(0)})$, consists of a set of

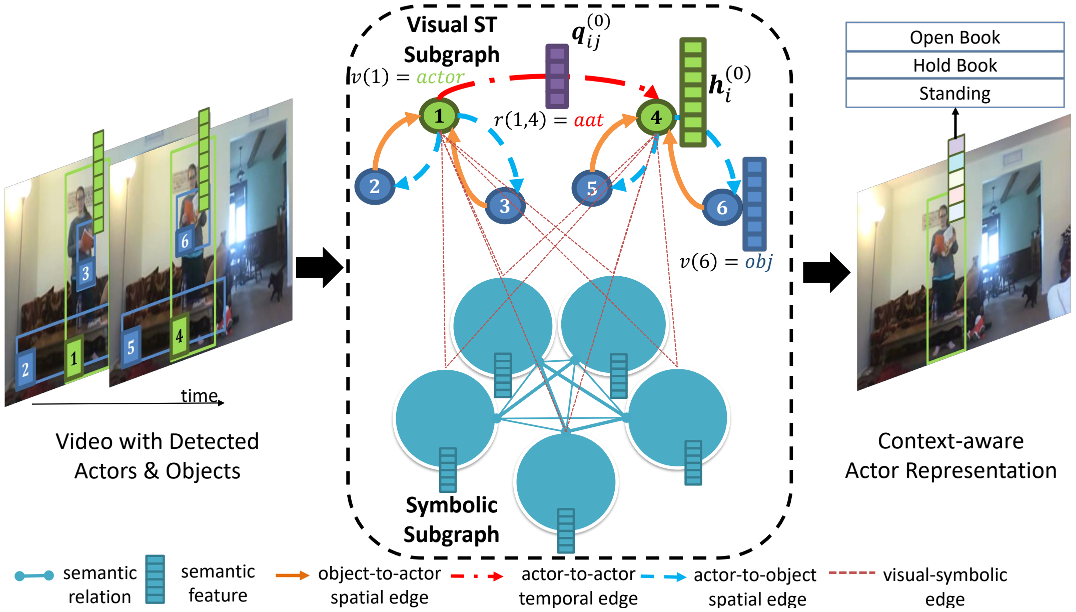


Figure 2.3: Given a video with actor and object regions, we aim to learn region representations that capture (1) local visual spatio-temporal interactions between actors and objects, and (2) global semantic interactions, such as action co-occurrences. These powerful contextual cues are encoded in a heterogeneous attributed graph with multiple types of nodes (e.g., actor, object, symbolic) and edges (e.g., actor-to-object spatial, visual- symbolic). In this work, we perform representation learning on such heterogeneous graphs to obtain task-driven, context-aware region representations. (Sample frames from the Charades [2] dataset.)

nodes $\mathcal{V} = \{1, \ldots, |\mathcal{V}|\}$ of different types, a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of different types, a set of initial node attributes $\mathcal{H}^{(0)} = \{\mathbf{h}_i^{(0)}\}_{i \in \mathcal{V}}$, and a set of initial edge attributes $\mathcal{Q}^{(0)} = \{\mathbf{q}_{ij}^{(0)}\}_{(i,j) \in \mathcal{E}}$. Each node is associated with a single node type from a set $\mathcal{T}$ of possible types, and each pair of nodes is connected by an edge which is associated with an edge type from a set $\mathcal{R}$ of possible types (relationships). These associations are defined via a node type mapping function $\nu : \mathcal{V} \to \mathcal{T}$, and an edge type mapping function $r : \mathcal{E} \to \mathcal{R}$. Note that the dimensionality of the initial attribute describing each node (and edge) may vary depending on their type: $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d_{\nu(i)}}$. Weighted heterogeneous graphs are additionally associated with a scalar weight function $w : \mathcal{E} \rightarrow \mathbb{R}$. The graph connectivity can be compactly represented as a set $\mathcal{A} = \{A_\epsilon\}_{\epsilon=1,\ldots,|\mathcal{R}|}$ of adjacency matrices, where $(A_\epsilon)_{ij} = w_\epsilon(i,j)$ denotes a directed edge of type ϵ from node i to node j with scalar weight $w(i,j)$. The edge-type-conditioned neighborhood $\mathcal{N}_\epsilon(i) = \{j : (i,j) \in \mathcal{E}, r(i,j) = \epsilon\}$ is defined to contain all vertices which are connected to node i via an incoming edge of type ϵ.

In our framework, we focus on heterogeneous graphs that consist of a spatio-temporal visual subgraph and a symbolic subgraph (Fig. 2.3). In particular, we represent a video as a graph consisting of two major subgraphs: (a) a *video-specific* visual spatio-temporal subgraph, whose nodes $\mathcal{V}^v$ correspond to actor and object regions in a video and whose edges $\mathcal{E}^v$ encode potential local spatio-temporal visual interactions among these regions, and (b) a symbolic subgraph, whose nodes $\mathcal{V}^s$ correspond to semantic concepts and whose edges $\mathcal{E}^s$ encode global interactions in a semantic interaction space and is *shared among all videos of a dataset*. Therefore, each node of our hybrid visual-symbolic graph is associated with one of three node types: actor (a), object (o) or symbolic (s), i.e., $\mathcal{T} = \{a, o, s\}$, while our set of edge types can be split into three subsets $\mathcal{R} = \mathcal{R}^v \cup \mathcal{R}^{vs} \cup \mathcal{R}^s$, where $\mathcal{R}^v$ is the set of visual spatio-temporal edge types, such as *actor-to-object-spatial* or *actor-to-actor-temporal*, $\mathcal{R}^{vs}$ is the set of types associated with edges connecting visual nodes with symbolic nodes subgraphs, and $\mathcal{R}^s$ is the set of symbolic edge types. Accordingly, the set of adjacency matrices can be split into three subsets: $\mathcal{A} = \{\mathcal{A}^v_\epsilon\}_{\epsilon\in\mathcal{R}^v} \cup \{\mathcal{A}^{vs}_\epsilon\}_{\epsilon\in\mathcal{R}^{vs}} \cup \{\mathcal{A}^s_\epsilon\}_{\epsilon\in\mathcal{R}^s}$, and we can *infer the types of the connected nodes from the type of the edge that connects them*. We emphasize that each connected pair of nodes in our graph is connected with a single edge of a single type, thus we can *infer the type of the connected nodes from the type of the edge that connects them*. This is a property of our input graph that is crucial for the design of our graph-based representation learning model. Next, we describe general steps for constructing the input visual-symbolic graph (details for the input visual-symbolic graphs that we use in each task will be provided in Sections 2.5.1.3 and 2.5.2.3).

Visual spatio-temporal subgraph. The main insight of our approach is that events in natural videos typically arise from spatio-temporal interactions between actors and objects. Suppose we are given ground-truth bounding boxes or object detections at each frame of an input video with T frames. These regions constitute the nodes $\mathcal{V}^v$ of the visual spatio-temporal subgraph.

We first need to define the node assignment function $\nu : \mathcal{V}^v \rightarrow \mathcal{T}^v = \{a, o\}$ that assigns the actor or object type to each visual node. Second, we add directed edges that connect visual nodes and model latent local spatio-temporal interactions. Edge types $\mathcal{R}^v$ are based on the types of nodes that are connected and whether the nodes correspond to the same frame or not (e.g., object-to-actor spatial (*obj-act-s*), actor-to-object spatial (*act-obj-s*), object-to-object spatial (*obj-obj-s*), actor-to-actor temporal (*act-act-t*)). Next, we compute initial attributes of nodes and edges. A node-type-specific region feature extraction function $\phi_{\nu(i)}$ is used to compute initial node attributes for each visual node:

$$\mathbf{h}_i^{(0)} = \phi_{\nu(i)}(\mathbf{r}_i, I) \in \mathbb{R}^{d_{\nu(i)}}. \tag{2.1}$$

Similarly, each visual directed edge $(i, j) \in \mathcal{E}^v$ is described by an initial edge attribute $\mathbf{q}_{ij}^{(0)} \in \mathbb{R}^{d_{r(i,j)}}$ whose dimension depends on the edge type $r(i, j)$ and is computed with an edge-type-specific function $\phi_{r(i,j)}(\mathbf{r}_i, \mathbf{r}_j, I)$:

$$\mathbf{q}_{ij}^{(0)} = \phi_{r(i,j)}(\mathbf{r}_i, \mathbf{r}_j, I) \in \mathbb{R}^{d_{r(i,j)}}. \tag{2.2}$$

Fig. 2.2 illustrates a toy example with a visual graph consisting of $A = 1$ actor and $O = 2$ object detections per timestep for $T = 2$ timesteps. Depending on the task we can use different node type assignment function, edges, node and edge attributes to represent potential interactions, regions and spatial relationships, respectively. For example, for the task of activity detection, we use a visual subgraph that models the interactions of an actor (person) with contextual object nodes in each frame, as well as the change in the actor's appearance and interactions with other actors among different timesteps.

Symbolic subgraph. Let us recall that $\mathcal{V}^s$ and $\mathcal{E}^s$ denote the symbolic subgraph's node set and edge set, respectively. For simplicity we assume that all symbolic nodes have type s, while symbolic edges have type $s - s$, i.e., $\mathcal{T}^s = \{s\}$ and $\mathcal{R}^s = \{ss\}$. Each symbolic node i is initialized with a semantic attribute $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d_s}$ of dimensionality d_s using a feature extraction function $\phi_s(\cdot)$. Edges in the symbolic graph are associated with scalar weights, which encode semantic relationships. These edge weights are summarized in the fixed adjacency matrix $\mathcal{A}_{ss}^s \in \mathbb{R}^{|\mathcal{V}^s| \times |\mathcal{V}^s|}$. Note that symbolic edges are not associated with edge attributes.

Our framework supports various types of symbolic subgraphs. For example, for the task of activity detection, we choose to use a subgraph with symbolic nodes corresponding to action labels, node attributes initialized with word embeddings of the action labels, and scalar edge weights capturing action co-occurrences.

Visual-Symbolic edges. As their name suggests, these directed edges $\mathcal{E}^{vs}$ connect the nodes of the visual subgraph with the nodes of symbolic graph. Although latent, we can specify a priori the allowed visual-symbolic node connections. For example, for the task of activity detection, where we choose to use symbolic nodes corresponding to action semantic concepts, we only connect actor nodes and symbolic nodes. Similar to the symbolic edges, we also do not associate an edge attribute with the visual-symbolic edges.

2.3.2 VS-ST-MPNN: Representation Learning on Visual-Symbolic Spatio-Temporal Graphs

Our proposed VS-ST-MPNN iteratively refines the representations of detected actors, objects and their interactions based on the contextual information captured in the two subgraphs of the input heterogeneous graph. It extends the original Message Passing Neural Network (MPNN) framework [145], which we discussed earlier in this chapter, so that it can take into account the various node and edge types, as well as the special subgraph structure.

The first key idea of our VS-ST-MPNN is to use edge-type-specific message functions, edge-type-specific edge update functions, and node-type specific node update functions, instead of using the same functions with the shared parameters for all node/edge types as in the original MPNN. Given the input graph representing a video I with initial node and edge attributes, $\{\mathbf{h}_i^{(0)}\}_{i\in\mathcal{V}}$

and $\{\mathbf{q}_{ij}^{(0)}\}_{(i,j)\in\mathcal{E}}$, respectively, we introduce novel graph propagation rules to iteratively compute context-aware node and edge embeddings. In particular, at the l-th iteration, our VS-ST-MPNN first computes a message along each edge (i, j) with a an edge-type-specific and layer-specific learnable message passing function $\text{MSG}_{r(i,j)}^{(l)}(\cdot)$ that takes as inputs the current embedding of the receiver node $\mathbf{h}_i^{(l-1)}$, the current embedding of the sender node $\mathbf{h}_j^{(l-1)}$, and the current edge embedding $\mathbf{q}_{ij}^{(l-1)}$ (if available). We would like to emphasize again, that in contrast to the original MPNN network [145], each one of the message passing functions $\{\text{MSG}_{\epsilon}^{(l)}(\cdot)\}_{\epsilon\in\mathcal{R}}$ is a neural network parameterized with learnable weights $\theta_{\epsilon}^{(l)}$ that depend on the edge type ϵ. Then, the embedding of the i-th node is updated with a node-type-specific node update function $\text{NodeUPD}_{\nu(i)}^{(l)}(\cdot)$ after aggregating messages from its neighborhood $\mathcal{N}(i) = \{j : (i, j) \in \mathcal{E}\}$ with an aggregation function $\text{AGGR}^{(l)}(\cdot)$. We also update the embedding of every attributed edge (i, j) with an edge-type-specific edge update function $\text{EdgeUPD}_{r(i,j)}^{(l)}(\cdot)$. Formally, the (l)-th layer of VS-ST-MPNN is defined by the following graph propagation rules:

$$\mathbf{m}_{ij}^{(l)} = \text{MSG}_{r(i,j)}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, \mathbf{q}_{ij}^{(l-1)}\right) \in \mathbb{R}^{d_{r(i,j)}^{(l)}}, \tag{2.3}$$

$$\mathbf{h}_i^{(l)} = \text{NodeUPD}_{\nu(i)}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \text{AGG}^{(l)}\left(\left\{\mathbf{m}_{ij}^{(l)}\right\}_{j\in\mathcal{N}(i)}\right)\right), \tag{2.4}$$

$$\mathbf{q}_{ij}^{(l)} = \text{EdgeUPD}_{r(i,j)}^{(l)}\left(\mathbf{q}_{ij}^{(l-1)}, \mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}\right). \tag{2.5}$$

The second key idea of our VS-ST-MPNN is to perform message passing in stages instead of sending messages and updating all nodes of the input graph in parallel. The intuition is to first refine the region representations based on local spatio-temporal interactions between actors and objects, and then further refine them based on global semantic interactions on the latent semantic space embodied in the symbolic subgraph. In particular, in the first stage we iteratively compute messages along the visual edges and refine the visual nodes based on local visual spatio-temporal context for L^v iterations. These iterations of visual message passing (or equivalently first L^v layers of the VS-ST-MPNN) are called the Visual Context Module (VCM). During these iterations there are no messages computed alongside the visual-symbolic or symbolic edges, and the symbolic node attributes are updated with the identity transformation, i.e.

$$\left\{\mathbf{m}_{ij}^{(l)}\right\}_{(i,j)\in\mathcal{E}\setminus\mathcal{E}^v} = 0, \quad l = 1, \ldots, L^v, \tag{2.6}$$

$$\mathbf{h}_i^{(l)} = \mathbf{h}_i^{(l-1)}, i \in \mathcal{V}^s, \quad l = 1, \ldots, L^v. \tag{2.7}$$

In the second stage, we update the symbolic nodes and edges as follows. First, at the $(L^v + 1)$-th iteration, messages are computed alongside the visual-to-symbolic edges to update symbolic nodes with visual evidence. Then, we perform L^s iterations of symbolic message passing on the symbolic graph to model interactions on the shared semantic interaction space. During these iterations, there are no messages computed alongside the visual-symbolic or visual edges and the visual node/edge attributes are updated with the identity transformation. Finally, we compute a message alongside each symbolic-to-visual edge to update the visual nodes (actors and objects) with the semantic-context-aware embeddings. These iterations of visual-symbolic, symbolic and symbolic-visual message passing (or equivalently last $(L^s + 2)$ layers of the

VS-ST-MPNN) are called the Semantic Context Module (SCM).

In summary, our VS-ST-MPNN performs heterogeneous message passing and consists of two main modules: (a) the *Visual Context Module (VCM)*, that refines region node embeddings based on the local spatio-temporal interactions among actors and objects that are encoded in the visual subgraph, and (b) the *Semantic Context Module (SCM)*, that models global interactions between spatio-temporally distant regions on the shared semantic interaction space encoded in the symbolic subgraph. Different instantiations of the graph propagation rules in Eq. 2.3-2.5 give rise to different flavours of our VS-ST-MPNN framework, which we will discuss next.

2.3.2.1 Visual Context Module

Given the input visual subgraph with initial node and edge attributes, $\{\mathbf{h}_i^{(0)}\}_{i \in \mathcal{V}^v}$ and $\{\mathbf{q}_{ij}^{(0)}\}_{(i,j) \in \mathcal{E}^v}$, respectively, the Visual Context Module performs L^v iterations of heterogeneous visual message passing to obtain visual context-aware actor and node embeddings. Since we build our visual subgraph on top of noisy region proposals and with edges capturing potential interactions, we expect that it will contain many spurious edges and irrelevant contextual object nodes. To handle these type of graphs, we employ attention mechanisms to discover relevant neighboring nodes for the update of each region feature. To compute messages and update node and edge embeddings on the visual subgraph, we propose two attention-based versions of the graph propagation rules, one inspired by Graph Attention Networks (GAT) [162] and one inspired by Transformers [120]. We will now describe the instantiation of the message passing function (Eq. 2.3), node update function (Eq. 2.4) and edge update function (Eq. 2.5) for each one of them.

Visual Message Passing with Heterogeneous Graph Attention Network (HetGAT).

– *Attention mechanism:* At each visual message passing iteration $l = 1, \dots, L^v$ we first refine the strength of the connections between regions by computing *attention coefficients*, $a_{ij}^{(l)}$, that capture the relevance of node j (message sender) for the update of node i (message receiver). In contrast to GAT [162], our model learns an attention mechanism that is specialized for each type of interaction and utilizes edge features for its computation. The attention coefficients for the l-th iteration are computed as follows:

$$a_{ij}^{(l)} = \exp\left(\gamma_{ij}^{(l)}\right) / \left(\sum_{k \in \mathcal{N}^v_{r(i,j)}(i)} \exp\left(\gamma_{ik}^{(l)}\right) \right), \tag{2.8}$$

$$\gamma_{ij}^{(l)} = \rho\left(\left(\mathbf{v}_a^{r(i,j)}\right)^T \left[W_{ar}^{\nu(i)} \mathbf{h}_i^{(l-1)}; W_{as}^{\nu(j)} \mathbf{h}_j^{(l-1)}; W_{ae}^{r(i,j)} \mathbf{q}_{ij}^{(l-1)} \right] \right). \tag{2.9}$$

Here, $\nu(i)$ is the type of node i, $r(i,j)$ is the type of the edge from node j to node i, $\mathcal{N}^v_{r(i,j)}(i)$ is the set of visual nodes connected with node i via an incoming edge of type $r(i,j)$, $\mathbf{h}_i^{(l-1)}$ is the embedding of the i-th node at the previous iteration, $\mathbf{q}_{ij}^{(l-1)}$ is the embedding of the edge from j to i at the previous iteration, and ρ is a non-linearity, such as Leaky-ReLU [163]. $(\mathbf{v}_a^{r(i,j)}, W_{ar}^{\nu(i)}, W_{as}^{\nu(j)}, W_{ae}^{r(i,j)})$ are the learnable weights of our attention mechanism (although we have dropped the layer index (l) from these weights for readability, we use different attention weights at each iteration).

In other words, on the first visual message passing iteration, a node-type-specific linear transformation, parameterized by a weight matrix $W_{as}^{\nu(i)} \in \mathbb{R}^{d_V \times d_{\nu(i)}}$, is applied to every sender node, where d_V is the visual node embedding size. Similarly, current receiver node embeddings are linearly transformed with learnable weights $W_{ar}^{\nu(j)} \in \mathbb{R}^{d_V \times d_{\nu(j)}}$. Importantly, our attention mechanism takes into account the current edge embeddings after projecting them with learnable, edge-type-specific weights $W_{ae}^{r(i,j)} \in \mathbb{R}^{d_V \times d_{r(i,j)}}$. Given the projected node and edge embeddings, the attention mechanism is a single-layer feedforward neural network, parameterized by an edge-type specific weight vector $\mathbf{v}_a^{r(i,j)} \in \mathbb{R}^{d_V}$. Note that in Eq. 2.8 we compute separately normalized attention coefficients for each type of incoming edges to a receiver node. Intuitively, an actor node at time t will separately attend to actor nodes at the previous frame and object nodes at the current frame.

– *Message computation:* After computing the attention coefficient for each edge (i, j) from node j to node i, we compute a message along that edge with an edge-type-specific message computation function $\text{MSG}_{r(i,j)}^{(l)}$ that takes the form:

$$\mathbf{m}_{ij}^{(l)} = a_{ij}^{(l)} \left(W_{ms}^{\nu(j)} \mathbf{h}_j^{(l-1)} + W_{me}^{r(i,j)} \mathbf{h}_{ij}^{(l-1)} \right) \in \mathbb{R}^{d_V}, \tag{2.10}$$

where $W_{ms}^{\nu(j)}$ and $W_{ms}^{\nu(j)}$ are learnable projection matrices for the node and edge attribute, respectively. In practice, we found that using the same weights in the attention and message computation leads to better performance ($W_{ms}^{\nu(j)} = W_{as}^{\nu(j)}, W_{me}^{r(i,j)} = W_{ae}^{r(i,j)}$).

– *Node update:* Following the message computation, the node embedding is updated using an aggregation of incoming messages from different edge types and a residual connection,

$$\mathbf{h}_i^{(l)} = \sigma \left(W_u^{\nu(i)} \mathbf{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}^v(i)} \mathbf{m}_{ij}^{(l)} \right) \in \mathbb{R}^{d_V}, \tag{2.11}$$

where $\mathcal{N}^v(i)$ is the set of visual nodes that are connected with node i, $W_u^{\nu(i)}$ is a learnable projection matrix that we employ if there is a size mismatch between the current node embeddding and the message size, and $\sigma(\cdot)$ is a non-linearity, such as ReLU.

– *Edge update:* A natural way of updating the edge embedding is by similarly combining the current edge embedding with the embeddings of the connected nodes, modulated by the attention coefficient:

$$\mathbf{q}_{ij}^{(l)} = a_{ij}^{(l)} \sigma \left(W_{ae}^{r(i,j)} \mathbf{q}_{ij}^{(l-1)} + W_{as}^{\nu(j)} \mathbf{h}_j^{(l-1)} + W_{ar}^{\nu(i)} \mathbf{h}_i^{(l-1)} \right) \in \mathbb{R}^{d_V}. \tag{2.12}$$

In practice, passing the computed message through a non-linearity performs comparably:

$$\mathbf{q}_{ij}^{(l)} = \sigma \left(\mathbf{m}_{ij}^{(l)} \right) \in \mathbb{R}^{d_V}. \tag{2.13}$$

Visual Message Passing with Heterogeneous Graph Transformer (HetGTx). Motivated by the success of multi-head Transformer-based architectures, we propose an alternative heterogeneous message passing mechanism, called HetGTx. Overall, the key differences from the HetGAT message passing are the use of multi-head scaled-dot-product attention and the use of multi-layer feedforward networks for the node and edge update. We also assume for simplicity that node (edge) attributes have already been mapped to a common dimension d_V with learnable node-type-specific (edge-type-specific) embedding functions.

– *Attention mechanism:* Multi-head attention allows the model to implicitly attend to information from different representation subspaces of neighboring nodes. Additionally, our model is designed to (a) explicitly attend to information in different subgraphs which arise from different edge types, and (b) utilize edge features, both of which enhance the model's encoding capability. The proposed attention mechanism employs K edge-type-specific scaled-dot-product attention heads for each message-passing iteration l. The attention coefficients a_{ij} for each attention head $k = 1, \ldots, K$ for the l-th iteration are computed as follows:

$$a_{ij}^{(k,l)} = \exp\left(\gamma_{ij}^{(i,l)}\right) / \left(\sum_{n \in \mathcal{N}_{r(i,j)}^{v}(i)} \exp\left(\gamma_{in}^{(k,l)}\right)\right), \tag{2.14}$$

$$\gamma_{ij}^{(k,l)} = \frac{\left(Q^{r(i,j)}\mathbf{h}_i^{(l-1)}\right)^T \left(K^{r(i,j)}\mathbf{h}_j^{(l-1)}\right)}{\sqrt{d_V}} + \mathbf{w}_e^T \mathbf{q}_{ij}^{(l-1)}, \tag{2.15}$$

where $Q^{r(i,j)} \in \mathbb{R}^{d_V \times d_V}$ is the *edge-type-specific* query projection matrix, with the current receiver node embedding $\mathbf{h}_i^{(l-1)}$ serving as the query. Similarly, $K^{r(i,j)} \in \mathbb{R}^{d_V \times d_V}$ is the key projection matrix, while $\mathbf{w}_e$ is an edge projection vector. The proposed self-attention mechanism applied for each edge type closely follows the scaled dot-product self-attention mechanism used in Transformers [120], while the edge feature embedding ($\mathbf{w}_e^T \mathbf{q}_{ij}^{(l-1)}$) can be seen as a generalization of the relative position bias used in architectures like the Swin Transformer [164]. Moreover, whereas the relative position bias is a learnable embedding of the relative position between tokens, our attention coefficients can incorporate any edge feature describing the interaction between actors and objects (for example, the difference in coordinates of the object centroids and human skeleton joint locations, or visual features from the union of bounding boxes of interacting entities).

– *Message computation:* Similar to HetGAT, we compute a message along each edge from node j to node i with an edge-type-specific message computation function $\mathrm{MSG}^{(l)}_{r(i,j)}$. However, our message now consists of K concatenated messages, each one of dimension $\lfloor d_V/K \rfloor$ from each head:

$$\begin{aligned}\mathbf{m}_{ij}^{(l)} = \Big[& a_{ij}^{(1,l)} \left(V_1^{r(i,j)} \mathbf{h}_j^{(l-1)} + E_1^{r(i,j)} \mathbf{q}_{ij}^{(l-1)} \right); \ldots ; \\ & a_{ij}^{(K,l)} \left(V_K^{r(i,j)} \mathbf{h}_j^{(l-1)} + E_K^{r(i,j)} \mathbf{q}_{ij}^{(l-1)} \right) \Big] \in \mathbb{R}^{d_V},\end{aligned} \tag{2.16}$$

where $[\cdot;\cdot]$ denotes concatenation and $\{V_k^{r(i,j)}\}_{k=1}^K$, $\{E_k^{r(i,j)}\}_{k=1}^K$ are learnable head-wise projection matrices for the values (sender node embeddings) and edge features.

– *Node update:* Following the message computation, the node embedding is updated using an aggregation of incoming messages from different edge types and attention heads similar to the Transformer architecture:

$$\hat{\mathbf{h}}_i^{(l)} = \mathrm{LN}\left(\mathbf{h}_i^{(l-1)} + \sum_{\epsilon \in \mathcal{R}^v} \left(W_O^{\epsilon} \sum_{j \in \mathcal{N}_\epsilon^v(i)} \mathbf{m}_{ij}^{(l)} \right) \right), \tag{2.17}$$

$$\mathbf{h}_i^{(l)} = \mathrm{LN}\left(\hat{\mathbf{h}}_i^{(l)} + \mathrm{FFN}(\hat{\mathbf{h}}_i^{(l)}) \right), \tag{2.18}$$

where $\mathrm{LN}(\cdot)$ denotes LayerNorm [165], and FFN is a 2-layer MLP with hidden size $2 \times d_V$. An important difference from the original Transformer is that we aggregate messages from all attention heads for each edge type and project them to an edge-type-specific vector. We do so using a set of learnable matrices $\{W_O^{\epsilon} \in \mathbb{R}^{d_V \times d_V}\}_{\epsilon \in \mathcal{R}^v}$.

– *Edge update:* Following the node update computation, we update the edge embeddings as:

$$\hat{\mathbf{q}}_{ij}^{(l)} = \text{LN}\left(\mathbf{q}_{ij}^{(l-1)} + W_o^{r(i,j)}\mathbf{m}_{ij}\right), \tag{2.19}$$

$$\mathbf{q}_{ij}^{(l)} = \text{LN}\left(\hat{\mathbf{q}}_{ij}^{(l)} + \text{FFN}(\hat{\mathbf{q}}_{ij}^{(l)})\right). \tag{2.20}$$

After L^v rounds of node and edge updates, using the HetGAT message passing equations (Eq. 2.10- 2.12), or the HetGTx message passing equations (Eq. 2.16- 2.20) we obtain refined, visual context-aware node and edge embeddings: $\mathbf{h}_i^{(L^v)} \in \mathbb{R}^{d_V}$ and $\mathbf{q}_{ij}^{(L^v)} \in \mathbb{R}^{d_V}$.

2.3.2.2 Semantic Context Module

The Semantic Context Module performs message passing across the visual-symbolic and symbolic edges.

– *Integration of visual evidence with the symbolic graph:* As a first step, we update the attributes of the symbolic graph using visual evidence, i.e., the visual context-aware representations of the nodes of the visual st-graph. To achieve this, at the $(L^v + 1)$-th iteration, we pass messages from visual nodes i to symbolic nodes c across the visual-symbolic edges $(c, i) \in \mathcal{E}^{vs}$:

$$\mathbf{m}_{ci} = \omega_{ci}^{vs} W_p^{vs} \mathbf{h}_i^{(L^v)} \in \mathbb{R}^{d_S}, \tag{2.21}$$

where $W_p^{vs} \in \mathbb{R}^{d_S \times d_L}$ is a learnable projection weight matrix and ω_{ci}^{vs} is a data-dependent soft-assignment weight that represents the confidence of assigning the current embedding of visual node i to the symbolic node c:

$$\omega_{ci}^{vs} = \frac{\exp\left(\left(\mathbf{w}_c^{vs}\right)^T \mathbf{h}_i^{(L^v)}\right)}{\sum_{c' \in \mathcal{N}^{vs}(i)} \exp\left(\left(\mathbf{w}_{c'}^{vs}\right)^T \mathbf{h}_i^{(L^v)}\right)}. \tag{2.22}$$

Thus, our model learns associations between the nodes of the visual st-graph and those of the symbolic graph, without requiring access to the ground-truth semantic labels of regions. Once the messages are computed, each symbolic node is represented by concatenating the symbolic attribute (e.g., linguistic embedding) with the weighted sum of linearly transformed visual node embeddings:

$$\mathbf{h}_c^{(L^v+1)} = \left[\mathbf{h}_c^{(0)}; \sigma\left(\sum_i \mathbf{m}_{ci}\right)\right] \in \mathbb{R}^{d_s + d_S}. \tag{2.23}$$

– *Semantic graph convolutions:* We are now ready to obtain semantic interaction-aware symbolic embeddings by passing messages across the symbolic edges. To do so, we leverage the flexibility of our heterogeneous message passing framework and design message functions that are suited to the symbolic subgraph. Recall that the structure of the symbolic graph (adjacency matrix) is fixed and shared across all videos of a dataset, and the symbolic graph does not have edge attributes. Thus, instead of using an attention-based and edge-feature-aware message function as we did for passing messages on the visual subgraph (Eq. 2.10 or Eq. 2.16), we use a simpler message function (graph convolution) [3]:

$$\mathbf{m}_{cc'}^{(l)} = \frac{1}{\sqrt{deg(c)deg(c')}} W_s \mathbf{h}_{c'}^{l-1} \in \mathbb{R}^{d_S}, \tag{2.24}$$

where $deg(c)$ is the degree of the c-th node in the symbolic subgraph (with added self-loops): $deg(c) = \sum_{c'} \mathcal{A}^s_{cc'} + 1$. The embedding of each symbolic node is updated by aggregating messages from all neighboring symbolic nodes:

$$\mathbf{h}_c^{(l)} = \sigma\left(\sum_{c' \in \mathcal{N}^s(c)} \mathbf{m}_{cc'}\right), \quad l = L^v + 2, \ldots, L^v + 1 + L^s, \tag{2.25}$$

– *Update of visual st-graph:* The evolved symbolic node embeddings obtained after L^s iterations of graph convolutions on the symbolic graph can be mapped back to the visual subgraph, so that the representation of the visual nodes can be enriched by global semantic context. To achieve this we compute mapping weights (attention coefficients) from symbolic nodes to visual nodes:

$$\omega_{i,c}^{sv} = \frac{\exp\left(\left(\mathbf{v}_a^{sv}\right)^T \left[\mathbf{h}_c^{(L)}; \mathbf{h}_i^{(L^v)}\right]\right)}{\sum_{c' \in \mathcal{N}^{vs}(i)} \exp\left(\left(\mathbf{v}_a^{sv}\right)^T \left[\mathbf{h}_{c'}^{(L)}; \mathbf{h}_i^{(L^v)}\right]\right)}, \tag{2.26}$$

where $\mathbf{v}_a^{sv} \in \mathbb{R}^{d_L + D_s}$ is a learnable attention vector, $L = L^v + 1 + L^s$.

The final visual node embedding is then obtained by updating its current state with the aggregated messages from symbolic nodes:

$$\mathbf{m}_{ic} = \omega_{i,c}^{sv} W_p^{sv} \mathbf{h}_{c'}^{(L)} \tag{2.27}$$

$$\mathbf{h}_i = \sigma\left(\mathbf{h}_i^{(L^v)} + \sum_{c' \in \mathcal{V}^s} \mathbf{m}_{ic'}\right) \tag{2.28}$$

These context-aware representations $H = \{\mathbf{h}_i\}_{i=1}^{M}$ of visual nodes can be fed to recognition networks to solve downstream video understanding tasks. In the next sections, we show how our framework can be applied to various activity recognition tasks.

2.4 VS-ST-MPNN for Region-based Activity Recognition

We now return to our original goal of designing and learning region-based discriminative models $p(Y|I, R)$ for tackling structured output prediction tasks, such as the prediction of multiple human activity labels per frame [2] or the prediction of the activity performed by a person and the object affordances in each temporal segment of a video [1]. At a high level, our proposed video understanding frameworks consist of two main components: a visual encoder designed based on the VS-ST-MPNN described thus far, and a task-specific recognition network. The visual encoder learns representations for the regions, while the recognition network maps these visual representations to the desired video labeling.

2.4.1 VS-ST-MPNN for Sub-activity and Object Affordance Detection

The first application that we consider for our VS-ST-MPNN is sub-activity and object affordance detection. Consider the scenario where an assistive robot observes a human performing a task, e.g., *making cereal*. During the course of the captured video, the human interacts with several objects and performs multiple sub-activities. The robot needs to recognize the sequence of sub-activities performed by the human (e.g., *reaching*, *pouring*, etc.), as well as the object affordances, i.e., how an object is being used at each temporal segment (e.g., *reachable*, *pourable*, *stationary*, etc.).

Let us assume that we are given an input video I with T ground-truth temporal segments, i.e., we know the start and end time of each sub-activity. Suppose we are also given the human bounding boxes $\left\{\mathbf{r}^a_{t,i}\right\}_{i=1,\ldots,A}$ and object bounding boxes $\left\{\mathbf{r}^o_{t,i}\right\}_{i=1,\ldots,O}$ at each segment t, so that the set of regions is given by $R = \left\{\left\{\mathbf{r}^a_{t,i}\right\}_{i=1,\ldots,A}, \left\{\mathbf{r}^o_{t,i}\right\}_{i=1,\ldots,O}\right\}_{t=1}^T$. The goal of the sub-activity and object affordance detection task is to predict a sub-activity label from a set of discrete sub-activity labels $\mathcal{K}^a$ for each actor region at each temporal segment, and also predict an object affordance label from a set of discrete object affordance labels $\mathcal{K}^o$ for each object of each temporal segment. Let $\mathbf{y}^a_{t,i} \in \mathcal{K}^a$ be a random variable that indicates the sub-activity of the i-th actor at the t-th segment. Similarly, we denote with $\mathbf{y}^o_{t,i} \in \mathcal{K}^o$ the object affordance label of the i-th object at segment t. We model the conditional distribution of a video labels $Y = \left\{\left\{\mathbf{y}^a_{t,i}\right\}_{i=1,\ldots,A}, \left\{\mathbf{y}^o_{t,i}\right\}_{i=1,\ldots,O}\right\}_{t=1}^T$ given the input video I and regions R by assuming conditional independence between the sub-activity and object affordance labels of each region and each segment:

$$p(Y|R, I) = \prod_{t=1}^{T}\prod_{i=1}^{A} p(\mathbf{y}^a_{t,i}|R, I) \prod_{j=1}^{O} p(\mathbf{y}^o_{t,j}|R, I). \tag{2.29}$$

Although the above conditional independence assumption is very strong, we rely on the ability of our VS-STMPNN to capture long-range spatio-temporal context in each node embedding, which in turn tends to lead to smooth and coherent node-level predictions. As we will show in our experiments, although we do not model dependencies between the predicted sub-activity and affordance labels as powerful probabilistic graphical models do [166], our proposed model is able to produce more accurate predictions given the same input visual graph and features.

2.4.1.1 Sub-activity and Object Affordance Detection Network

We model the distribution of each sub-activity (object affordance) for region i and temporal segment s as a Categorical distribution that is parameterized via neural network that first encodes the input video sequence and set of regions R into a set of region embeddings H, and then applies a subactivity classifier on top of each actor embedding and an object affordance classifier on top of each object embedding. In other words, we treat the problem as visual graph node classification task.

The region embeddings H are obtained by building a Visual-Symbolic graph $G(R, I)$ (with the visual subgraph nodes corresponding to the actor and object regions we just described, spatial edges capturing interactions among humans and objects at each temporal segment, and temporal edges capturing interactions across temporal segments) and applying our VS-ST-MPNN on that graph:

$$H = MPNN(G(R, I); \theta_g), \tag{2.30}$$

where θ_g denotes the set of trainable parameters of the VS-ST-MPNN.

Then we apply a trainable subactivity classifier with weights and bias

$(W_{act}, \mathbf{b}_{act})$ on top of each actor embedding, and a trainable object affordance classifier $(W_{aff}, \mathbf{b}_{aff})$ on top of each object embedding:

$$\mathbf{p}^a_{s,i} = \text{softmax}(W^T_{act}\mathbf{h}_i + \mathbf{b}_{act}) \in \mathbb{R}^{|\mathcal{K}^a|}, \quad i \in \mathcal{J}^a_s, \tag{2.31}$$

$$\mathbf{p}^o_{s,i} = \text{softmax}(W^T_{aff}\mathbf{h}_i + \mathbf{b}_{aff}) \in \mathbb{R}^{|\mathcal{K}^o|}, \quad i \in \mathcal{J}^o_s, \tag{2.32}$$

$$\mathbf{y}^a_{s,i}|R, I \sim Cat(\mathbf{p}^a_{s,i}), \quad i \in \mathcal{J}^a_s, \tag{2.33}$$

$$\mathbf{y}^o_{s,j}|R, I \sim Cat(\mathbf{p}^o_{s,i}), \quad i \in \mathcal{J}^o_s, \tag{2.34}$$

where $\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$, $\mathcal{J}^a_s$ is the set of actor regions at segment s, and similarly $\mathcal{J}^o_s$ is the set of object regions at segment s.

2.4.1.2 Joint Training of the VS-ST-MPNN and Recognition Networks

The parameters θ_g of our graph-based representation learning model can be jointly trained with the sub-activity and object affordance classifiers. The joint training results in task-driven, discriminative, context-aware region embeddings.

Formally, given training data of N videos $\{I^{(n)}\}^N_{n=1}$ and their structured labels $\{Y^{(n)}\}^N_{n=1}$, we jointly optimize the following training objective using Stochastic Gradient Descent (SGD):

$$\min_{\theta_g, \theta_r} \sum_{n=1}^{N} \sum_{s=1}^{S} \left(\sum_{o \in \mathcal{J}^o_s} \mathcal{L}(\mathbf{y}^{(n)}_{s,o}, \mathbf{p}^o_{s,i}) + \sum_{j \in \mathcal{J}^a_s} \mathcal{L}(\mathbf{y}^{(n)}_{s,j}, \mathbf{p}^a_{s,j}) \right), \tag{2.35}$$

where $\mathcal{L}(\cdot)$ is the cross-entropy loss:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c} y_c \log \hat{y}_c, \tag{2.36}$$

with $y_c = 1$ if the ground-truth label is c and $y_c = 0$ otherwise, and $\hat{y}_c$ being the predicted probability for label c.

2.4.2 VS-ST-MPNN for Temporal Activity Detection

The goal of this task is to temporally detect all the activities that occur in an untrimmed video $I \in \mathbb{R}^{3\times H\times W\times T}$ with T frames. This involves predicting the activity, or activities, present at each frame of the video sequence. Let $y_{t,c} \in \{0, 1\}$ be a binary random variable that indicates whether the c-th activity from a set of C possible activity labels is present at frame t. We model the conditional distribution of a video labeling $Y = \left\{\{y_{1,c}\}_{c=1}^{C}, \dots, \{y_{T,c}\}_{c=1}^{C}\right\}$ given the input frames I and extracted region proposals R by assuming conditional independence of the presence of each activity at each frame:

$$p(Y \mid R, I) = \prod_{t=1}^{T} \prod_{c=1}^{C} p(y_{t,c} \mid R, I). \tag{2.37}$$

This independence assumption, although strong, is employed by most of the recent deep learning approaches for temporal segmentation, which rely on the ability of deep neural networks (such as recurrent neural networks and graph neural networks) to capture long-range temporal context in the frame representations. These representations then tend to lead to coherent and smooth predictions, without the need for modeling dependencies among predicted labels. Similar findings have been observed in the image segmentation literature, where state-of-the-art deep learning models are performing pixel-level predictions without the need for post-processing with Conditional Random Fields [167].

2.4.2.1 Activity Detection Network

We model the distribution of each activity at time t as a Bernoulli that is parameterized via a neural network that first encodes the input video sequence I and set of regions R into a sequence of d-dimensional features $\mathbf{f}_1, \ldots, \mathbf{f}_T$, and then applies binary classifiers $W_{cls} \in \mathbb{R}^{d \times C}$ (with biases $\mathbf{b}_{cls} \in \mathbb{R}^C$) to yield activity scores $\mathbf{p}_t \in \mathbb{R}^C$ at each timestep:

$$\mathbf{p}_t = \text{sigmoid}(W_{cls}^T \mathbf{f}_t + \mathbf{b}_{cls}) \in \mathbb{R}^C, \tag{2.38}$$

$$y_{t,c} \mid R, I \sim \text{Bernoulli}(p_{t,c}), \tag{2.39}$$

where sigmoid$(\cdot)$ is the element-wise application of the sigmoid function: $\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \in [0, 1]$.

We will now describe how we compute the sequence of features $\mathbf{f}_t$ by combining our VS-ST-MPNN, that models region interactions, with deep recurrent neural networks, that model long-term temporal dynamics. Given the input video I and region proposals (2D bounding boxes) R, we construct a heterogeneous graph $G(R, I) = (\mathcal{V}, \mathcal{E}, \mathcal{H}^{(0)}, \mathcal{Q}^{(0)})$ as described in Section 2.3.1. The heterogeneous, attributed graph G is then fed to the VS-ST-MPNN with trainable parameters θ_g, which yields context-aware region embeddings $H \in$

$\mathbb{R}^{M \times d_V}$, as described in Section 2.3.2:

$$H = \text{MPNN}(G(R, I); \theta_g). \tag{2.40}$$

The next step aims to generate a frame-level feature by aggregating the actor embeddings, since our goal is to predict all the activities present in the frame and we only have access to frame-level activity annotations. This frame-level feature will be fed to binary activity classifiers as explained in Eq. 2.39. To achieve this, we use average pooling to aggregate the actor representations at each frame, yielding a region-based frame feature $\hat{\mathbf{f}}_t$:

$$\hat{\mathbf{f}}_t = \frac{1}{A} \sum_{i \in \mathcal{J}_t^a} \mathbf{h}_i, \tag{2.41}$$

where $\mathcal{J}_t^a$ is the set of visual actor nodes at frame t. These region-based frame representations are optionally passed through a deep recurrent neural network, such as the Bidirectional Gated Recurrent Unit (BiGRU) [168], yielding the final context-aware frame embeddings $\mathbf{f}_t = [\overrightarrow{\mathbf{f}_t}; \overleftarrow{\mathbf{f}_t}] \in \mathbb{R}^d$:

$$\overrightarrow{\mathbf{f}_t} = \text{BiGRU}(\overrightarrow{\mathbf{f}_{t-1}}, \hat{\mathbf{f}}_t; \theta_{GRU}) \tag{2.42}$$

$$\overleftarrow{\mathbf{f}_t} = \text{BiGRU}(\overleftarrow{\mathbf{f}_{t+1}}, \hat{\mathbf{f}}_t; \theta_{GRU}), \tag{2.43}$$

where θ_{GRU} are the trainable parameters of the BiGRU network.

Note that an interesting avenue for future work would be to adopt Multiple Instance Learning (MIL) and apply binary classifiers on top of each actor's embedding to obtain activity predictions which would be aggregated to yield frame-level labels. This approach might be better suited to videos with a large number of actors. In this thesis we construct the visual graph so that it only contains a few number of actors at each frame, either because the videos in the chosen datasets contain one or two actors, or by designing methods for decomposing the video in sub-videos centered around a primary actor as will describe in Chapter 4.

2.4.2.2 Training Objective

The parameters θ_g of our graph-based representation learning network can be jointly trained with the parameters $\theta_{rec} = (\theta_{GRU}, W_{cls}, \mathbf{b}_{cls})$ of the task-specific recognition network $\phi_{rec}(\cdot)$, consisting of the BiGRU and the binary activity classifiers. This joint training results in task-driven, discriminative, context-aware region embeddings.

Formally, given training data of N_s videos $\{I^{(n)}\}_{n=1}^{N_s}$ and their structured labels $\{Y^{(n)}\}_{n=1}^{N_s}$, we jointly learn the parameters θ_g of the VS-ST-MPNN with the parameters θ_r of the recognition network by optimizing the following training objective $\mathcal{L}(\cdot)$ using Stochastic Gradient Descent (SGD):

$$\min_{\theta_g, \theta_r} \frac{1}{N_s} \sum_{n=1}^{N_s} \mathcal{L}\left(Y^{(n)}, \phi_{\text{rec}}\left(\text{MPNN}(G(R, I); \theta_g), F; \theta_r\right)\right), \tag{2.44}$$

where $\mathcal{L}$ is the binary cross-entropy loss (that can be derived by maximizing the likelihood of labeled data assuming *i.i.d.* training samples):

$$\mathcal{L}\left(Y^{(n)}, \hat{Y}^{(n)}\right) = \sum_{t=1}^{T} \sum_{c=1}^{C} y_{t,c} \log p_{t,c} + (1 - y_{t,c}) \log(1 - p_{t,c}). \tag{2.45}$$

Importantly, our framework does not require access to ground-truth semantic labels of regions or scene graphs during training.

Table 2.1: Overview of various models that we evaluated on activity recognition tasks. They stem from different choices of the Visual Message Passing algorithm used in the Visual Context Module (VCM), whether the Semantic Context Module (SCM) is used, whether graph edge attributes are used in the VCM, and whether a deep temporal model is used. All methods are based on the graph-based representation learning framework discussed in this chapter.

Model	VCM	SCM	Edge Attr.	Temporal Model
V-HetGAT	HetGAT	✗	✓	✗
V-HetGTx	HetGTx	✗	✓	✗
V-HetGAT-NodeOnly	HetGAT	✗	✗	✗
V-HetGTx-NodeOnly	HetGTx	✗	✗	✗
VS-HetGAT	HetGAT	✓	✓	✗
VS-HetGTx	HetGTx	✓	✓	✗
VS-HetGAT-BiGRU	HetGAT	✓	✓	✓

2.5 Experimental Evaluation

To demonstrate the effectiveness and generality of our method, we conduct experiments on the two aforementioned video understanding tasks that require reasoning about visual and semantic interactions between semantic entities: a) sub-activity and object affordance classification (Sec. 2.5.1), and b) temporal activity detection (Sec. 2.5.2). We trained and tested various instantiations of our Graph Neural Network for representation learning on heterogeneous spatio-temporal graphs with different design and combinations of the visual and semantic context modules. Table 2.1 gives an overview of all the compared versions of our model.

2.5.1 Sub-activity and Object Affordance Detection

We begin with the first application of our VS-ST-MPNN for the task of sub-activity and object affordance detection given ground-truth temporal segmentation. First, we describe the CAD-120 dataset, a public benchmark dataset that provides annotations for the sub-activity performed by a human subject and the affordances of the objects at each temporal segment of a video. Then, we present the metrics that we use to evaluate the performance of our region-based framework as well as the implementation details in Sections 2.5.1.2 and 2.5.1.3. After comparing different variants of our model in Section 2.5.1.4, we compare our best performing model variant with state-of-the-art sub-activity and object affordance detection methods in 2.5.1.5. In Section 2.5.1.6 we present an ablation analysis on the effect of (a) the message size, (b) the number of visual graph node attribute updates, and (c) the node update functions on the final performance. Last, we provide qualitative results in Section 2.5.1.7.

2.5.1.1 CAD-120 Dataset

The CAD-120 dataset [1] provides 120 RGB-D videos, with each video showing a daily activity comprised of a sequence of sub-activities (e.g., *moving*, *drinking*) and object affordances (e.g., *reachable*, *drinkable*). Given ground-truth temporal segments, the task is to classify each actor in each segment into one of 10 sub-activity classes and each object into one of 12 object affordance classes. Figure 2.4 depicts sample frames with annotations from a video in CAD-120. Note that local actor and object representations are not sufficient

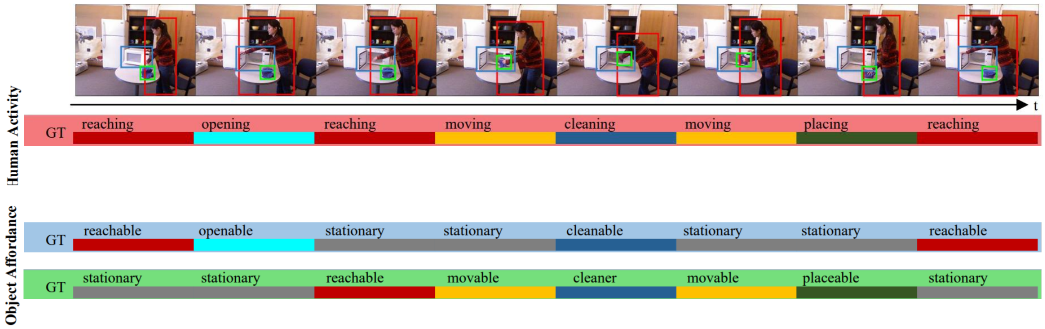


Figure 2.4: Sample annotated frames from the CAD-120 dataset [1]. This example video shows a person cleaning a microwave. The goal is to predict a sub-activity label for the actor and an affordance label for each object in each temporal segment. Figure adapted from [11].

for recognizing the sub-activities and object affordances, but rather an understanding of spatio-temporal interactions between human poses and objects over longer time periods is essential.

2.5.1.2 Evaluation Metrics

Evaluation is performed with 4-fold, leave-one-subject-out, cross-validation using F1-scores averaged over all classes as an evaluation metric. The F1-score for a class c is the harmonic mean of the precision and recall metrics:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall}, \tag{2.46}$$

where Precision is the percentage of correct predictions for class c out of the total number of times that class c was predicted, and Recall is the percentage of correct predictions for class c out of the total number of ground-truth instances of class c:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{2.47}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}. \tag{2.48}$$

2.5.1.3 Implementation Details

Construction of the Visual Subgraph. Since the CAD-120 dataset [1] provides a visual spatio-temporal graph (including hand-crafted features of actors and objects and geometric relations), it is a particularly good test-bed for comparing different GNNs. This visual graph is instantiated on the actor and objects of each temporal segment of an input sequence. Actor node features, $\phi_a(\mathbf{r}_i, I) \in \mathbb{R}^{1030}$, correspond to human skeleton joint positions, body pose and hand position features, with a cumulative binning of the feature values into 10 bins. Object node features, $\phi_o(\mathbf{r}_i, I) \in \mathbb{R}^{180}$, correspond to the location of the object and its trajectory in the temporal segment. There are 5 edge types: edges connecting objects in the same temporal segment (*obj-obj-s*), edges connecting objects with the actor within a temporal segment (*obj-act-s*), edges connecting the actor with objects within a temporal segment (*act-obj-s*), edges connecting actors between two consecutive temporal segments (*act-act-t*), and edges connecting objects between two consecutive temporal segments (*obj-obj-t*). Edge features describe the geometric relationship between nodes (regions) i and j, such as the difference in centroids and distance between them, and each edge type is associated with a feature of different dimensionality: $\phi_{oos}(\mathbf{r}_i, \mathbf{r}_j, I) \in \mathbb{R}^{200}$, $\phi_{oas}(\mathbf{r}_i, \mathbf{r}_j, I) \in \mathbb{R}^{400}$, $\phi_{aos}(\mathbf{r}_i, \mathbf{r}_j, I) \in \mathbb{R}^{400}$, $\phi_{aat}(\mathbf{r}_i, \mathbf{r}_j, I) \in$

$\mathbb{R}^{160}$, and $\phi_{oot}(\mathbf{r}_i, \mathbf{r}_j, I) \in \mathbb{R}^{40}$. For an analytic description of these features that are used to initialize the node and edge attributes of the visual subgraph refer to Table I in [1].

Construction of the Symbolic Subgraph. We construct a symbolic graph that has $|\mathcal{V}^s| = 22$ nodes corresponding to the 10 sub-activity and 12 affordance classes, with edge weights $\mathcal{A}^s$ capturing per-segment class co-occurrences in training data (Figure 2.16). The attribute of each symbolic node is obtained by using off-the-shelf, pretrained GloVe [169] word embeddings [170] to represent the semantic class of the node ($\phi_s(c) \in \mathbb{R}^{300}, c \in \mathcal{V}^s$). Visual-symbolic edges $\mathcal{E}^{vs}$ connect actor visual nodes with sub-activity symbolic nodes, and object visual nodes with affordance symbolic nodes.

Hyperparameters. In Table 2.3, we compare state-of-the-art methods with our VS-HetGAT model and our V-HetGAT model (which operates only on the visual subgraph). To choose the hyperparameters for these models, i.e., the message dimensions d_V and d_S, the number of visual node updates L^v and the number of symbolic node updates L^s, we used 5-fold cross-validation by randomly sampling 50 video sequences for training and 13 for testing. We cross-validated the following hyperparameter values: $L^v = [1, 2, 4]$, $L^s = [1, 2]$, $d_V = [64, 256]$, $d_S = [64, 256]$. This resulted in the following hyperparameters being used in our VS-HetGAT model: $L^v = 4$, $L^s = 1$, $d_V = 256$, $d_S = 256$. The Leaky-ReLU [163] non-linearity is used in the additive attention computation (Eq. 2.9) with slope 0.2, while the ReLU non-linearity is used for the node and edge embedding update (Eq. 2.11). As described in Section 2.4.1.2, we use the sum of cross-entropy losses per node to jointly train our model and the sub-activity and affordance classifiers applied at each node of the visual subgraph. We train for 100 epochs with a batch size of 5 sequences and use the Adam [171] optimizer with an initial learning rate of 0.001. Dropout [172] with a rate of 0.5 is applied to all fully connected layers.

2.5.1.4 Comparison of Models

First, we compare different variants of our VS-ST-MPNN model for sub-activity and object affordance detection given ground-truth temporal segments in Table 2.2. As we can see, the V-HetGAT model achieves a high F1-score both for predicting sub-activity labels for actor regions (90.2%) and for predicting object affordance labels for object regions (89.3%) on the validation set of CAD-120.

Comparing V-HetGAT (NodeOnly) with V-HetGAT, as well as V-HetGTx (NodeOnly) with V-HetGTx, shows that for both choices of heterogeneous message passing, the performance improves when using edge features to update the edge embeddings. For example, object affordance detection improves from 80.1% to 89.3% when edge features are used in the messages of the HetGAT VCM. This is in line with our expectation that capturing geometric relations between connected regions is a useful cue for video understanding.

Interestingly, our more expressive HetGTx heterogeneous message passing framework does not detect sub-activities and object affordances as accurately as its HetGAT counterparts in this dataset, lagging by 8%. These results suggest that multi-head attention and multi-layer node/edge updates are not well-suited for this dataset. This is likely due to the fact that CAD-120 is a

Table 2.2: Comparison of different variants of our VS-ST-MPNN model for sub-activity and affordance detection with ground-truth temporal segmentation on the CAD-120 dataset [1]. We compare variants in terms of average macro F1-score for sub-activity and affordance detection computed from 4-fold cross-validation with testing on a new human subject in each fold (higher is better).

Model	Sub-activity F1 (%)	Object Affordance F1 (%)
V-HetGAT	90.2	**89.3**
V-HetGAT-NodeOnly	84.6	80.1
V-HetGTx	84.2	81.0
V-HetGTx-NodeOnly	72.4	64.3
VS-HetGAT	**91.3**	88.6

small dataset and the more expressive Transformer-based model might overfit in it. Also, recall that the visual subgraph is based on given actor and object regions and does not contain many spurious edges for which multi-head attention is suited for.

We also observe that adding the Semantic Context Module (VS-HetGAT) does not significantly improve performance in either task. This might suggest that the local visual spatio-temporal context that is captured by the VCM is sufficient for recognizing the activities and object affordances in this dataset. From qualitative results, we indeed observed that even when using just the visual spatio-temporal subgraph, our model is able to make consistent predictions for the sub-activity labels and object affordance labels at each temporal segment. Another reason might be that the temporal duration of the visual subgraph is very small (with a maximum duration of 24 timesteps) and therefore there is no significant gain from modeling long-range global interactions between regions on the symbolic graph. Furthermore, the symbolic graph that

Table 2.3: Comparison with state-of-the-art methods on the CAD-120 dataset [1]. We report average macro F1-score for sub-activity and affordance detection (with ground-truth temporal segmentation) computed from 4-fold cross-validation with testing on a new human subject in each fold. Our results are averaged across five runs, with the standard deviation reported in parentheses (hence the slight difference with the ablation results reported in Table 2.2, which are obtained from single runs due to computational resource constraints). Best results are denoted in bold, second-best in italics.

Method	Sub-activity F1-score (%)	Affordance Detection F1-score (%)
KGS [1]	80.4	81.5
ATCRF [166]	86.4	85.7
HCRF [173]	89.2	-
S-RNN [141]	83.2	88.7
S-RNN (d+a) [141]	82.4	**91.1**
GPNN [11]	88.9	88.8
STGCN [116]	88.5	-
VS-HetGAT (ours)	**90.4** (±0.8)	*89.2* (±0.3)
V-HetGAT (ours)	*89.6* (±1.1)	88.6 (±0.6)

we used in this dataset captures label co-occurrences within each temporal segment, and as we discussed in Figure 2.16 there are only a few co-occurring labels.

2.5.1.5 Comparison with the State of the Art

Table 2.3 compares the performance of our method with prior work on sub-activity and affordance detection with ground-truth temporal segmentation. First, we observe that our method obtains state-of-the-art results for sub-activity detection, with an average performance of **90.4**% after 5 random runs, and a best of **91.3**%. It also obtains the second best result on object affordance detection (89.2%) - being only second to the S-RNN (d+a) [141]. The superior performance of S-RNN (d+a) on object affordance detection could be attributed to its training regime, since it is trained to solve the joint task of detection and anticipation (anticipating the sub-activity and object affordance labels for 3 seconds in the future). However, this multitask training setup does not lead to an improvement in sub-activity detection, and we outperform S-RNN (d+a) by 8% in that task.

Delving deeper, we would like to emphasize that all compared methods leverage the same visual spatio-temporal graph with the same node and edge features. However, they can be divided into two complementary groups: the first group ([1, 166, 173]) contains probabilistic graphical models, i.e., it includes methods that use the graph to represent the joint probability of sub-activity (and object affordance) labels in a video. In particular, the energy of a particular assignment of sub-activity and object affordance labels to the human and objects in each temporal segment of a given video is a function of the labels and the local, unrefined node/edge features. That energy consists of various potentials that capture the dependencies between the sub-activity (and object affordance) labels as defined by the visual spatio-temporal graph. In contrast, the second group of methods, including ours, uses the graph structure to refine the initial, local human and object features based on the visual context. Given the context-aware node embeddings, each node (human/object) label is predicted independently of the others.

Even without leveraging the symbolic subgraph with the Semantic Context Module, our method improves upon the second group of graph-based representation learning models, which were applied on the same attributed visual spatio-temporal graph, thus validating our novel heterogeneous message

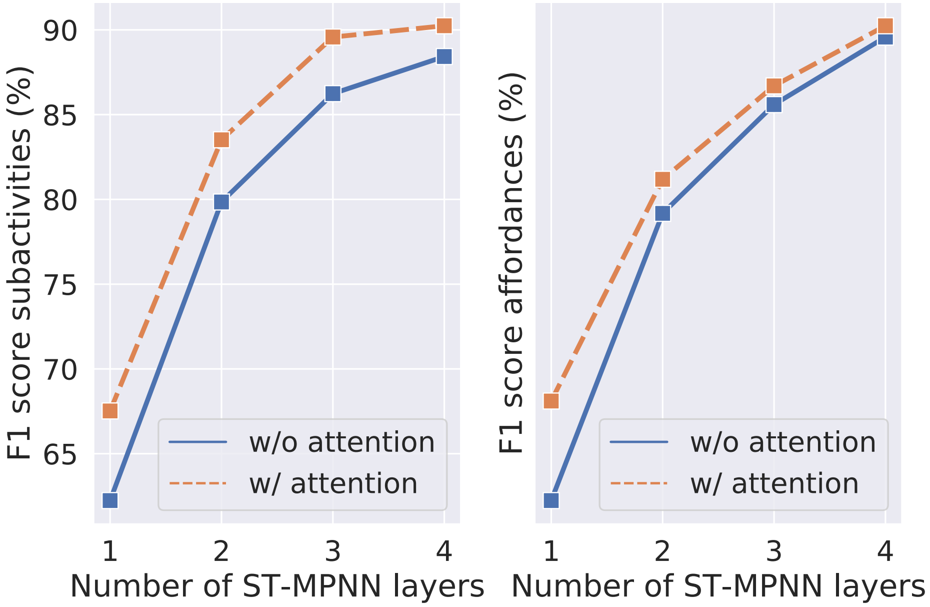


Figure 2.5: Performance of the V-HetGAT model on CAD-120 for varying number of node updates and message functions. Using an attention mechanism (*w/ attention*) outperforms using a fixed visual adjacency matrix (*w/o attention*).

and node/edge update functions. Furthermore, our VS-ST-MPNN performs competitively with powerful probabilistic graphical models, in spite of not explicitly modeling spatio-temporal label dependencies. For example, it significantly outperforms the KGS [1] probabilistic framework by more than 8.0% on both tasks. This demonstrates the quality of the learned region embeddings.

2.5.1.6 Ablation Studies

In this section, we perform an ablation analysis of our VCM (referred to as the V-HetGAT model in Table 2.1). We first study the impact of the attention mechanism and inclusion of edge attributes on the tasks of sub-activity and object affordance detection. We also study the effect of the number of visual message passing iterations and message size on the recognition performance.

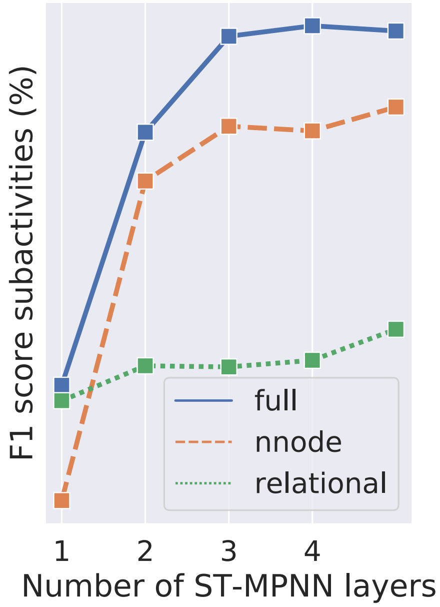


Figure 2.6: Effect of node update function of the V-HetGAT model on CAD-120 sub-activity detection performance. Updating nodes based on both neighboring node and incoming edge attributes (*full*) is superior to updating them using just the nodes (*nnode*) or edges (*relational*).

Last, we show how the input visual graph structure affects performance.

What is the effect of (a) using an attention mechanism and (b) taking edge attributes into account for node embedding updates on the sub-activity and object affordance detection performance? First, in Figure 2.5 we compare the performance of a baseline model that iteratively updates each node embedding by simply averaging the projected current states of neighboring nodes versus our attention-based model that updates each node by computing a weighted average of the neighboring node embeddings using a data-dependent, learnable attention mechanism. Although the visual subgraph used in this dataset is instantiated on top of ground-truth regions and does not contain many spurious edges, as it can be seen, using the attention mechanism improves

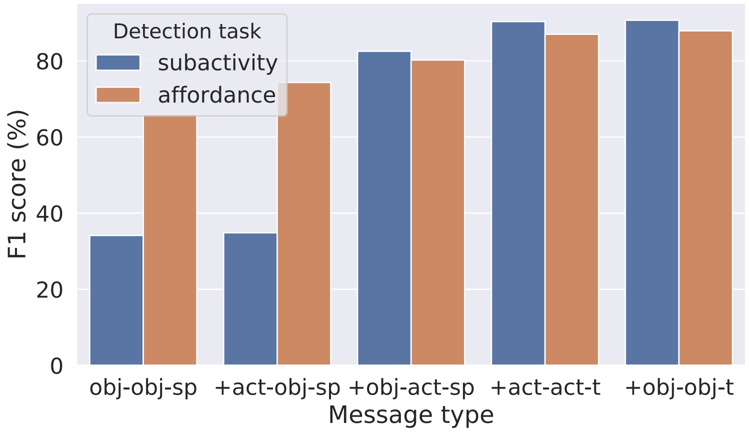


Figure 2.7: Performance on CAD-120 [1] for sub-activity and object affordance detection given ground-truth temporal segments for varying input visual subgraph structure. We incrementally add edges connecting actors and objects, starting from using only *object-to-object spatial* edges.

performance for both tasks (sub-activity and object affordance detection given ground-truth temporal segments) and for different number of node updates. As the number of layers increases the gap between performance of the two models closes, however the attention mechanism helps achieve a superior performance with a fewer number of node updates compared to the model without attention. This is in line with our expectation that using attention to refine the scalar visual edge weights helps the model adaptively identify relevant neighboring regions at each timestep. For example, given a scene with a person, a bowl and a bottle of milk, the bowl should intuitively be more relevant for learning a context-aware person representation when the person is reaching for it. Second, Figure 2.6 suggests that using the attributes of both the neighboring nodes and adjacent edges (*full* node update) is better than using only those of the neighboring nodes, validating the usefulness of edge features (*relational* node update).

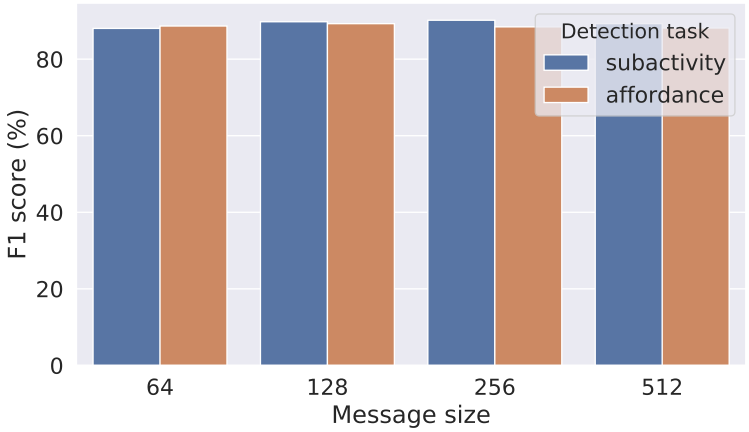


Figure 2.8: Effect of varying message size on CAD-120 [1] sub-activity and object affordance detection.

How does the number of layers of the Visual Context Module affect the sub-activity and object affordance detection performance? Another interesting observation from both Figures 2.5 and 2.6 is that increasing the number L^v of visual message passing iterations improves performance in all cases. This improvement was expected, since after L^v iterations every visual node embedding contains information about its L^v-hop spatio-temporal neighborhood. Results suggest that aggregating context from more than 3 temporal segments in the past does not significantly improve sub-activity detection performance, while object affordance detection benefits from even larger spatio-temporal neighborhoods and has not saturated even after 4 layers.

What is the impact of the visual spatio-temporal subgraph structure? In Figure 2.7 we show the contribution of each edge type on the final sub-activity and object affordance detection performance on CAD-120. Note how adding the object-to-actor-spatial (*obj-to-obj-sp*) edges leads to a significant improvement in the sub-activity detection (from 35% to 82%) due to the improved

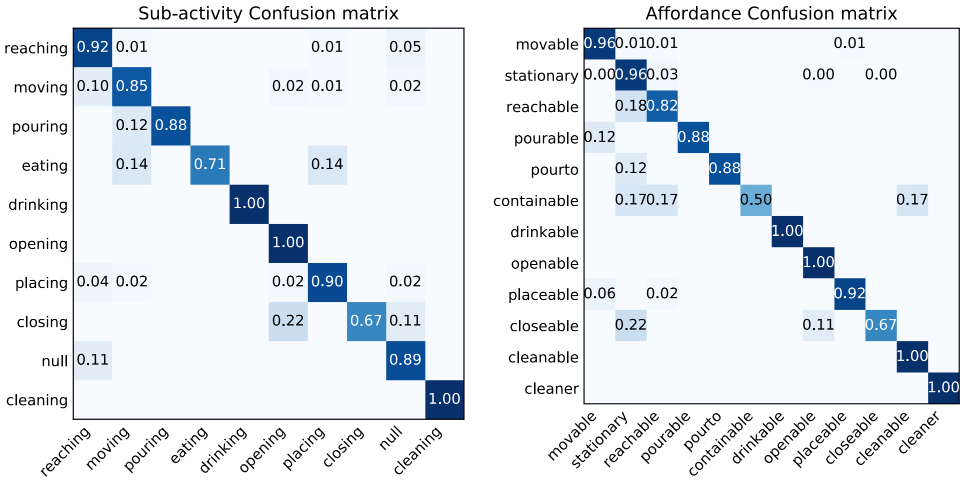
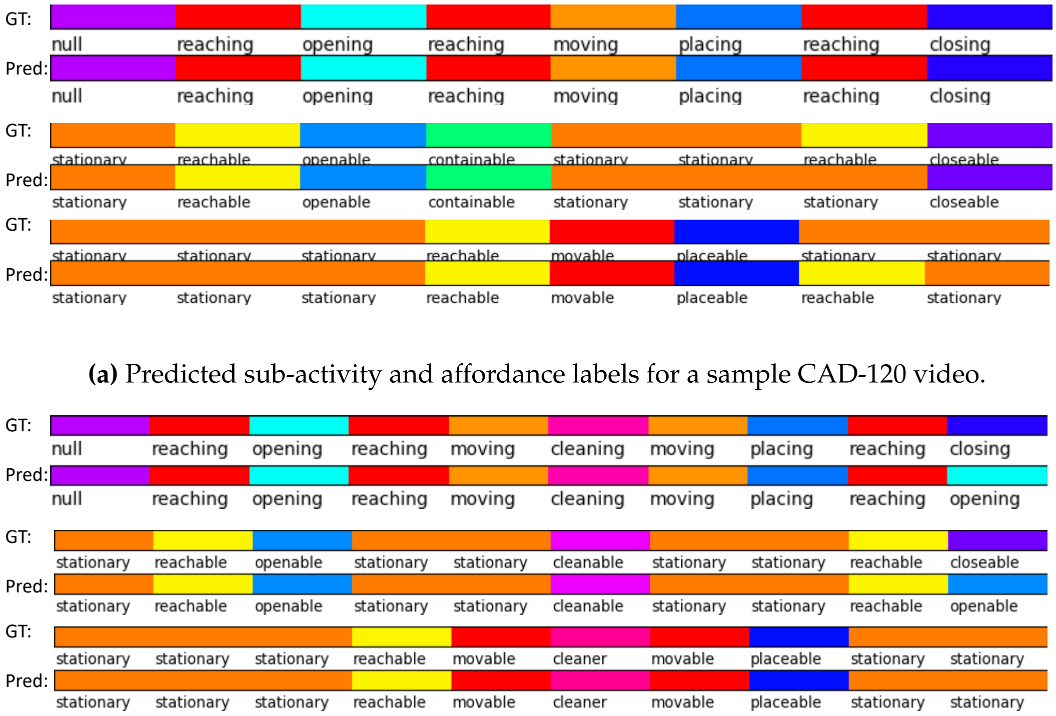


Figure 2.9: Confusion matrix for sub-activity classification (left) and object affordance classification (right) on the CAD-120 dataset [1].

actor node embeddings that have aggregated information from their local spatio-temporal neighborhood that includes contextual objects. Although the addition of object-to-actor-spatial edges does not modify the incoming messages to the object nodes, object affordance detection is also slightly improved. This can be attributed to the fact that the refinement of object and actor representations is performed jointly in each layer of our V-HetGAT model, in contrast to prior work [116] that applies separate GNNs on subgraphs emerging for different node types. Adding more edges, such as object-to-object temporal (*obj-obj-t*) does not improve performance further.

What is the effect of the visual message size? We test our VCM with varying visual message sizes $d_V = [64, 128, 256, 512]$ and plot the performance in Figure 2.8. We observe that the performance is almost consistent for different message sizes in this dataset.



(a) Predicted sub-activity and affordance labels for a sample CAD-120 video.

(b) Predicted sub-activity and affordance labels for a sample CAD-120 video.

Figure 2.10: Qualitative results for sub-activity and object affordance detection with our model (V-HetGAT) given ground-truth temporal segmentation on two sample videos from the CAD-120 dataset [1]. For each video, we show the ground-truth (*GT*) and predicted (*Pred*) labels for one actor and two objects per temporal segment.

2.5.1.7 Qualitative Results

Figure 2.9 shows the confusion matrices for predicting sub-activity and affordance labels given ground-truth temporal segments with our proposed model. As it can be seen, given the context-aware region embeddings learned by our V-HetGAT model, we are able to correctly classify most sub-activities/object affordances. For both tasks, there is a strong diagonal with a few errors such as *closing* being misclassified as *opening*, and objects that are *reachable* being misclassified as *stationary*. Figure 2.10 shows the comparison of the predicted sub-activity and object affordance labeling with the ground-truth labeling for two *cleaning objects* videos. It can be seen that our framework can correctly predict most labels. Interestingly, it seems that in the second-to-last temporal segment of the first video sample, our model confuses which one of the two objects is reachable. Another observation is that in the second example, although the model misclassifies the sub-activity *closing* as *opening*, it makes a consistent object affordance prediction (*openable*). This consistency in the labels is not explicitly enforced in our probabilistic model, but rather arises from the context-aware region embeddings learned by our VCM.

2.5.2 Temporal Activity Detection

In the previous subsection, we have described how the problem of sub-activity and object affordance detection given ground-truth temporal segmentation can be cast as a visual node classification problem and be successfully tackled with our VS-ST-MPNN. Indeed, most evaluations of graph-based representation learning frameworks in the literature focus on node classification of graph classification tasks. Though such evaluation tasks provide insights on the discriminability and descriptiveness of each individual region representation, they fail to paint a clear picture of how graph-representation learning methods can extend beyond node classification. In this section, we show how we can deploy our VS-ST-MPNN for the the task of temporal activity detection.

2.5.2.1 Charades Dataset

Charades [2] is a large dataset with 9848 RGB videos of daily activities crowd-sourced from people recording videos in their own homes. It provides temporal annotations for 157 activity classes, many of which involve human-object interactions. Fig. 2.11 depicts sample frames from 5 videos in Charades. Note that there are multiple activities captured in each video sequence, including atomic activities like *standing up* and human-object interactions like *watching TV*. Each video contains an average of 6.8 activity instances, many of which are co-occurring.

2.5.2.2 Evaluation Metric

The goal in this dataset is to predict accurate activity labels for every frame of an input video, where multiple activities can be present at each frame. Following [2], activity detection performance is measured in terms of per-frame mean Average Precision (mAP), evaluating activity predictions for 25 equidistant frames in each one of the 1.8k validation videos.

This metric treats the per-frame activity labeling task as an information retrieval problem and summarizes the precision-recall curve. For each activity class, we rank all the aforementioned frames based on their confidence scores for that class, ordered from the highest to lowest, and compute Average Precision (AP). Given the descending-score-rank of N frames for the c-th activity class, the Interpolated Average Precision is computed using the official

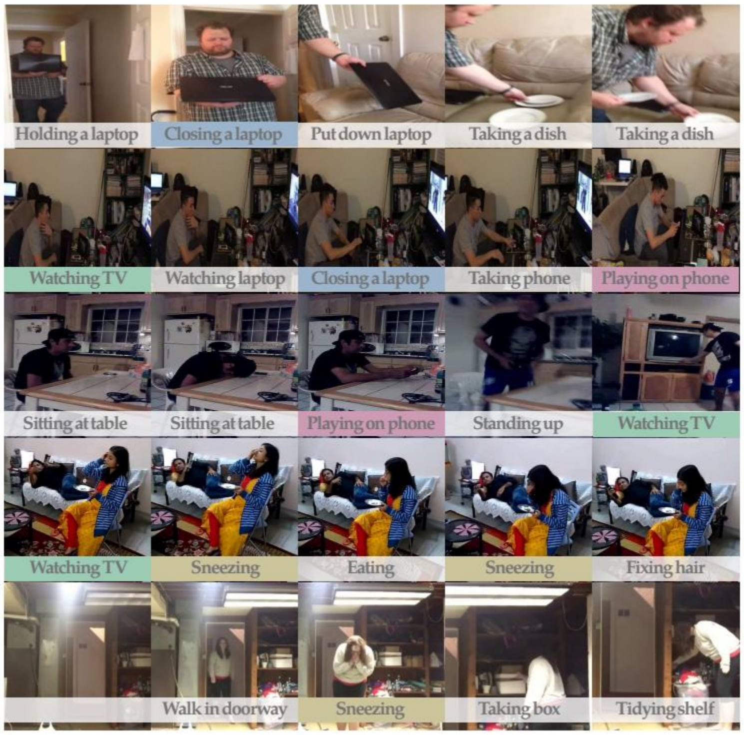


Figure 2.11: Sample frames from five Charades videos showing examples of various activity labels. Note that multiple activities can be co-occurring at the same frame, e.g., *Holding a laptop* and *Walk in doorway* in the top left frame. Figure reproduced from [2].

evaluation script (`Charades_v1_localize`) as follows [174]:

$$\mathrm{AP}(c) = \frac{\sum_{k=1}^{N}(\mathrm{Pr(k)} \times \mathrm{rel(k)})}{\sum_{k=1}^{N} \mathrm{rel}(k)}, \tag{2.49}$$

where N is the total number of frames, $Pr(k)$ is the precision for the top-k retrieved frames, and $\mathrm{rel}(k)$ is an indicator function that is equal to 1 if the k-th ranked frame is a true positive and zero otherwise. The denominator is equal to the total number of true positive frames. Ater computing the average precision for each activity class, the results are averaged over all classes to obtain the mean AP (mAP). We note that this framewise metric does not

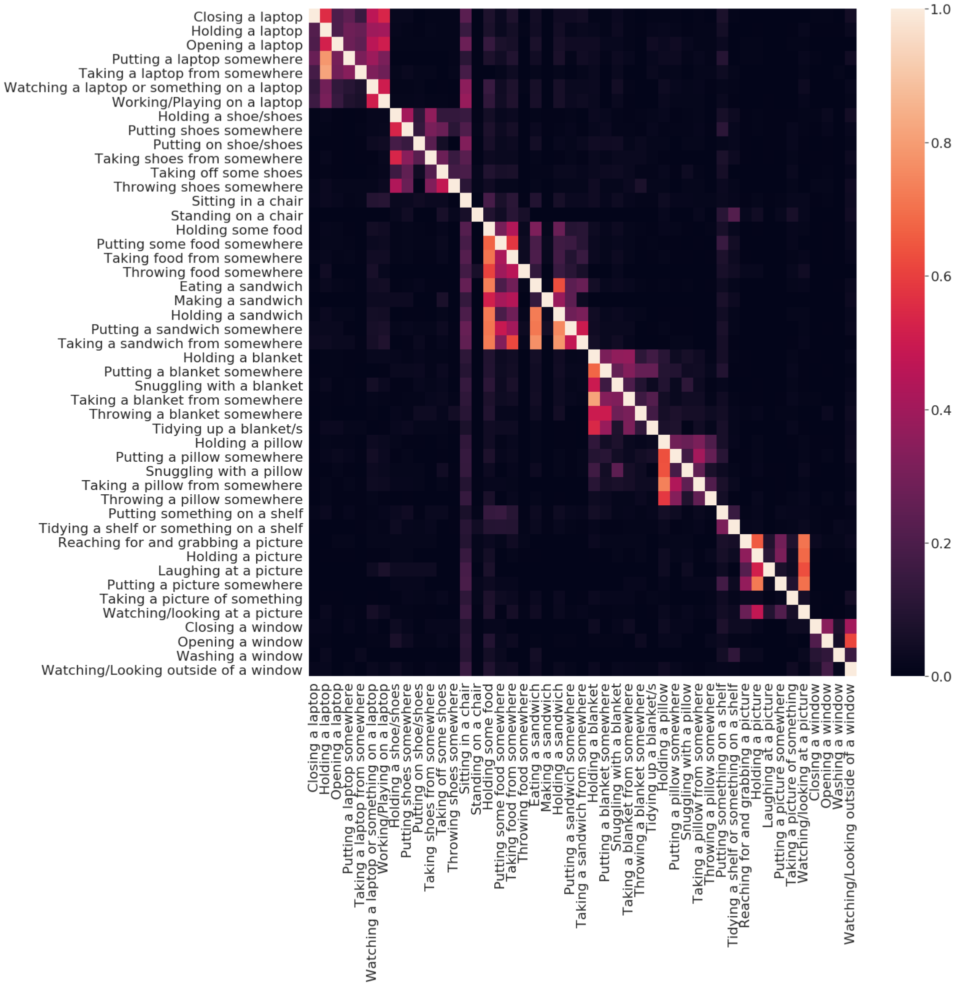


Figure 2.12: Illustration of per-frame co-occurrences of a subset of action classes from the training annotations of the Charades dataset.

explicitly account for the start or end frame of each detected activity instance. Nonetheless, we adopt it for a fair comparison with prior work in this dataset, and discuss alternative metrics that take into account the temporal overlap of activity detections with ground-truth activity instances in Chapter 4.

2.5.2.3 Implementation Details

Construction of the Visual Subgraph. The nodes of the visual sub-graph correspond to object bounding boxes returned by an object detector, specifically the Faster-RCNN [175] object detector trained on the MS-COCO [176] image dataset. We detect objects from the 80 categories of the MS-COCO dataset, retaining all detections with a confidence score above a small threshold of 0.15. Bounding boxes are enlarged by a relative margin of 30% at each side. Regions belonging to the *Person* MS-COCO object class are assigned the *Actor* node type, while the rest of non-person, non-background object detections are assigned the *Object* node type. When there are more than $A = 2$ actors or $O = 10$ objects detected in a video frame, we select the top-2 actor detections and top-10 object detections per frame based on their prediction confidence scores. Note that we do not use the predicted classes of the detected objects in our model, and treat detections as either actors or class-agnostic object region proposals.

Rather than using the object detector features for describing the actor and object regions (visual nodes), we exploit the rich spatio-temporal feature map of an I3D 3D convolutional network [44].

These features capture appearance and motion cues and are obtained by processing each 64-frame temporal chunk of the input video with the I3D RGB model in a sliding window fashion. In particular, we use the `Mixed_4f` feature map of the I3D, which has a spatial output stride of 16 pixels, a temporal output stride of 4 frames and 832 channels. We chose this feature map, since it retains spatial information, but is also located closer to the activity classification layer of the finetuned I3D model. After temporally downsampling each spatio-temporal feature map to obtain an effective temporal downsampling by a factor of 16 frames, we apply RoIAlign [175] to pool features from each visual node region at each downsampled frame. This leads to a feature map of $832 \times 7 \times 7$ per region per frame. To obtain a single feature vector for each actor and object node, we perform max-pooling over space:

$$\mathbf{h}_i^{(0)} = \phi_o(\mathbf{r}_i, I) = \mathrm{MaxPool}\left(\mathrm{RoIAlign}\left(F_t, \mathbf{r}_i\right)\right), \; i \in \mathcal{J}_t^o \tag{2.50}$$

$$\mathbf{h}_i^{(0)} = \phi_a(\mathbf{r}_i, I) = \mathrm{MaxPool}\left(\mathrm{RoIAlign}\left(F_t, \mathbf{r}_i\right)\right), \; i \in \mathcal{J}_t^a, \tag{2.51}$$

where $\mathbf{r}_i$ denotes the bounding box of the i-th region proposal, $\mathcal{J}_t^o$ ($\mathcal{J}_t^a$) denotes the set of indices of object (actor) regions located at frame t, and F_t is the I3D feature map corresponding to the t-th frame. The maximum temporal span of visual spatio-temporal subgraphs on Charades is 109 time-steps after temporal downsampling ($\sim$ 1 minute), which is significantly longer than the duration of video clips processed by holistic deep networks (e.g., $\sim$ 2 seconds). The *object-to-actor spatial* edges connect objects to actors in the same frame t, and similarly *actor-to-object spatial* edges connect actors to objects in the same frame. The *actor-to-actor temporal* edges are constrained to connect an actor at timestep t with other actors at the previous timestep $t-1$. Edge attributes ($\mathbf{q}_{ij}^{(0)}$) are initialized with the relative spatial location and size of the connected regions.

Construction of the Symbolic Subgraph. Our symbolic graph is chosen to have nodes corresponding to the 157 activity classes of the dataset and edge weights corresponding to per-frame label co-occurrences in training data, with a temporal downsampling of one every 100 frames. Figure 2.12 shows the symbolic adjacency matrix $\mathcal{A}^s$ for a subset of action classes of the Charades dataset. Visual-symbolic edges $\mathcal{E}^{vs}$ connect only actor nodes with symbolic nodes. Obtaining a linguistic attribute for each symbolic node is not trivial, since activity names often contain multiple words. To circumvent that, each activity class is separated into a verb and an object, and the average of their GloVe [169] word embeddings is used as the initial symbolic node attribute ($\phi_s(c) \in \mathbb{R}^{300}, c \in \mathcal{V}^s$).

Region-based Model. As described in Section 2.4.2.1, after applying our Graph Neural Network on the hybrid input visual-symbolic graph, we average the context-aware actor embeddings at each frame, input them to a two-layer bidirectional Gated Recurrent Unit (BiGRU), and finally feed the resulting hidden states to binary action classifiers for per-frame multi-label action classification.

Holistic Model. One of our experiments on this dataset compare the performance of our region-based video understanding framework with a holistic video understanding model. The holistic model replaces our region-based representation of each frame with a holistic frame feature that is extracted from the `Mixed_5c` layer of the I3D RGB model. Then, we mimic our region-based architecture, by feeding the sequence of holistic frame features to a two-layer BiGRU and apply binary action classifiers on the hidden states of the BiGRU to predict activity labels at each timestep.

Inspired by two-stream approaches [177, 45] that combine the activity predictions of two separately-trained models, we also combine the predictions of the holistic and region-based models via late fusion. In particular, given the trained region-based model and holistic model, we average their scores (logits) per activity and pass them through a sigmoid to obtain the final prediction per activity class.

Hyperparameters. In Table 2.6 we compare state-of-the-art methods with our framework for activity detection that is combining our region-based using the VS-HetGAT variant of our VS-ST-MPNN at its core.

The hyperparameters for this fused model were chosen via cross-validation and are: $L^v = 3$, $d_V = 512$, $L^s = 1$, $d_S = 256$. We jointly train the VS-ST-STMPNN and biGRU for 40 epochs with a binary cross-entropy loss applied per frame, using a batch size of 16 sequences. Note that the I3D model that is used to extract both local region features and holistic frame features is separately fine-tuned [178] on Charades for the task of temporal activity detection and then its weights are frozen when training the rest of our framework weights. We also apply Dropout with a rate of 0.5 on all fully connected layers and use the Adam optimizer, with an initial learning rate of $1e^{-4}$.

2.5.2.4 Comparison of Models

We present the activity detection performance of various variants of our VS-ST-MPNN region embedding model in Table 2.4 and compare it with two baselines. The first baseline predicts activities for an actor based on the local actor feature (initial node attribute) after passing it through a MultiLayer Perceptron (MLP) with hidden size 512 (same as our region embedding size). Predicting activities based on the local region features yields a low baseline

Table 2.4: Comparison of region embedding models on Charades [2]. We compare different variants of our region-based activity detection framework for different instantiations of the visual message passing, as well as baselines that employ local region features (*MLP*) or Graph Convolutional Networks [3] (*GCN*). Performance is evaluated with the frame-level mAP metric. *Visual*: visual context, *Semantic*: semantic context, *Long Term*: long-term temporal context.

Method	Visual	Semantic	Long Term	mAP (%)
MLP	✗	✗	✗	10.7
GCN	✓	✗	✗	12.2
V-HetGAT	✓	✗	✗	13.7
V-HetGAT (NodeOnly)	✓	✗	✗	13.2
V-HetGTx	✓	✗	✗	13.6
VS-HetGTx	✓	✓	✗	14.5
VS-HetGAT	✓	✓	✗	15.3
VS-HetGAT-BiGRU	✓	✓	✓	**18.6**

performance of 10.7% frame-level mAP. The second baseline applies a GCN (with the same number of layers and region embedding size as our VS-ST-MPNN) on the visual subgraph, which computes region embeddings by taking into account the contextual regions based on the input graph, but treats all nodes and edges in the same way and does not make use of edge features. Leveraging contextual region embeddings, even when they are obtained with this vanilla GNN, boosts performance to 12.2%, as expected.

Refining the local actor features by using our VCM (with either the HetGAT or the HetGTX heterogeneous message passing formulation) improves performance by 3% over the local features and $\sim$ 1.5% over the region embeddings obtained with the GCN. By furher comparing the variant of our model that uses edge-type-specific attention mechanisms but only node embedddings (V-HetGAT (NodeOnly)) with the baseline GCN, we see that just this property of our architecture improves performance from 12.2% to 13.2%.

As in our experiments on the CAD-120 dataset, the performance of the VCM improves when edge features are used for the attention computation and node updates. However, the improvement in this dataset (13.2% $\rightarrow$ 13.7% is not as pronounced as in CAD-120. This is possibly due to the initial edge attributes that are used in each setup. For example, the *actor-to-actor-temporal* edge attributes in CAD-120 are initialized with richer information capturing the distance between each corresponding human skeleton joint for 8 joints, while on Charades we initialize the same edge attribute just with the relative position and size between the human bounding boxes. In contrast to our experiments on CAD-120, where the V-HetGTx model is inferior to the V-HetGAT model, here we observe that they lead to almost the same activity detection performance. One reason that we might not observe an improved performance with the more expressive V-HetGTx model that employs multi-head scaled-dot product attention is that the structure of our visual graph is sparse with a few edges. For example, each actor node is connected with at most 2 actor nodes at the previous frame, and at most 10 object regions at the current frame. Due to the simplicity and competitive performance of the HetGAT heterogeneous message passing architecture, we will use this in most of the experiments that follow.

Continuing our comparison of model variants, we observe that graph-based representation learning on the hybrid visual-symbolic graph (VS-HetGAT)

Table 2.5: Comparison between a region-based model and a holistc model for video understanding on Charades. Performance is evaluated with the frame-level mAP metric (higher is better).

Method	Region-based	Holistic	mAP (%)
VS-HetGAT-BiGRU	✓	✗	18.6
I3D-BiGRU	✗	✓	21.5
Late Fusion (I3D+VS-HetGAT+BiGRU)	✓	✓	23.4

achieves a performance of 15.3% and, thus, yields a significant absolute improvement of **5% over the baseline**. Additionally, modeling long-term temporal dynamics with a BiGRU further improves performance, indicating that (a) the local visual spatio-temporal interactions captured by our VCM, (b) the global long-range semantic region interactions captured by our SCM, and (c) the long-term temporal dynamics of region-based frame representation captured by the BiGRU are valuable contextual cues and are complementary to each other.

In Table 2.5, we compare our region-based activity detection framework (VS-HetGAT-BiGRU), with the holistic activity detection, that uses later stage, I3D clip-level features to represent each frame. As can be seen, these frame-level features are powerful and outperform the region-based representation (18.6% vs 21.5). First, the holistic frame features are extracted from a later block of the I3D network (`Mixed_5c`), and are thus more discriminative than the initial region features (pooled from `Mixed_4f`) used in our region-based framework. Furthermore, the holistic frame representation has access to global scene cues that might be helpful for recognizing an activity, such as the room the activity takes place (e.g., kitchen). Interestingly, combining the predictions of the two models via late-fusion further improves performance to 23.4%, indicating that the representations learned by our model are complementary to holistic scene cues. This is the model that we will compare with state-of-the-art methods in the next section.

2.5.2.5 Comparison with the State of the Art

As shown in Table 2.6, our framework outperforms all other methods on temporal activity detection, with a mAP of **23**.**7**% (averaged across 3 random runs) by using only raw RGB frames. Note that all the compared methods on this Table (except for the STGCN) are holistic activity detection frameworks, that do not exploit region proposals. As we explained before, our region-based framework is by design complementary to these approaches. Importantly, our model yields a relative improvement of 24% over the alternative graph-based approach [116], which uses both RGB and optical flow inputs, as well as additional actor embeddings trained at the ImSitu dataset [179].

2.5.2.6 Ablation Studies

In this section, we perform an ablation analysis of our basic model (referred to as the VS-HetGAT model in Table 2.1). We study the effect of (a) the number of visual message passing iterations (L^v), and (b) the visual message sizes on the final performance. We also show how the structure of the input visual and symbolic subgraphs affects activity detection performance. Last, we discuss alternative implementations of our visual message passing and edge update functions.

Table 2.6: Temporal activity detection results on Charades [2]. Performance is measured via frame-level mAP. R: RGB, F: optical flow,VGG: using two-stream frame features extracted with the VGG 2D CNN, I3D: using two-stream frame features extracted with the I3D 3D CNN.

Method	Feat	Input	mAP (%)
Predictive-corrective [180]	VGG	R	8.9
Two-stream [181]	VGG	R+F	8.94
Two-stream + LSTM [181]	VGG	R+F	9.6
R-C3D [69]	VGG	R+F	12.7
ATF [181]	VGG	R+F	12.8
RGB I3D [178]	I3D	R	15.63
I3D [178]	I3D	R+F	17.22
I3D + LSTM [178]	I3D	R+F	18.12
RGB I3D + super-events [178]	I3D	R	18.64
I3D + super-events [178]	I3D	R+F	19.41
STGCN [116]	I3D	R+F	19.09
I3D + biGRU	I3D	R	21.7
I3D + 3TGMs + super-events [64]	I3D	R+F	22.3
I3D + biGRU + VS-ST-MPNN (Ours)	I3D	**R**	**23.7**(±0.2)

Table 2.7: Ablation study for number of visual message passing iterations on Charades validation set.

Number of iterations	mAP (%)
0	10.7
1	12.5
2	13.5
3	13.8

Table 2.8: Ablation study for visual message size on Charades validation set.

Message Size	mAP (%)
64	9.4
128	11.6
256	13.6
512	13.7

How does the number of layers and the message size of the Visual Context Module affect the activity detection performance? As expected, increasing the number of layers, i.e., visual node updates, improves performance due to the incorporation of more context. However, it also increases the number of learnable parameters which might lead to overfitting. For example, incorporating context from the 1-hop neighborhood of each visual node improves performance from 10.7% to 12.5%. Adding information from the 2-hop neighborhood further improves performance to 13.5%, with smaller gains when more layers are added. To understand the effect of the visual message size on the activity detection performance, we train models with message size d_V set to 64, 128, 256 and 512. The results shown in Table 2.8 suggest that as the message size increases, the activity detection performance improves, possibly

Table 2.9: Activity detection results on Charades for varying *actor-to-actor-temporal* connections.

Stride (τ)	Half Win. (w_l)	Past	Future	mAP (%)
1	1	✓	-	13.7
1	1	✓	✓	13.8
2	1	✓	✓	14.2
1	2	✓	✓	16.0
2	2	✓	✓	**16.1**
3	2	✓	✓	15.8
3	3	✓	✓	16.0
4	2	✓	✓	15.1
4	3	✓	✓	15.2

because larger message sizes can allow for the exchange of richer information across the visual nodes. However the improvement starts to saturate at $d_V = 256$. Recall that in the much smaller and simpler CAD-120 dataset the performance was almost constant for different message sizes (Figure 2.8).

What is the impact of the visual subgraph structure? We trained and tested our V-HetGAT model with various design choices of the *actor-to-actor-temporal edges* to study the effect that the temporal context has on the activity detection results. In particular, we connect an actor at timestep t with other actors at timesteps $t \pm s \cdot \tau, \tau = 1, \ldots, w_l$, where s is the stride of the temporal edges and w_l is the size of the temporal window in the past (and future). The results presented in Table 2.9 show that the design of the input visual subgraph significantly affects performance, and in particular that incorporating longer temporal context, including connection with past and future timesteps, facilitates activity detection. As we can see, having a larger temporal stride (e.g., 2 frames instead of 1) slightly improves performance even when using the

Table 2.10: Effect of the symbolic graph design on the activity detection performance, measured via frame-level mAP on the Charades validation set. We study varying number of symbolic nodes, varying initialization of symbolic node attributes (Init.), and varying adjacency matrices. *Coocc*: adjacency matrix of activity co-occurences, *Dense*: dense adjacency matrix, *Ling. Sim*: adjacency matrix of linguistic similarities, *Glove*: nodes initialized with GloVE word embeddings, *random*: nodes initialized with random attributes (300-dimensional vectors drawn from a $\mathcal{N}(0, I)$ normal distribution).

SCM	Nb. nodes	Init.	Adj.	mAP (%)
-	-	-	-	13.7
✓	157	Glove	coocc	**15.1**
✓	157	Glove	dense	14.9
✓	157	Glove	ling. sim.	**15.1**
✓	157	random	coocc	**15.1**
✓	157	random	dense	14.8
✓	64	random	dense	14.8
✓	256	random	dense	14.9

same number of temporal connections (13.8 → 14.2). This could be attributed to the fact that adjacent frames contain redundant information, so employing a temporal stride greater than 1, might result in actor embeddings with more meaningful temporal context. However, further increasing the temporal stride from 3 to 4 frames degrades performance, possibly because it includes noisy information from temporally distant frames. Another observation we make is that using too few temporal connections (e.g., one temporal connection to the previous frames) does not allow the graph neural network to capture sufficient temporal context, such as the change in the appearance of an actor (performance of 13.7%), while using too many temporal connections increases the running time and offers diminishing gains.

What is the impact of the symbolic subgraph structure? When we defined the symbolic subgraph in Section 2.3.1, we had mentioned that our framework supports multiple types of symbolic graphs. For all the experiments reported so far on Charades, we had used symbolic graphs whose nodes correspond to activity classes (157 symbolic nodes), scalar edge weights capturing co-occurrences (coocc) between activities and symbolic node attributes are initialized with word embeddings (Glove). In this ablation study, we experiment with alternative symbolic graphs, including graphs with different scalar edge weights (e.g., determined based on linguistic similarity of activity classes) and simpler symbolic graphs with latent symbolic concepts. In particular, we experiment varying number of symbolic nodes, varying symbolic adjacency matrices and varying symbolic node attribute initializations. Results summarized in Table 2.10 suggest that our Semantic Context Module can model semantic interaction in a latent semantic space and is robust to the initialization of node attributes and the choice of adjacency matrix. For example, even when using a fully-connected symbolic graph with randomly initialized symbolic attributes (300-dimensional node embeddings drawn from a $\mathcal{N}(0, I)$ normal distribution) leads to an improvement of 1% over just using the VCM (13.7 → 14.8). The improvement due to passing messages in the semantic space can be largely attributed to the modeling of interactions between regions that might be spatio-temporally distant, since embeddings from all actor nodes are transferred to the common semantic space. However, results seem to suggest that using a structured graph, i.e. using an symbolic adjacency matrix $\mathcal{A}^s$ that captures co-occurrence frequencies or linguistic similarities, slightly improves activity detection performance.

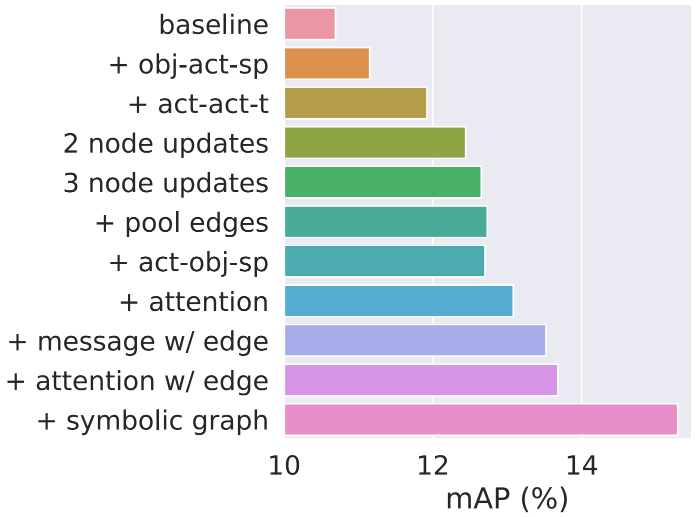


Figure 2.13: Performance ablation on Charades when incrementally adding components of our full model (VS-HetGAT), starting with early stage RGB I3D features pooled from actor regions.

What is the contribution of each component of the VS-HetGAT model to the final performance? In Figure 2.13 we study in more detail the contribution of each component of our model to the final performance, in order to validate their necessity. We start with a baseline model that classifies actions per frame based on the local actor features (actor node attributes). Adding a single round of *obj-act-sp* and *act-act-t* visual messages yields a first significant improvement in the performance (more than 1%). Frame-level mAP keeps improving as we perform more rounds of visual node and edge updates. Adding an edge-type specific attention mechanism for adapting the graph connectivity also benefits our model. Importantly, using the edge features in messages and the attention computation leads to further improvements. Our ablation ends by adding the Semantic Context Module, which boosts performance by 2%.

2.5.2.7 Qualitative Results

Performance improvement per activity class. To gain a better understanding of the benefits of representation learning on the input visual-symbolic, we highlight in Fig. 2.17 the activity classes with the highest positive and negative difference in performance when adding different types of edges (messages to nodes) For example, by harnessing visual human-object interaction cues via *object-to-actor spatial* edges, our model is able to better recognize actions such as *Watching television*. Adding *visual-symbolic* and *symbolic* edges and applying the Semantic Context Module seems to particularly help with rare classes, such as *Holding a vacuum*, which has only 213 training examples (3% of available annotated segments), and classes with strong co-occurrences.

As can be seen in the t-SNE visualization (Figure 2.15), although the visual context-aware actor embeddings are already capturing meaningful label relationships (e.g., *open* and *hold book*), the integration of long-range semantic interactions via the symbolic graph results in more tightly clustered embeddings and well-defined groups, facilitating activity detection.

In Figure 2.18 we **visualize the attention** computed along the *object-to-actor spatial* edges, by showing the two object detections that have the highest attention coefficients. As it can be seen, attention focuses on regions that contain relevant context, such as the television, chairs, tables, pots etc. In the second row, we can also see how attention shifts from the kitchen stove to the table, as the person moves. However, not all attended regions are relevant to the action performed by the actor. Furthermore, our model has the tendency to attend to large regions, since they provide more context, or might miss

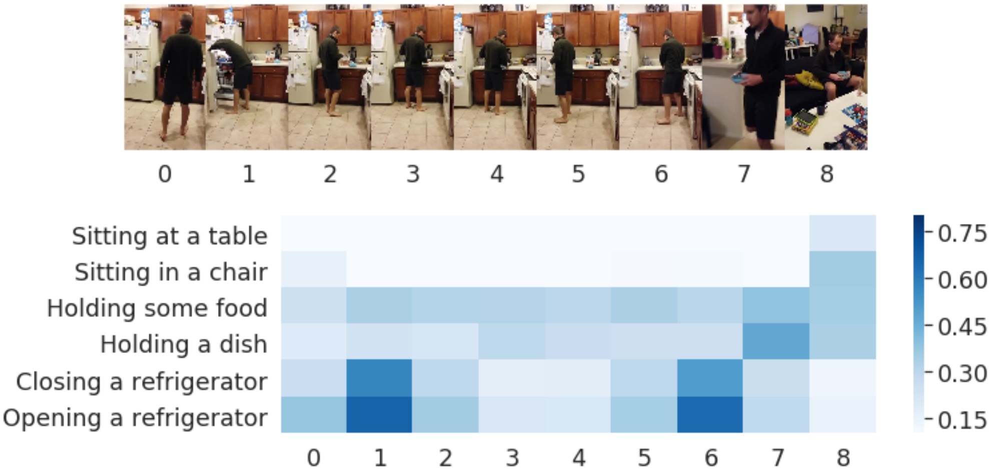


Figure 2.14: Qualitative results on Charades. Action predictions of our V-HetGAT model for 9 equidistant frames of a sample Charades video.

relevant small regions, such as the closet, as shown in the final row.

Furthermore, in Figure 2.19 we provide some **sample action predictions** (scores) for 9 frames of 3 videos from the Charades dataset. These predictions are obtained with our V-HetGAT model, without leveraging global frame features or long-range temporal dynamics. The proposed model is able to detect fine-grained actions that involve human-object interaction, such as *Drinking from a cup*, *Opening a door*, *Looking outside*, *Walking through a doorway* etc.

Model complexity. Since our visual st-graph is designed to capture only local spatio-temporal interactions, we can compute messages in parallel and process the entire Charades validation set (around 2K videos at 1.5FPS) in 2 minutes on a single Titan XP GPU, given initial features pooled from actor/object regions.

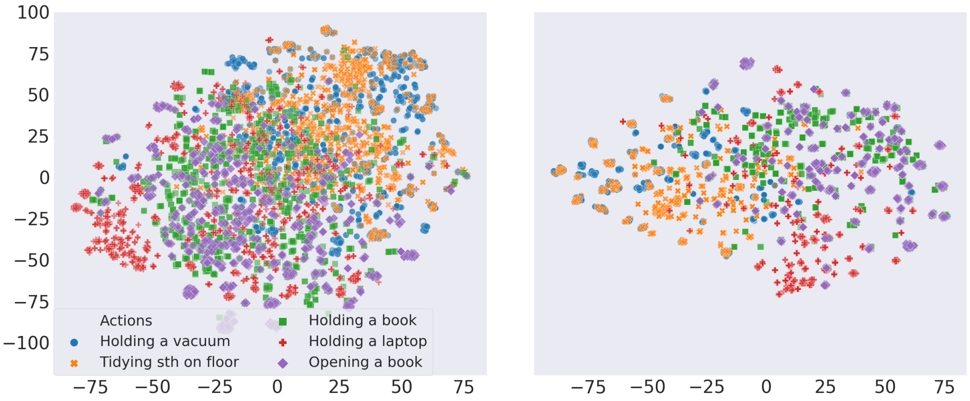


Figure 2.15: Qualitative evaluation of the Semantic Context Module (SCM). t-SNE visualization of actor node embeddings from Charades validation set obtained before and after adding the SCM. We show 1121 random samples per class for 5 selected action classes. (*Best viewed zoomed in and in color*.)

2.6 Conclusion

In this chapter, we have proposed novel heterogeneous Message Passing Neural Networks, composed of a Visual Context Module and a Semantic Context Module, for representation learning on heterogeneous graphs, which encode visual and semantic interactions among actor and object regions in a video. The key idea was learn different message functions for different edge types, and to take into account edge attributes, such as relative geometric relations between regions, in order to compute context-aware region embeddings. Experimental evaluation has shown that by jointly learning these region embeddings with activity recognition networks, e.g., classifiers, our framework outperformed baselines using local region features or contextual embeddings obtained by Graph Convolutional Neural Networks applied on a homogeneous visual graph. Our proposed VS-ST-MPNN model improves upon prior Graph Neural Networks in terms of sub-activity detection performance on the CAD-120 dataset, setting a new state of the art. Also, combining region-based activity predictions with predictions based on holistic, clip-level RGB features led to state-of-the-art temporal activity detection performance on the Charades dataset, significantly outperforming two-stream, holistic approaches that utilize both RGB and Optical Flow inputs. Our ablation studies have suggested that, given enough training data, our framework can perform equally well for different choices of heterogeneous message and node update functions. Moreover, qualitative analysis has shown that the learned model automatically attends to relevant contextual objects when aggregating relevant visual context for activity prediction at each timestep, and that richer interactions (e.g., more types of interactions or long-term interactions) encoded in the input graph lead to better embeddings learned by our model.

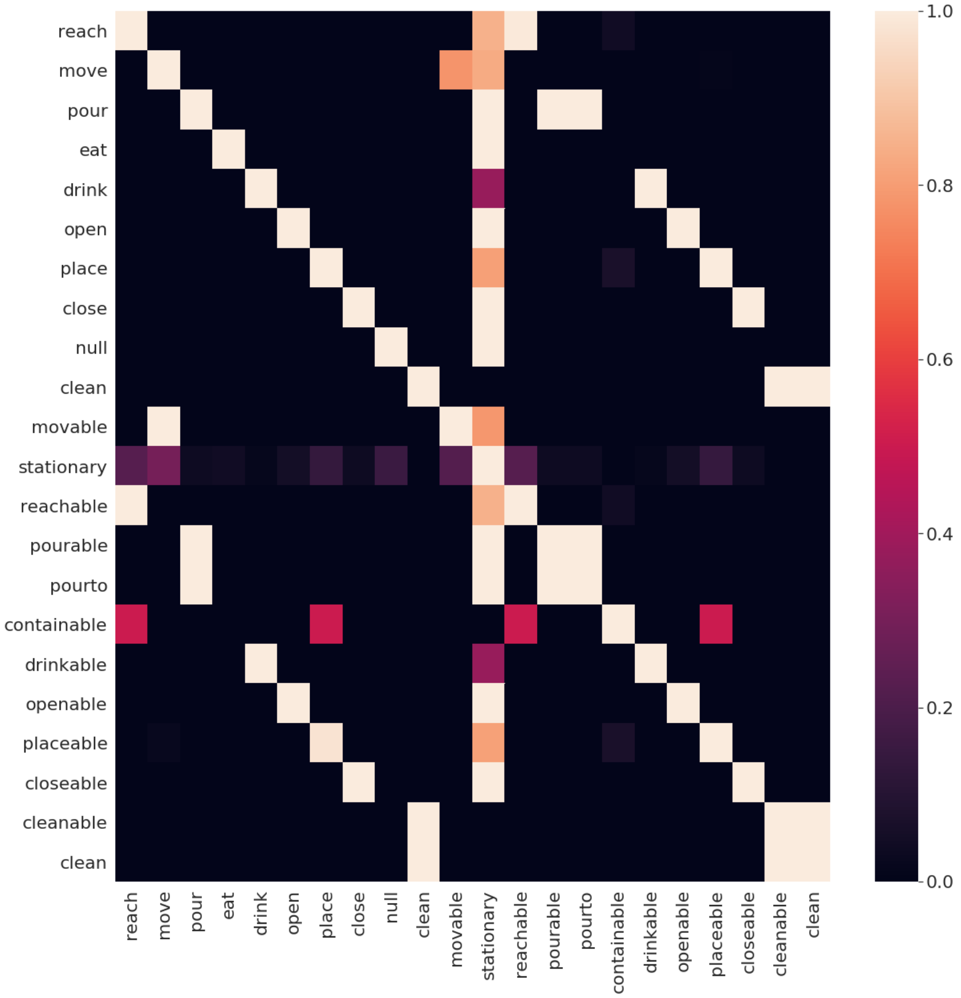
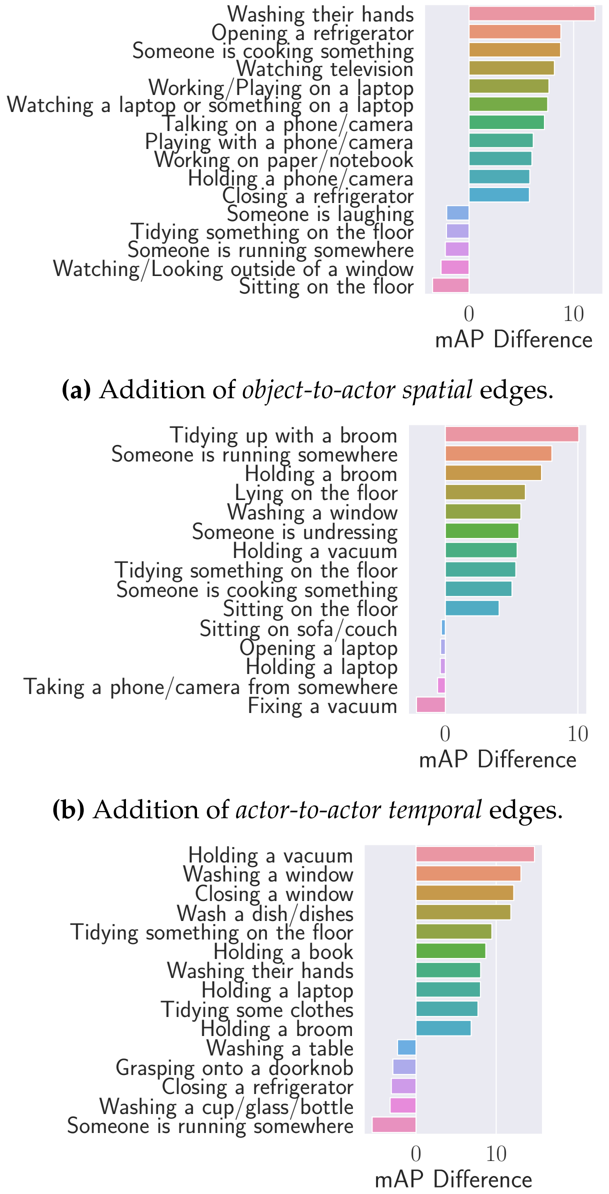


Figure 2.16: Symbolic graph adjacency matrix for CAD-120 dataset. Note that due to the small number and the nature of activity/affordance labels in this dataset, the graph is small, with sparse connections, many of which are rather obvious due to redundancy of labels (like *poorable* and *pourto*, or *clean* (activity), *clean* (affordance), *cleanable*). This might hinder our SCM from learning semantic context-aware node embeddings.



(a) Addition of *object-to-actor spatial* edges.

(b) Addition of *actor-to-actor temporal* edges.

(c) Addition of *visual-symbolic* edges.

Figure 2.17: Activity classes with the highest positive and negative performance difference by incrementally adding various types of graph edges. (a) Incorporating spatial structure benefits actions that involve interactions with objects far away from the actor, such as *watching television* or *cooking*. (b) Adding *actor-to-actor temporal* messages helps with long actions, such as *running*, and actions involving objects that are hard to detect (*Holding a broom*). (c) Adding visual-symbolic edges and performing global semantic graph convolutions benefits actions that have a few training examples, such as *Holding a vacuum* or have strong co-occurrences, such as *Holding a book*.

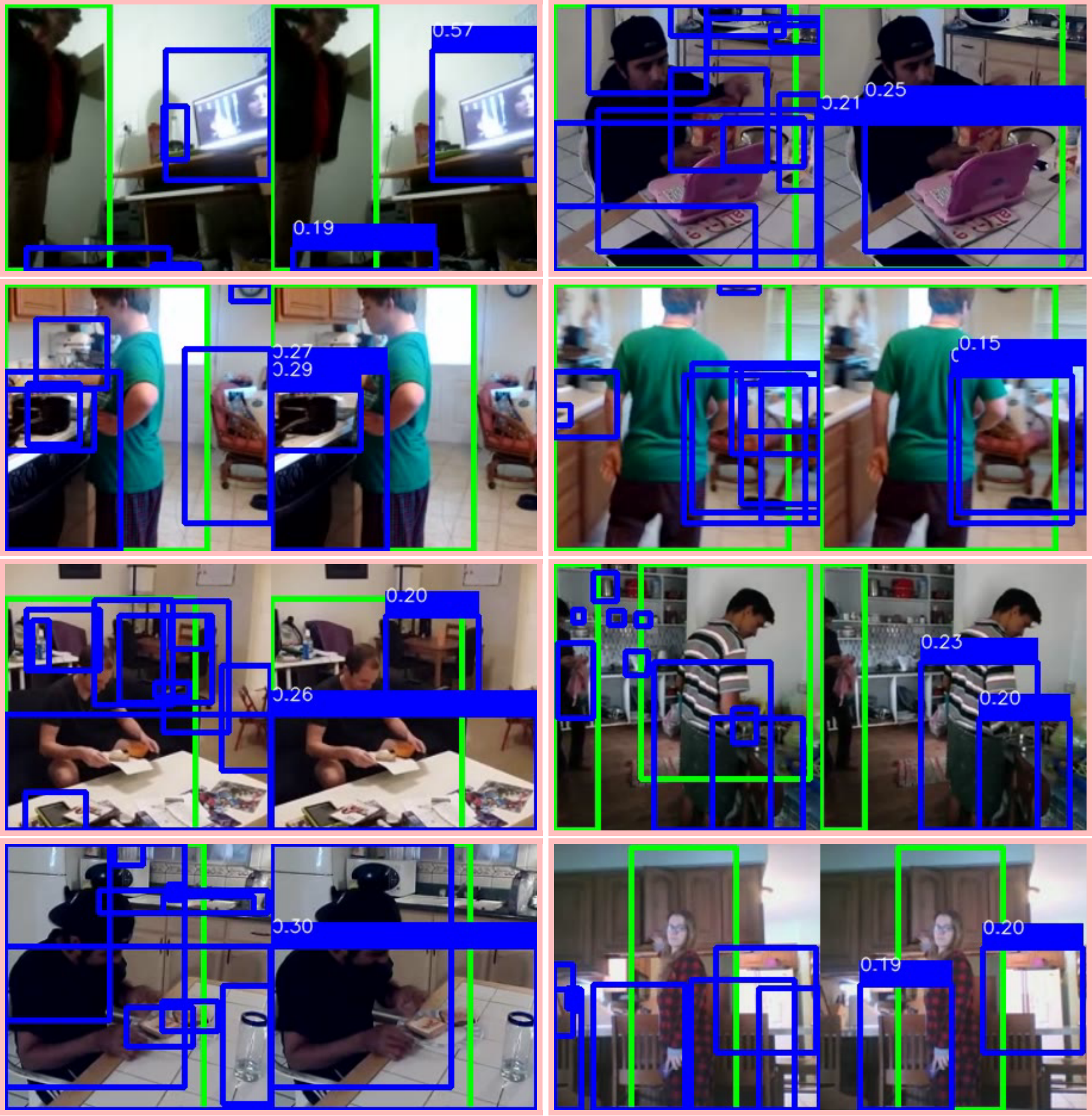


Figure 2.18: Visualization of attention over objects for updating the embedding of the actor on sample frames from Charades dataset. Each pair of images shows: the original frame with the actor detection in green and object detections in blue (*left*) and the actor and the two objects with largest attention coefficients (*right*).

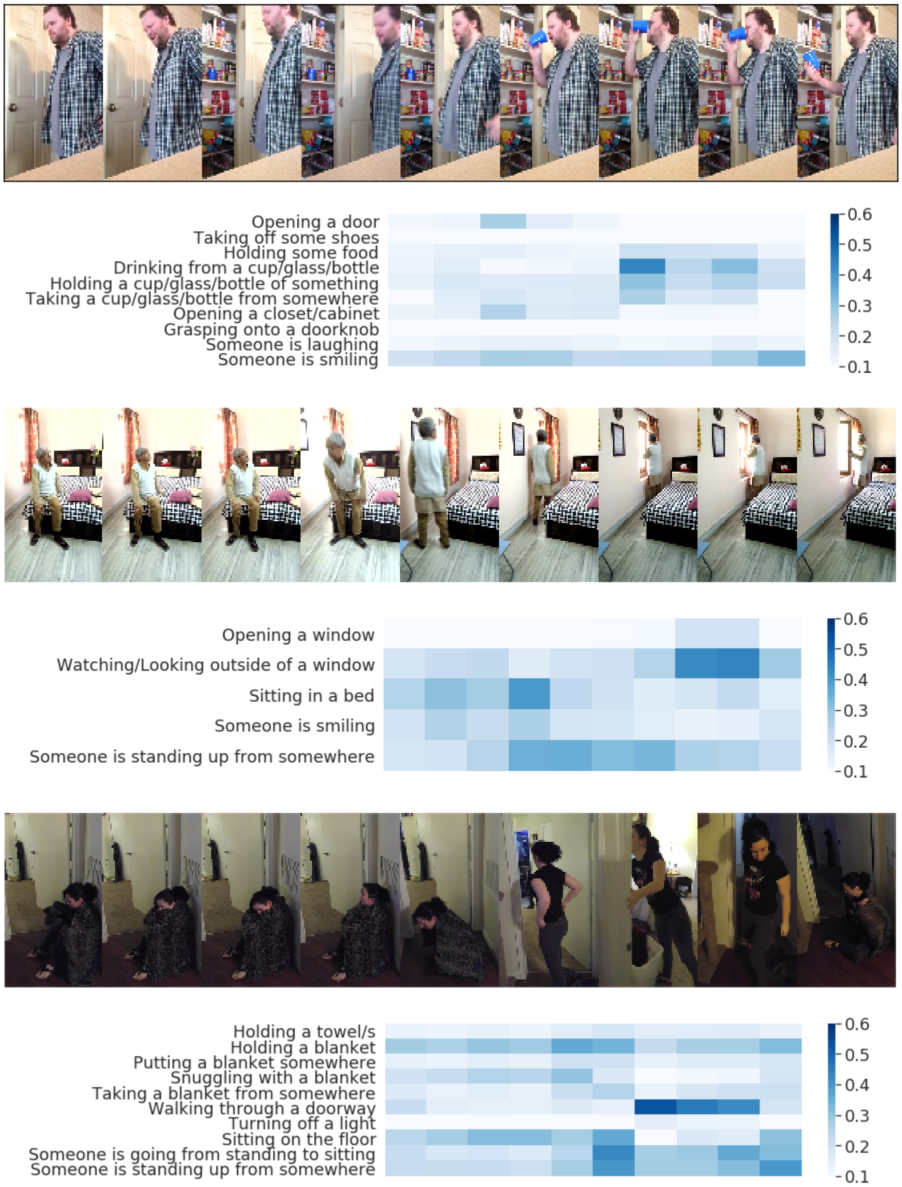


Figure 2.19: Temporal action predictions when using region embeddings computed by our V-HetGAT model for three sample videos from the Charades validation set. Action scores for 10 equidistant frames are shown for each video.

Chapter 3

Discriminative and Conditional Generative Region-based Models for Language-Driven Object Grounding in Videos

3.1 Problem Formulation

In this chapter, we are interested in grounding object words of ground-truth or generated visual descriptions of a video segment depicting an event. In contrast to our temporal activity detection experiments in Chapter 2, for the tasks tackled in this chapter, we assume that the temporal extent of the event is known a priori.

Let Y denote a visual description of a given video segment I with a duration of L frames. We represent $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ as a sequence of T words from a vocabulary $\mathcal{V}$, where $\mathbf{y}_t$ is the one-hot encoding of the t-th word, i.e., $\mathbf{y}_t \in \{0, 1\}^{|\mathcal{V}|}$ and $\|\mathbf{y}_t\|_1 = 1$. In the *VOG* task, the goal is localize each groundable word (e.g. object nouns and pronouns) in the input video, i.e., we are interested in localizing each mentioned groundable word with a bounding box $\widehat{\mathbf{b}}_t$ at each annotated frame for every word $\mathbf{y}_t$ in the sentence that belongs to a pre-defined groundable vocabulary $\mathcal{V}_o \subseteq \mathcal{V}$. In the *GVD* task, the goal is to both generate a visual description $\widehat{Y}$ and localize each generated groundable word $\widehat{\mathbf{y}}_t$ with a bounding box $\widehat{\mathbf{b}}_t$ at each annotated frame.

Our goal is to design a model that can tackle both tasks. To achieve this, we treat the problem of grounding as a problem of word-to-region alignment by leveraging region proposals. These region proposals $R = \{\mathbf{r}_m \in \mathbb{R}^4\}_{m=1}^{M}$ are obtained by an off-the-shelf object detector that is trained on an external image dataset. In particular, we use a Faster-RCNN object detector trained on the Visual Genome image dataset [182]. The object detector is applied at L_f selected frames (e.g., uniformly sampled frames) with M' regions detected at each frame ($M = L_f \times M'$). Then, the grounding problem is reduced to identifying which region at each selected frame corresponds to the t-th word.

3.2 Related Work

Grounding text in images and videos is an active research field in the intersection of vision and language. Depending on the type of text that is being grounded, grounding tasks can be divided into two groups: the first group of methods attempts to ground a single sentence or phrase (referring expression) in the video [183, 184, 185], while the second group tries to find correspondences between video regions and multiple parts of a sentence, such as object words [4, 186]. Since in this thesis we are interested in video understanding based on actors and objects, we focus on this latter, more fine-grained task of grounding each object word of a sentence in a video.

3.2.1 Fully-supervised Grounded Description and Object Grounding in Videos

Early attempts at visual grounding of phrases or object words focused mostly on images rather than videos. For example, Gorniak et al. [187] attempt to ground objects on images of constrained, synthetic scenes and employ semantic parsers. Another line of work relies on probabilistic models [188, 189], such as CRFs, for associating words in sentences to visual concepts, like 3D objects [189]. The parameters of these models are learned based on labeled training sets, where each training sample consists of an image, a sentence and ground-truth word-to-region alignments. Deep learning approaches for visual grounding have also been primarily focused on images and can be grouped in "region-proposal-based" approaches, which treat grounding as ranking of multiple region proposals, and "one-stage" approaches, which directly predict regions as outputs.

Region-proposal-based grounding. Building upon advances in object detectors and region proposal networks, these approaches address grounding by associating words with region proposals. For example, captioning-based approaches [4] propose learning grounding by generating sentences word-by-word using an attention mechanism over region proposals. The ground-truth text-to-region alignments are used to supervise the attention mechanism, so that it attends to the regions associated with each word. Dogan et al. [190] also leverage region proposals and employ recurrent neural networks to perform sequential grounding on images. Their framework applies an LSTM on the sequence of phrases to be grounded, starting from the last phrase, an LSTM on the region proposals and an LSTM on the sequence of phrase-box pairs, capturing the context of already grounded phrases. The model is trained by penalizing incorrect grounding decisions (phrase-grounding box pairs). To facilitate the learning of meaningful attention coefficients over regions, Zhou et al. [4] learn contextual region embeddings using self-attention applied on top of the set of regions from all frames. We also follow the same approach and focus on learning contextualized region representations based on visual and semantic interactions between actors and objects.

One-stage methods. Inspired by single-stage object detectors like YOLO [191] and DETR [192], these methods [193, 194, 195] directly predict bounding boxes grounding each word/phrase. For example, Yang et al. [194] adds a textual embedding to the YOLOV3 object detector for visual grounding in images. MDETR [195] adapts DETR to the task of visual grounding and employs large-scale pretraining on text-image pairs. It extracts image and text features using a convolutional backbone and a transformer language model, respectively, projects them into a shared embedding space and applies a joint transformer on them. Then, as in DETR, a transformer decoder is applied to object queries and bounding boxes are predicted by attending to the encoder's hidden states. MDETR has also been used for frame-level visual grounding in videos [186].

Going beyond grounding words given a human-annotated sentence, the related task of jointly generating a visual description and grounding the generated words to their corresponding visual regions (called Grounded

Visual Description) has attracted the interest of the research community [81, 4, 186]. Zhou et al. [4] generate video descriptions with grounded object words by using a video captioning model with a region attention mechanism, while Ruan et al. [186] generate captions with a holistic, captioning model, which does not look into regions and then apply a separate grounding model on the generated captions.

3.2.2 Weakly-Supervised Grounded Visual Description

Developing models that can both generate a sentence and link the generated words to their corresponding visual regions and which can be trained only with video-description pairs is a nascent research area. Zhou et al. [4] proposed using attention-based captioning models for generating sentences. Their GVD model then grounds words based on region attention coefficients. However, in contrast to prior work on phrase grounding that computes attention using the whole phrase as query [196], GVD attention is computed based on previous words (partially generated sentence), thus it is agnostic to the word being grounded.

A recent line of work has attempted to mitigate this issue. Ma et al. [197] proposed a cyclical training regime for WS-GVD of images and videos that involves two attention mechanisms: one based on the partial caption and another based on the groundable word. By forcing the words generated using these two attention mechanisms to match the ground-truth words, the mechanisms are implicitly regularized to produce similar attention weights during training. Other approaches explicitly supervise the region attention during training on image-caption pairs, either by using attention coefficients based on future relevant words [198], or by leveraging the word-to-region alignments of a separately trained image-to-text matching model [199]. In summary, a common thread in prior work is the usage of a regular region attention module of an UpDown [5] captioning model for grounding, which is regularized *only during training* based on auxilliary models or attention mechanisms. In contrast, inspired by discrete latent-variable models for image captioning/neural machine translation [200, 201, 202, 203], our key innovation is to treat word-to-region alignments as discrete latent variables in a grounded visual description CVAE model and exploit the prior or approximate posterior alignment distributions to infer the latent word-to-region alignments. This enables us to consider the past, future and current words for localizing each object word in the input image or video *during testing*.

3.2.3 Weakly-Supervised Visual Object Grounding

Grounding words (rather than whole sentences [82] or phrases [83, 84]) in images and videos is an active research field in the intersection of vision and language. Early attempts for weakly-supervised visual grounding given textual descriptions of images and videos relied on graphical models [85, 86]. Powered by advances in region proposal generation, a large group of recent methods [87, 88] cast the task as a *Multiple Instance Learning* (MIL) problem. These methods define an image-sentence matching score determined by word-to-region alignments and learn how to correctly match images to sentences using ranking losses. Such methods have also been extended to videos [89, 90, 9] with frame-sentence matching scores and mechanisms to account for missing objects. However, these MIL-based methods cannot both generate sentences and ground objects. This limitation is lifted by the *captioning-based* GVD-Grd method [4], which grounds each word based on region attention coefficients, computed with the previous words as query, combined with region-to-class similarity coefficients. These are obtained by transferring object class knowledge from external datasets. In this work, we also use captioning as a downstream task, but we localize words with the distributions of a conditional generative model, leveraging the full sentence context.

3.2.4 Joint Vision-Language Representation Learning

Inspired by advances in pretrained NLP models [204], researchers have also started to use large-scale vision-text corpora to learn task-agnostic, cross-modal vision-language representations, which can be used to facilitate downstream tasks. For example, Transformer-based models [205, 206] learn task-agnostic, visiolinguistic representations using only pairs of images with object proposals and associated textual descriptions. Our goal, however, is fundamentally different as we are interested in training visual grounding systems on small-scale datasets. Importantly, we rely on text as weak supervision for learning how to ground and we do not need bounding box annotations directly on the target dataset. While transformer models can also be used to tackle downstream tasks such as referring expression grounding [205], we note that this requires finetuning on a smaller, fully-annotated dataset.

3.2.5 Modeling Sequential Data with Variational Autoencoders

Our proposed CVAE-based captioning model is also related to regular or Conditional VAEs that are developed for modeling sequential data in NLP applications. In particular, VAEs with *sequences of latent variables* [207, 208, 209, 210, 211, 212] instead of a single latent variable driving the whole sequential generation process [213, 214, 215, 216] are more closely related to our work. However, the majority of those have non-interpretable, continuous latent variables, unlike our discrete latent word-to-region alignments. A notable exception is the approach of Graber et al. [217] that uses sequential discrete variables to model interactions between entities in interacting systems. Still, all these works share the same goal of modeling the likelihood of sequential data, while we propose exploiting the latent variables for grounding. To this end, we need to avoid training an inference model that produces posteriors almost identical to the prior, thus ignoring the word to be grounded. Researchers are actively exploring various techniques to mitigate this *posterior collapse* issue by modifying: the training objective [218, 219, 220, 221, 222, 223], the training procedure [224] or the decoder architecture [225]. Similarly, we propose controlling the relative factor between sentence reconstruction term and the prior regularization term [218, 226, 223]. In the previous chapter we developed models for region representation learning based on spatio-temporal heterogeneous graphs. We showed how the refined, context-aware representations can be used to improve performance in labeling tasks, such as predicting multiple activity labels per frame, or predicting subactivity and affordance labels for actor and object regions, respectively. However, predicting labels from a predefined set of possible activity categories/object affordances is not sufficient for conveying the rich content of visual data. To understand a video, a computer vision system needs to capture objects, actions, attributes and other semantic components. Natural language sentences are a compact way to describe this rich visual content. When used as annotations for training, textual descriptions allow algorithms to learn richer semantic concepts. Furthermore, by describing visual content with natural language, machines can better communicate with humans. Due to these advantages, the interdisciplinary, multimodal field of vision and language has recently attracted the interst of the computer vision community.

In this chapter, we explore region-based approaches for language-driven video understanding. Linking words to visual regions provides a fine-grained bridge between the vision and language modalities and is a fundamental block of many applications, such as human-robot interaction [227, 228], visual question answering [229, 230], and even unsupervised neural machine translation [231]. Therefore, visual grounding (associating linguistic symbols to visual entities) has become a prominent research area at the intersection of vision and language [196, 87, 190, 232] and is the focus of this chapter. In particular, we consider two related tasks, (1) *Visual Object Grounding (VOG)*, where given an input video (or image) and its visual description, the goal is to *localize* the referred semantic entities in the visual input, and (2) *Grounded Visual Description (GVD)*, where given an input video (or image), we must jointly *generate* a natural language description and *localize* the generated words.

Training visual grounding systems typically requires annotations of textual descriptions combined with bounding boxes for each groundable word. In Section 3.3, we begin our exploration of region-based language-driven video understanding systems in this fully-supervised setting. However, since constructing datasets with such fine-grained bounding box annotations is rather time-consuming and costly, in Section 3.4 we propose novel models for solving the VOG and GVD tasks that can be trained with weak supervision in the form of video-caption pairs.

3.3 Fully-Supervised Generation and Grounding of Visual Descriptions with Discriminative Models

In this section, we address the VOG and GVD tasks under the assumption that full supervision about the object words and their corresponding bounding boxes is given. We begin by giving a detailed description of the GVD attention-based encoder-decoder model proposed by Zhou et al. [4], which is the basis of our region-based frameworks for fully-supervised GVD and VOG.

3.3.1 Review of the GVD Attention-based Model

Given an input RGB video segment I with candidate region proposals R extracted by an off-the-shelf object detector, the GVD model proposed by Zhou et al. [4], illustrated in Fig. 3.1, employs an encoder-decoder architecture which consists of a visual encoder and a language decoder that communicate via cross-modal attention mechanisms. The encoder and decoder parameterize

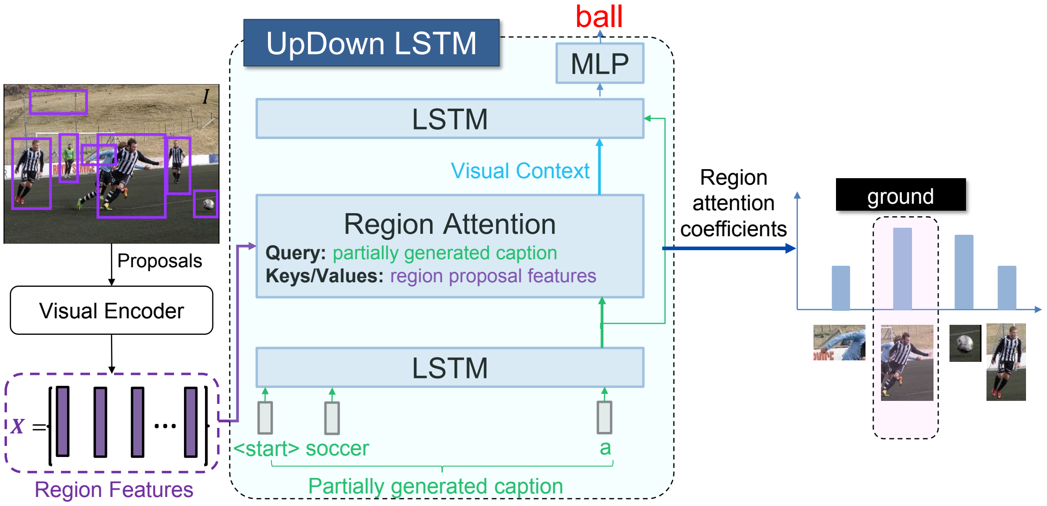

ball
UpDown LSTM
MLP
LSTM
Visual Context
Region Attention
Query: partially generated caption
Keys/Values: region proposal features
LSTM
<start> soccer
a
Partially generated caption
I
Proposals
Visual Encoder
X =
Region Features
Region attention coefficients
ground

Figure 3.1: Overview of the GVD model proposed by Zhou et al. [4] for tackling the Visual Object Grounding (VOG) and Grounded Video Description (GVD) tasks. Given an input RGB video segment with region proposals, the GVD model learns how to ground words by learning how to generate captions with a hierarchical LSTM equipped with a region attention mechanism. The region attention coefficients are used to ground each word by selecting the region with maximum attention coefficient at each frame.

the conditional distribution of a sentence Y given the region proposals R and the image I, i.e., $p(Y \mid R, I) = \prod_t p(\mathbf{y}_t \mid \mathbf{y}_{<t}, R, I)$, and are described next.

Visual encoder. The visual encoder consists of a *holistic video encoder* and a *region encoder*, as shown in Fig. 3.2.

The *video encoder* extracts a sequence of holistic frame features $F = \{\mathbf{f}_l\}_{l=1}^L$ by (a) passing the RGB frames of the video segment through a 2D convolutional network $\phi_{\text{RGB}}(\cdot)$ to yield a sequence of RGB feature vectors $\left\{\hat{\mathbf{f}}_l^{\text{RGB}}\right\}_{l=1}^L$, (b) passing optical flow frames through another 2D convolutional network $\phi_{\text{OF}}(\cdot)$ to extract a sequence of flow features $\left\{\hat{\mathbf{f}}_l^{\text{OF}}\right\}_{l=1}^L$, and then (c) further modeling long-term temporal dynamics by applying a BiGRU [168] RNN on

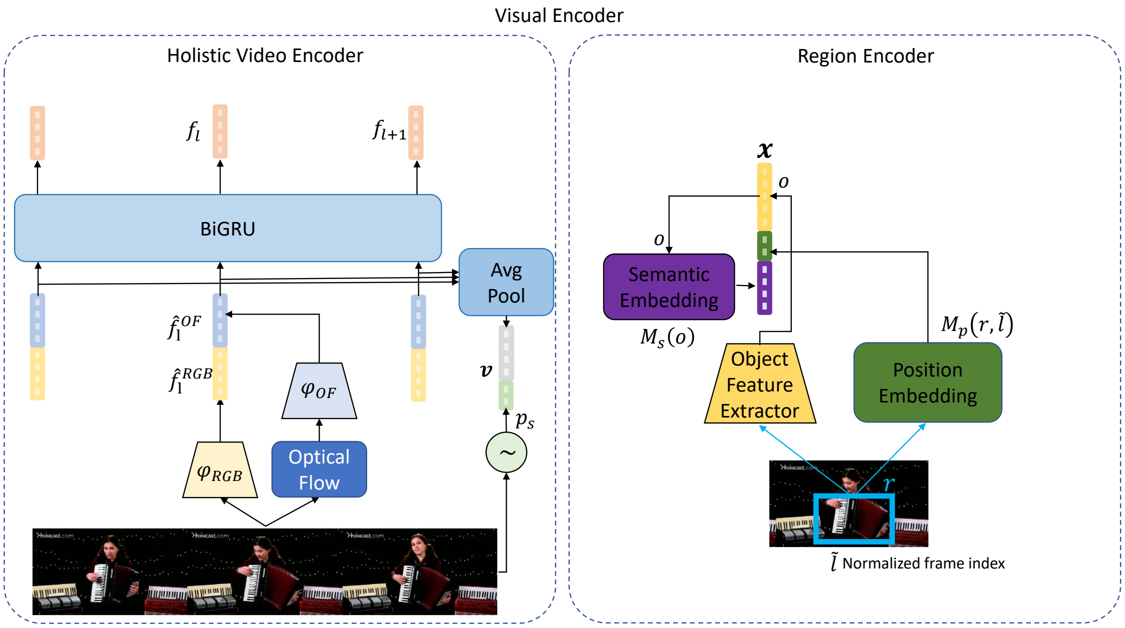


Figure 3.2: Overview of the visual encoder which represents a video based on holistic, frame-level information, as well as based on regions extracted using an off-the shelf object detector.

top the sequence of concatenated RGB features and motion features:

$$\hat{\mathbf{f}}_l^{\text{RGB}} = \phi_{\text{RGB}}(I_l) \in \mathbb{R}^{D_r}, \tag{3.1}$$

$$\hat{\mathbf{f}}_l^{\text{OF}} = \phi_{\text{OF}}(O_l) \in \mathbb{R}^{D_o}, \tag{3.2}$$

$$\overrightarrow{\mathbf{f}_l} = \text{BiGRU}(\overrightarrow{\mathbf{f}_{l-1}}, [\hat{\mathbf{f}}_l^{\text{RGB}}; \hat{\mathbf{f}}_l^{\text{OF}}]) \tag{3.3}$$

$$\overleftarrow{\mathbf{f}_l} = \text{BiGRU}(\overleftarrow{\mathbf{f}_{l+1}}, [\hat{\mathbf{f}}_l^{\text{RGB}}; \hat{\mathbf{f}}_l^{\text{OF}}]), \tag{3.4}$$

$$\mathbf{f}_l = [\, \overrightarrow{\mathbf{f}_l}\, ;\, \overleftarrow{\mathbf{f}_l}\,] \in \mathbb{R}^{D_f}, \tag{3.5}$$

where O_l is the l-th optical flow frame and $[\cdot;\cdot]$ denotes the concatenation operation.

The frame features are summarized in a global feature vector $\mathbf{v}$ describing the video segment, which is obtained by averaging the temporal sequence of concatenated frame-wise appearance features $\left\{\hat{\mathbf{f}}_l^{\text{RGB}}\right\}_{l=1}^{L}$ and motion features $\left\{\hat{\mathbf{f}}_l^{\text{OF}}\right\}_{l=1}^{L}$. Recall that the input video is assumed to be a known temporal segment of a longer untrimmed video. To integrate that information to the global feature vector, it is also augmented with a D_s-dimensional embedding $\mathbf{p}_s$ of the segment positional information (i.e., total number of segments, segment index, start time (normalized by the video duration) and normalized end time):

$$\mathbf{v} = \left[\frac{1}{L}\sum_l \hat{\mathbf{f}}_l^{\text{RGB}}; \frac{1}{L}\sum_l \hat{\mathbf{f}}_l^{\text{OF}}; \mathbf{p}_s\right] \in \mathbb{R}^{D_r+D_o+D_s}. \tag{3.6}$$

The *region encoder*, namely the second module of the visual encoder, represents each region proposal with the following three features: (a) an object appearance feature, (b) a learnable position embedding and (c) a semantic embedding of region-to-class similarity scores. The **object appearance feature** $\mathbf{o} \in \mathbb{R}^{d_o}$ is extracted from the penultimate layer of the object detector. The **position embedding** $M_p([\mathbf{r};\tilde{l}]) \in \mathbb{R}^{d_p}$ is a learnable embedding (linear transformation layer followed by a ReLU non-linearity) of the bounding box coordinates $\mathbf{r}$ (normalized by the frame size) and the index $\tilde{l}$ of the frame that the region belongs to (normalized by the number of frames in the video segment):

$$M_p([\mathbf{r};\tilde{l}]) = \text{ReLU}(W_p^T[\mathbf{r};\tilde{l}] + \mathbf{b}_p) \in \mathbb{R}^{d_p}. \tag{3.7}$$

The **semantic embedding** $M_s(\mathbf{o}) \in \mathbb{R}^{|\mathcal{V}_o|}$ is computed by applying a set of object classifiers on the region feature and normalizing the resulting scores across the number of groundable object words with the softmax operator:

$$M_s(\mathbf{o}) = \text{softmax}\left(W_s^T \mathbf{o} + \mathbf{b}_s\right) \in \mathbb{R}^{|\mathcal{V}_o|}, \tag{3.8}$$

where $W_s = [\mathbf{w}_1; \ldots; \mathbf{w}_{|\mathcal{V}_o|}] \in \mathbb{R}^{d_o \times |\mathcal{V}_o|}$ is a matrix of $|\mathcal{V}_o|$ object classifiers and $\mathbf{b}_s = [b_1; \ldots; b_{|\mathcal{V}_o|}] \in \mathbb{R}^{|\mathcal{V}_o|}$ is a vector of scalar biases. To initialize these weights and biases, we transfer object class knowledge from the external image dataset that the object detector was trained on, by finding the nearest neighbor from the annotated object classes of the external image dataset for each one of the $|\mathcal{V}_o|$ groundable words. Then, we initialize W_s and $\mathbf{b}_s$ with the corresponding object classifiers, i.e., the weights and biases, from the last linear layer of the object detector.

The object appearance feature, position embedding and semantic embedding are concatenated and a parameterized linear transformation $W_g \in \mathbb{R}^{d \times \left(d_o + d_p + |\mathcal{V}_o|\right)}$ is applied to obtain the grounding-aware region feature $\mathbf{x}$ introduced by Zhou et al. [4]:

$$\mathbf{x} = W_g \left[\mathbf{o}; M_p([\mathbf{r}; \tilde{l}]); M_s(\mathbf{o})\right] \in \mathbb{R}^d. \tag{3.9}$$

Language decoder. The distribution over words in the vocabulary $p(\mathbf{y}_t|\mathbf{y}_{<t}, R, I)$ is a categorical distribution conditioned on the partial caption $\mathbf{y}_{<t}$, region proposals R, and the input image I. Zhou et al. parameterize it with a fully-connected layer $g(\cdot)$, that receives as input the state $\mathbf{s}_t$ of a trainable language

model that summarizes $\mathbf{y}_{<t}$, R and I:

$$g(\mathbf{s}_t) = \operatorname{softmax}(W_{cls}\mathbf{s}_t) \in \mathbb{R}^{|\mathcal{V}|}, \tag{3.10}$$

$$\mathbf{y}_t \mid \mathbf{y}_{<t}, R, I \sim \operatorname{Cat}(g(\mathbf{s}_t)), \tag{3.11}$$

where $W_{cls} \in \mathbb{R}^{|\mathcal{V}| \times d}$ are trainable weights (biases are omitted for brevity). The chosen language model is a variant of the UpDown LSTM [5], which is composed of trainable word embeddings, emb : $\mathbb{R}^{|\mathcal{V}|} \rightarrow \mathbb{R}^{d_w}$, and two LSTM [233] layers with hidden states $\mathbf{u}_t \in \mathbb{R}^d$ and $\mathbf{s}_t \in \mathbb{R}^d$, respectively. To leverage the output of the visual encoder, the language decoder also employs two cross-modal attention mechanisms: (a) an attention mechanism $f(\cdot, \cdot)$ over holistic frame features, and (b) an attention mechanism $k(\cdot, \cdot)$ over region features. These attention mechanisms aggregate visual context from relevant frames/regions given the hidden state $\mathbf{u}_t$ as query that summarizes the description generated so far.

Intuitively, the first LSTM (RNN^1) captures the history of previous words as well as the gist of the video and its hidden state $\mathbf{u}_t$ is used as the query for the two attention mechanisms, while the second LSTM (RNN^2) additionally captures the history of attended region and frame features:

$$\mathbf{u}_t = \text{RNN}^1 \left(\mathbf{u}_{t-1}, [\mathbf{v}; \operatorname{emb}(\mathbf{y}_{t-1})]\right) \tag{3.12}$$

$$\mathbf{s}_t = \text{RNN}^2 \left(\mathbf{s}_{t-1}, \left[\sum_{l=1}^{L} f^{(l)}(\mathbf{u}_t, F)\mathbf{f}_l; \sum_{i=1}^{M} k^{(i)}(\mathbf{u}_t, X)\mathbf{x}_i; \mathbf{u}_t\right]\right). \tag{3.13}$$

The attention mechanisms which aggregate visual context from relevant frames/regions given the hidden state $\mathbf{u}_t$ as query that summarizes the description generated so far are implemented using additive attention:

$$f^{(l)}(\mathbf{u}_t, F) = \frac{\exp\left(\mathbf{w}_f^T \tanh(W_f[\mathbf{u}_t; \mathbf{f}_l]))\right)}{\sum_{l'=1}^{L} \exp\left(\mathbf{w}_f^T \tanh(W_f[\mathbf{u}_t; \mathbf{f}_{l'}]))\right)} \tag{3.14}$$

$$k^{(i)}(\mathbf{u}_t, X) = \frac{\exp\left({\mathbf{w}_\mathbf{k}}^T \tanh(W_k[\mathbf{u}_t; \mathbf{x}_i])\right)}{\sum_{j=1}^{M} \exp\left(\mathbf{w}_k^T \tanh(W_k[\mathbf{u}_t; \mathbf{x}_j])\right)}, \tag{3.15}$$

where $\mathbf{w}_f \in \mathbb{R}^{d_a}$, $W_f \in \mathbb{R}^{d_a \times 2d}$, $\mathbf{w}_k \in \mathbb{R}^{d_a}$, and $W_k \in \mathbb{R}^{d_a \times 2d}$ are learnable attention weights and d_a is the hidden attention dimension.

3.3.2 AO-GVD: Grounding with Contextual Region Representations

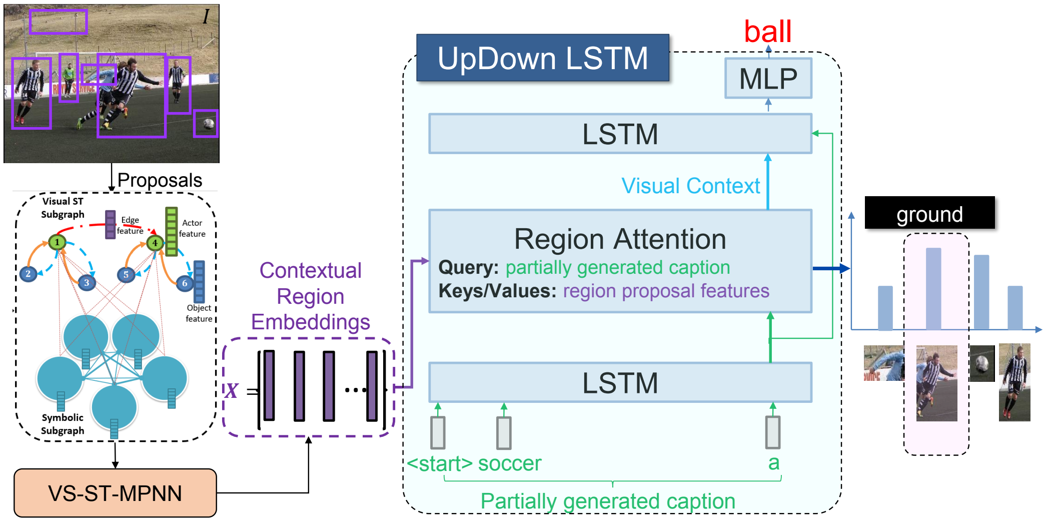


Figure 3.3: Illustration of our proposed AO-GVD model which builds upon the GVD model by computing contextual embeddings for the candidate region proposals based on local spatio-temporal visual interactions among actors and objects, as well as long-range semantic interactions captured in a heterogeneous visual-symbolic graph. These contextual region embeddings are then fed as input to the UpDown LSTM for generating grounded video descriptions.

We are now ready to describe how to augment the original GVD model proposed by Zhou et al. [4] with contextual region embeddings that take into account local spatio-temporal visual interactions among actors and objects, as well as long-range semantic interactions. We propose to do so by augmenting the visual encoder described in Section 3.3.1 with our VS-ST-MPNN region representation model, in order to refine the local, grounding-aware region features. An overview of our model, called Actor-Object Grounded Visual Description (AO-GVD), is shown in Fig. 3.3.

As we discussed in Chapter 2, the VS-ST-MPNN model is very flexible and can be applied to refine the local features of video regions as long as an appropriate visual-symbolic graph $G(R, I)$ is defined. The key question is: *how do we design an input visual-symbolic graph tailored to the task of grounding*? For the visual subgraph of $G(R, I)$, which is instantiated on top of the region proposals, we need to define: the node assignment function that assigns the *actor* or *object* type to each region proposal, the connectivity of the visual subgraph, i.e., which nodes are connected with directed edges and what are their edge types, and the node (edge) feature extraction function that is used to compute initial node (edge) attributes for each visual node (edge). For the symbolic subgraph of $G(R, I)$, we need to define the number of symbolic nodes, their initial attributes and the scalar symbolic edge weights. Recall that our VS-ST-MPNN assumes that symbolic edges are not associated with edge attributes. Next, we present our proposed visual-symbolic graph construction for the grounded visual description and visual object grounding tasks.

Visual spatio-temporal subgraph. Each node of the visual subgraph corresponds to a region from the set of region proposals R which are extracted using an off-the-shelf object detector trained on the Visual Genome dataset. Therefore, each region is associated with a Visual Genome object class. Our node type assignment function utilizes these object classes and assigns to each region a type (actor or object) depending on the Visual Genome class associated to this region. Specifically, regions belonging to one of 42 manually selected classes of the Visual Genome dataset are assigned the *actor* node type. These object classes are: *adult, baby, biker, bride, boy, catcher, chef, child, couple, cyclist, driver, fire extinguisher, girl, guy, groom, kid, lady, little girl, male, man, men, mother, motorcyclist, officer, passenger, pedestrian, person, player, pitcher, police officer, policeman, racer, referee, rider, she, skateboarder, skater, skier, tennis player, umpire, woman, worker, young man*. The rest of detected regions which belong to the rest of the 1600 Visual Genome object classes or the background class) are assigned the *object* node type. The node attributes are initialized with the grounding-aware region feature:

$$\mathbf{h}_i^{(0)} = \phi_o(\mathbf{r}_i, I) = W_g \left[\mathbf{o}; M_p([\mathbf{r}_i; \tilde{l}]); M_s(\mathbf{o})\right] \in \mathbb{R}^d, i \in 1, \ldots, M. \tag{3.16}$$

Edge attributes are initialized with the relative spatial location and size of the connected regions:

$$\mathbf{q}_{ij}^{(0)} = \left[\log\left(\frac{|x_{tl}^{(i)} - x_{tl}^{(j)}|}{w_i}\right), \log\left(\frac{|y_{tl}^{(i)} - y_{tl}^{(j)}|}{h_i}\right), \log\left(\frac{w_i}{w_j}\right), \log\left(\frac{h_i}{h_j}\right)\right], \tag{3.17}$$

where $(x_{tl}^{(i)}, y_{tl}^{(i)})$ is the top-left corner of the bounding box of the i-th region and h_i and w_i are the height and width, respectively.

Symbolic subgraph and visual-symbolic edges. We use a symbolic subgraph that has nodes corresponding to the groundable object classes $\mathcal{V}_o$ and scalar edge weights corresponding to per-sentence groundable object class co-occurrences in training data. We initialize each symbolic node attribute with a pretrained word embedding of its associated groundable object class. The visual-symbolic edges densely connect every visual node to every symbolic node. Both the symbolic edges as well as the visual-symbolic edges are only associated with scalar edge weights and do not have an edge attribute.

Computation of context-aware region embeddings. Given the constructed visual-symbolic graph $G(R, I)$, we can apply our VS-ST-MPNN (and in particular its V-HetGAT and VS-HetGAT variants) to obtain context aware region embeddings X, i.e.:

$$X = \text{MPNN}(G(R, I)). \tag{3.18}$$

Semantic context module variants. As we discussed in Chapter 2, our VS-ST-MPNN does not assume known correspondences between visual nodes and symbolic nodes and instead employs learned scalar edge weights (soft assignments) when computing messages across these edges (Eq. 2.22 and Eq. 2.26). For example, for our experiments on activity detection in Section 2.5.2.3, we had no prior knowledge of the correspondence between visual nodes (actors) and symbolic nodes (activity labels). However, as we explained earlier in the approach of Zhou et al. [4], we can guess the groundable object word that each region might be related to by transferring object class knowledge from external image datasets to obtain region-to-class similarity scores. Inspired by this, we propose two variants of our Semantic Context Module (SCM): (a) using the original SCM that employs visual-symbolic soft-assignment weights that are learned from scratch (Eq. 2.22), and (b) using fixed visual-symbolic scalar edge weights, and in particular the region-to-class (i.e., visual-to-symbolic node) similarity scores defined in Eq. 3.8.

3.3.3 Inference

Inference for grounded visual description. Following Zhou et al. [4], we perform a greedy decoding for sentence generation. That is, after feeding the special symbol $< BOS >$ (beginning of sentence), we predict a word at each timestep t and feed it as input to the next timestep. Formally, using the word probability distribution defined in Eq. 3.10- 3.11, we obtain:

$$\hat{\mathbf{y}}_t = \underset{c=\{1,\ldots,|\mathcal{V}|\}}{\operatorname{argmax}} \; p(y_{t,c} = 1 \mid \hat{\mathbf{y}}_{<t}, R, I) = \underset{c=\{1,\ldots,|\mathcal{V}|\}}{\operatorname{argmax}} \frac{\exp\left(\mathbf{w}_{cls}^{(c)}\mathbf{s}_t\right)}{\sum_{c'} \exp\left(\mathbf{w}_{cls}^{(c')}\mathbf{s}_t\right)}, \tag{3.19}$$

where $\mathbf{w}_{cls}^{(c)}$ is a row vector corresponding to the c-th row of matrix W_{cls}.

Each generated word that belongs to the groundable object classes $\mathcal{V}_o$ is grounded at a frame l by selecting the region with maximum attention coefficient at that frame. Formally, let $\mathcal{J}_l \subset \{1, \ldots, M\}$ be the subset of region indices that were extracted from the l-th frame. Then, the predicted localization for the t-th predicted word $\mathbf{y}_t$ at frame l, denoted as $\hat{\mathbf{b}}_t^{(k)}$, is:

$$\hat{\mathbf{b}}_{t,l} = \mathbf{r}_j, \text{ where } j = \underset{i \in \mathcal{J}_l}{\operatorname{argmax}} \, k^{(i)}(\mathbf{u}_t, X), \tag{3.20}$$

where $k^{(i)}(\mathbf{u}_t, X)$ is the region attention coefficient for the i-th region (Eq. 3.15) and $\mathbf{u}_t$ is the hidden state of RNN^1, i.e., the Top-Down Attention LSTM following the terminology from Anderson et al. [5], that captures the partial generated caption $\hat{\mathbf{y}}_{<t}$ (Eq. 3.12).

Inference for visual object grounding. In this case we are given both a video and a ground-truth sentence describing it. One way of localizing a given groundable word at a frame l is by selecting the region with maximum attention coefficient at that frame, as in the case of Grounded Visual Description (Eq. 3.20), with the only difference that we are now feeding the ground-truth words to the UpDown LSTM instead of the generated ones.

Another way of localizing a given groundable word proposed by Zhou et al. [4] is to fuse the region attention with the object classifier knowledge. Intuitively, the region-to-class similarities obtained via object class knowledge transfer capture our prior belief regarding which object word each region is associated with, while the region attention coefficients capture the relevant region by taking account the sentence to be grounded. For example, assume that we are given a video showing a crowd gathered around a piano player and we would like to ground the word *man* in the following sentence: "A man in a striped shirt is playing the piano". Based on the object classifier trained on Visual Genome we can associate all regions corresponding to men in the scene to the object word "man". That knowledge can be fused with the region attention to ideally select the "man in a striped shirt". To achieve this, the unnormalized region attention coefficients are summed with the unnormalized region-to-class similarity scores obtained via object knowledge transfer (Eq. 3.8), and are normalized to sum to 1 across all regions of the video to yield region proposal ranking scores $d_{t,i}$:

$$d_{t,i} = \frac{\exp\left(k^{(i)}(\mathbf{u}_t, X) + M_s^{(c_t)}(\mathbf{o}_i)\right)}{\sum_{i'=1}^{M} \exp\left(k^{(i')}(\mathbf{u}_t, X) + M_s^{(c_t)}(\mathbf{o}_{i'})\right)}, \tag{3.21}$$

$$\hat{\mathbf{b}}_{t,l} = \mathbf{r}_j, \text{ where } j = \operatorname*{argmax}_{i \in \mathcal{J}_l} d_{t,i} \tag{3.22}$$

where c_t is the groundable object class of the t-th word ($y_{t,c_t} = 1$). We will refer to this variant as GVD-Grd in our experiments.

3.3.4 Training Objective

The parameters of the model are trained based on video-caption-grounding triplets $\{(I^{(n)}, Y^{(n)}, B^{(n)})\}_{n=1}^{N}$ by minimizing a cross-entropy loss for word prediction ($\mathcal{L}_{word}$), a cross-entropy loss for word localization based on the attention coefficients $k^{(i)}(\mathbf{u}_t, X)$ ($\mathcal{L}_{att}$), and a cross-entropy loss for similarity scores between regions and each groundable object class ($\mathcal{L}_{cls}$), respectively.

$$\mathcal{L} = \mathcal{L}_{word} + \mathcal{L}_{att} + \mathcal{L}_{cls} \tag{3.23}$$

The first term of the GVD objective is a cross-entropy loss that encourages the predicted word distribution to be peaky around the correct t-th word of the sentence:

$$\mathcal{L}_{word} = -\sum_{t=1}^{T} \sum_{w=1}^{|\mathcal{V}|} y_{t,w} \log p(y_{t,w} = 1 \mid \mathbf{y}_{<t}, R, I), \tag{3.24}$$

where $p(\mathbf{y}_t \mid \mathbf{y}_{<t}, R, I)$ is the predicted distribution over words of the vocabulary at timestep t given the ground-truth partial caption until timestep $t-1$, $\mathbf{y}_{<t}$.

The second term encourages the model to attend to the correct region when generating a groundable word:

$$\mathcal{L}_{att} = -\sum_{t \in \{t | y_t \in \mathcal{V}_o\}} \gamma_{t,i} \log k^{(i)}(\mathbf{u}_t, X), \tag{3.25}$$

where $\mathbf{k}(\mathbf{u}_t, X) \in \mathbb{R}^M$ the vector of attention coefficients over the M region proposals R and $\gamma_t = [\gamma_{t,1}, \ldots, \gamma_{t,M}]$ is the indicator vector for positive/negative region proposals for the t-th groundable word, i.e., $\gamma_{t,i} = 1$ if the i-th region proposal has an IoU over 0.5 with the ground-truth box b_t and $\gamma_{t,i} = 0$ otherwise. (If multiple region proposals match with the ground-truth box, we use the one with the largest IoU).

The third term encourages the region-to-class similarities to correctly classify regions to groundable words. For each region with IoU over 0.5 with any ground-truth box, we encourage the region-to-class-similarity $M_s^{(i)}$ to classify the region to the groundable object class of the ground-truth box (c_i). (If multiple ground-truth boxes match with the region proposal, we use the object class of the one with maximum IoU):

$$\mathcal{L}_{cls} = -\sum_{i \in \mathcal{R}^*} c_i \log M_s^{(i)}[c_i], \tag{3.26}$$

where $\mathcal{R}^*$ denotes the set of region proposals that match with at least a ground-truth box and $M_s^{(i)} \in \mathbb{R}^{|\mathcal{V}_o|}$ are the region-to-class similarities for each region i obtained from transferring object classifier knowledge.

3.3.5 Experiments

To evaluate the proposed extension of the GVD model [4] with contextual region embeddings computed with our VS-ST-MPNN model, we use the ActivityNet Entities dataset, described in Section 3.3.5.1. In Section 3.3.5.2, we present the metrics used to evaluate all models for the tasks of *GVD* and *VOG* and in Section 3.3.5.3 we provide implementation details. In Section 3.3.5, we provide the performance of the fully-supervised discriminative encoder-decoder model for various choices of contextual region embeddings models. Then, we present qualitative results to illustrate the benefits of performing graph-based representation learning on hybrid visual-symbolic graphs.

3.3.5.1 ActivityNet Entities Dataset

The ActivityNet Entities dataset [4] is a public benchmark dataset that contains 15k videos with captions and more than 158k annotated bounding boxes of referred semantic entities. In particular, it contains 52k video segments (with a training/validation/testing split of 35k/8.6k/8.5k segments, respectively) annotated with a caption each. Each groundable word in a sentence, i.e. each word from a vocabulary of 431 groundable words, is annotated with a bounding box in one out of 10 uniformly sampled frames of the video where it can be clearly observed. To the best of our knowledge, it is the only video dataset with bounding box annotations for each groundable word of training video captions, and thus suitable for training our fully-supervised contextual grounded video description model.

3.3.5.2 Evaluation Metrics

The performance of a model on the *VOG* task (grounding given ground-truth sentences) is measured with Box Accuracy [89, 9, 4], which computes the percentage of correctly localized word instances for each groundable object class. A word instance is considered to be correctly localized if the Intersection-over-Union (IoU) metric between the predicted box and the ground-truth box is above 0.5. We compute Box Accuracy for each word in the vocabulary of groundable words $\mathcal{V}_o$ and report their average.

Metrics for *Grounded Visual Description* evaluate both grounding and captioning capabilities. Following Zhou et al. [4], the quality of generated textual descriptions is measured using standard metrics, such as Bleu (B@1, B@4), METEOR (M), CIDEr (C), and SPICE (S). Most of these metrics, with the exception of SPICE, are similarity measures based on n-gram matching between the generated sentence and ground-truth sentence(s). Briefly, Bleu@N [234] is a modified n-gram precision metric with a sentence-brevity penalty, calculated as a weighted geometric mean over different length n-grams up to length N. METEOR [235] aligns sentences based on exact, stem, synonym, and paraphrase matches between words and phrases, and then computes a weighted F-score. CIDEr [236] computes the cosine similarity between TF-IDF-weighted n-grams of length n and averages those similarities across n-gram lengths from 1 to $N = 4$, where TF-IDF stands for term frequency-inverse document frequency. Intuitively, a higher weight is applied on n-grams that frequently occur in the ground-truth sentence, while the weight is reduced for n-grams that commonly occur across all textual descriptions in the dataset, and are thus less informative about the visual content. SPICE [237] aims to assess the similarity of generated and ground-truth sentences based on semantic propositional content. To do so, it extracts from each ground-truth and generated sentence a semantic scene graph encoding the objects, attributes and relations mentioned in the sentence , and calculates an F-score defined over the conjunction of logical tuples representing semantic propositions in the scene graphs. Each scene graph encodes the objects, attributes and relations present in the sentence.

To evaluate grounding performance on generated sentences, we adopt the $F1_{all}$ and $F1_{loc}$ grounding metrics [4]. The $F1_{all}$ metric is the harmonic mean of precision and recall, where a predicted bounding box for a word in a generated sentence is considered as correct when both the word is correctly predicted (i.e., the word appears in the ground-truth sentence) and the bounding box sufficiently overlaps with the ground-truth bounding box (IoU > 0.5). Since the $F1_{all}$ metric is affected both by whether the groundable word appears in the generated sentence and by the localization accuracy, the $F1_{loc}$ metric assesses only the localization quality by focusing only on word instances in the ground-truth and generated sentences that were correctly predicted.

Formally, given a groundable word class from the vocabulary $\mathcal{V}_o$ (e.g., *dog*), let us define the number of instances of that word in the generated sentences as A, the number of instances in the ground-truth sentences as B, the number of correctly predicted word instances in the generated sentences as C and the counterpart in the GT sentences as D, and the number of correctly predicted and localized word instances as E. Then,

$$\text{Prec}_{all} = \frac{E}{A}, \quad \text{Rec}_{all} = \frac{E}{B}, \quad F1_{all} = 2\frac{\text{Prec}_{all} \times \text{Rec}_{all}}{\text{Prec}_{all} + \text{Rec}_{all}}, \tag{3.27}$$

and

$$\text{Prec}_{loc} = \frac{E}{C}, \quad \text{Rec}_{loc} = \frac{E}{D}, \quad F1_{loc} = 2\frac{\text{Prec}_{loc} \times \text{Rec}_{loc}}{\text{Prec}_{loc} + \text{Rec}_{loc}}. \tag{3.28}$$

The precision and recall for the two metrics are computed for each groundable object class, but are set to zero if an object class has never been predicted. Finally, the scores are averaged by dividing them with the total number of object classes in the split. We compute these metrics on the validation set using the official evaluation scripts for captioning[1] and grounding[2]. The grounding results reported on the hidden test set were obtained in 2020 using the evaluation server [3].

3.3.5.3 Implementation Details

As explained in Section 3.3.2, we augment the grounded visual description model of Zhou et al. [4] (GVD) with our Visual and/or Semantic Context Modules. The GVD model uses a hierarchical LSTM decoder that generates a descriptive sentence based on global video features along with local region features of 100 region proposals extracted from 10 uniformly sampled frames of the video segment and it utilizes the attention coefficients to ground the

[1] https://github.com/LuoweiZhou/densevid_eval_spice/blob/bbab10c202e956266031a0dd6c791cba25b58e59/evaluate.py

[2] https://github.com/facebookresearch/ActivityNet-Entities/blob/aa5cd28383e5e9c63e875ada54057591a71509d9/scripts/eval_grd_anet_entities.py

[3] https://competitions.codalab.org/competitions/20537

nouns in the image.

GVD model hyperparameters. We use the same region proposals and features as Zhou et al. [4]. For each frame, we use a Faster R-CNN [72] detector with ResNext-101 [238] backbone pretrained on Visual Genome [182] to obtain region proposals. In particular, we retain the top 100 region proposals per frame, based on their detection confidence score. Each region is described by a $d_o = 2,048$-dimensional feature vector extracted from the $fc6$ layer of the ResNext-101. We also combine that region feature with a $d_p = 300$-dimensional trainable embedding of the bounding box coordinates (including the normalized frame index), and a 432-dimensional vector of object classification scores. We also use a global feature vector $\mathbf{v}$ of size $D_r + D_o + D_s = 3{,}122$. We use a vocabulary of 4,905 words including `UNK` (the symbol for rare words not included in the vocabulary) and `EOS` (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch, same as in GVD [4]. The region embedding/LSTM state size is $d = 1{,}024$, while additive attention mechanisms use a hidden layer of size $d_a = 512$.

Symbolic subgraph. Our symbolic graph has nodes corresponding to the 431 object classes. Based on our experimental results from employing the VS-ST-MPNN for temporal action localization (in Section 2.5.2.6), the VS-ST-MPNN is not sensitive to the choice of the symbolic graph connectivity, and using a dense graph performs almost as well as using action label co-occurrences as the scalar edge weights. Here we choose to use edge weights corresponding to per-sentence object label co-occurrences in training data. The adjacency matrix is binarized by thresholding co-occurence frequency values with a threshold of 0.2, removing spurious edges between object classes with very few co-occurrences. Figure 3.4 shows a part of the adjacency matrix. As we can see the matrix is relative sparse capturing meaningful connections, with the exception of some words like *it* or *person*, which frequently co-occur with most other groundable object classes. To obtain the linguistic embedding of each object, we use off-the-self GloVe [169] word embeddings of size $K = 300$.

VS-ST-MPNN hyperparameters. The hyperparameters of our VS-ST-MPNN model (variant VS-HetGAT (NodeOnly)) used in this dataset are: $L^v = 2$, $d_V = 1{,}024$, $L^s = 2$, $d_S = 256$. We used *object-to-object spatial* and *actor-to-object spatial* edges. Our batch size is 80 video clips, the learning rate is set to 0.0003 and we train for 30 epochs.

Visual subgraph. The actor nodes for each frame of the clip correspond to the top 10 object detections that belong to one of the 42 manually defined actor classes, defined in Section 3.3.2. The object nodes correspond to the remaining 90 object detections per frame, including background detections.

3.3.5.4 Experimental Results

Table 3.1 compares the performance of various variants of our AO-GVD model, stemming from different choices of the contextual embedding module. We compare: (a) multi-head attention (MHA) applied on the fully-connected, homogeneous visual graph instantiated on top of region proposals, (b) our V-HetGAT model employing heterogeneous visual message passing (Visual Context Module) and visual-symbolic message passing (Semantic Context

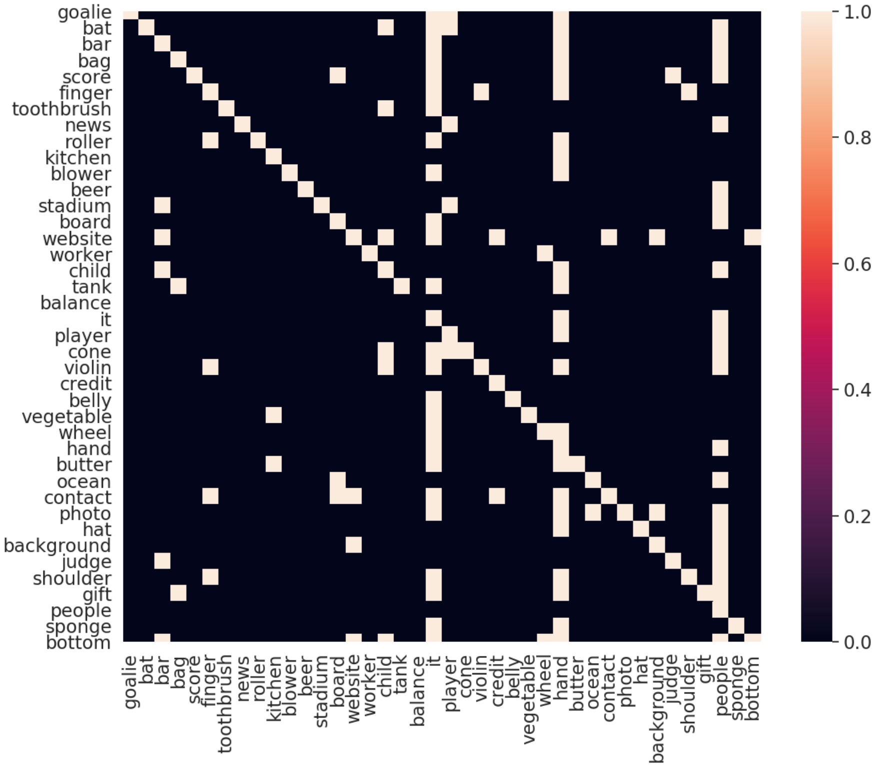


Figure 3.4: Illustration of per-sentence co-occurrences of a subset of groundable object words from the training set of the ActivityNet Entities dataset.

Module), (c) our original Semantic Context Module applied on top of visual context-aware region embeddings obtained via self-attention (MHA + SCM), (d) a modified version of the Semantic Context Module that uses fixed visual-symbolic scalar edge weights (SCM-VG), as we described in Section 3.3.2. As it can be seen, replacing MHA with our visual module does not improve captioning metrics, but it improves grounding of generated words with a relative improvement of 4% in $F1_{all}$ (24.1% → 25.2%). Adding our Semantic Context Module to MHA leads to an improvement across all Grounded Visual Description metrics, which is even more pronounced in the test set (improving CIDEr from 45.5 to 47.7%). Note that the initial region features

(Eq. 3.9) already capture semantic information by including region-to-class similarities. Therefore, the improvement in captioning cannot be attributed solely to the inclusion of semantic context, but rather to our semantic reasoning framework. Last, from the superior captioning performance of our third variant (AO-GVD, MHA+SCM-VG), we conclude that prior knowledge about correspondences between visual and symbolic nodes, if available, can possibly facilitate representation learning on the hybrid graph.

Table 3.1: Grounded visual description results on ActivityNet Entities [4] using the following metrics: Bleu (B), METEOR (M), CIDEr (C), SPICE (S), $F1_{all}$ and $F1_{loc}$. MHA: multi-head self-attention. VCM: Visual Context Module. SCM: Semantic Context Module. SCM-VG: our semantic context module with visual-to-symbolic node correspondences transferred from Visual Genome.

	$B@1$	$B@4$	M	C	S	$F1_{all}$	$F1_{loc}$
Validation set							
GVD (MHA) [4]	**23.9**	2.59	11.2	47.5	15.1	7.11	24.1
GVD (VCM + SCM) (ours)	23.4	2.41	11.1	47.3	14.8	7.28	25.2
GVD (MHA + SCM) (ours)	23.8	2.67	**11.3**	48.6	**15.2**	**7.35**	**25.3**
GVD (MHA + SCM-VG) (ours)	**23.9**	**2.78**	**11.3**	**49.1**	15.1	7.15	24.0
Test set							
Masked Transformer [239]	22.9	2.41	10.6	46.1	13.7	-	-
Bi-LSTM+TempoAttn [239]	22.8	2.17	10.2	42.2	11.8	-	-
GVD (MHA) [4]	23.6	2.35	11.0	45.5	14.9	7.59	25.0
AO-GVD (VCM + SCM) (ours)	23.1	2.34	10.9	46.1	14.5	-	-
AO-GVD (MHA + SCM) (ours)	23.6	2.54	11.2	47.7	15.0	7.30	24.4
AO-GVD (MHA + SCM-VG) (ours)	**24.1**	**2.63**	**11.4**	**49.0**	**15.1**	**7.81**	**27.1**

In Table 3.2 we report the Box Accuracy metric for grounding semantic entities mentioned in ground-truth visual descriptions. Similar to captioning, we do not observe a significant difference between our heterogeneous visual message passing model (V-HetGAT) and multi-head attention in this

Table 3.2: Visual Object Grounding results on the ActivityNet Entities [4] validation set. *MHA*: multi-head self-attention applied on region features (grounding-aware region features: projection of the concatenation of object detector feature, spatio-temporal location embedding and object class distribution.). *Dense ST*: fully-connected, homegeneous visual spatio-temporal graph instantiated on top of region proposals. *Spatial*: heterogeneous, spatial graph instantiated on each frame. *Attn.*: Grounding based on soft attention, *Grd*: Grounding by fusing region attention with object classifier knowledge. GVD* denotes our implementation and training of the GVD model.

	Visual Graph	Box Acc. (Attn.)	Box Acc. (Grd)
GVD (MHA) [4]	Dense ST	**34.5**	**41.6**
GVD* (MHA) [4] (ours)	Dense ST	31.89	40.45
AO-GVD (VCM + SCM) (ours)	Spatial	31.15	39.47
AO-GVD (VCM) (ours)	Spatial	32.10	40.27

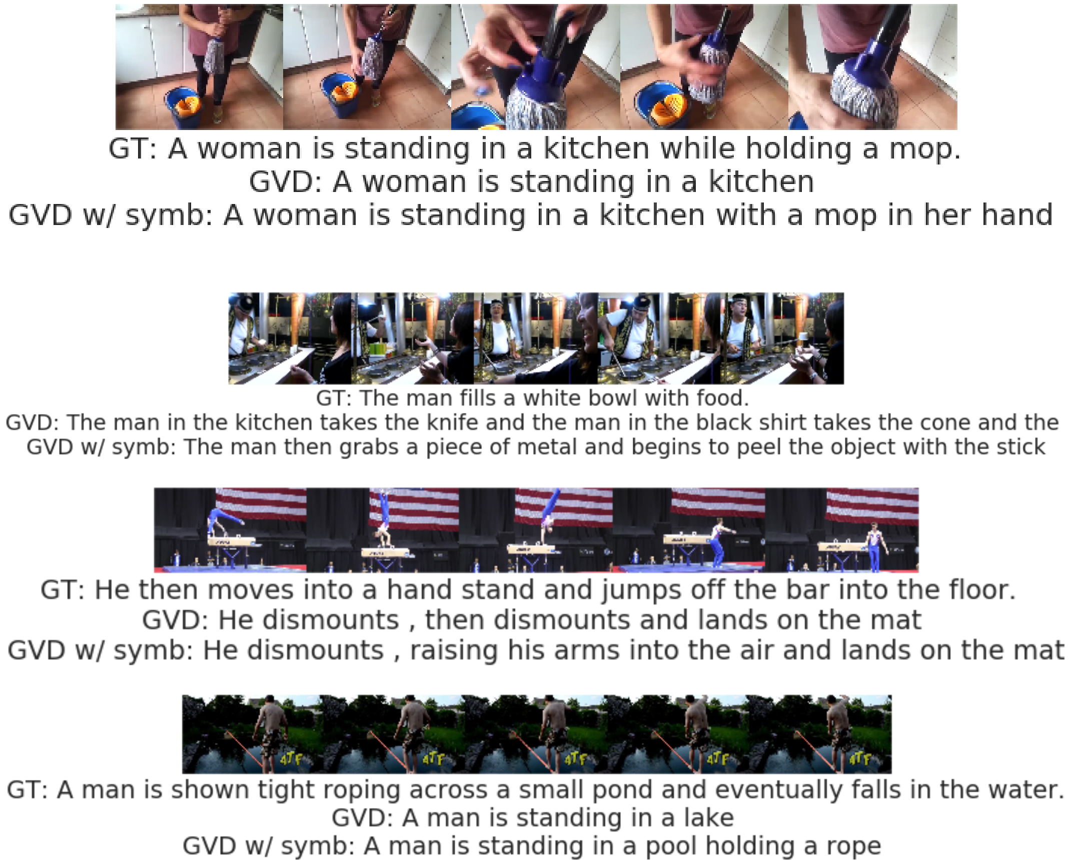
large-scale dataset under full supervision. As we will discuss later in this chapter, when bounding box annotations are not available during training, the performance of both self-attention and our V-HetGAt model degrade, but our model seems to be more robust.

3.3.5.5 Qualitative Results

Figure 3.5 illustrates video captioning results on sample video segments from the ActivityNet Entities validation set. Augmenting the self-attention module of GVD [4], with our Semantic Context Module seems to lead to richer, better grounded captions, capturing more details about the objects in the frames. For example, as we can see in the first example, our AO-GVD describes that the woman has a mop in her hand, instead of just describing that she is in a kitchen, as GVD does. Similarly, it refers to the rope in the last example, although it mistakenly describes the man as holding the rope instead of walking on it.

3.3.6 Conclusion

In the first half of this chapter, we have introduced an extension of the fully-supervised GVD model originally proposed by Zhou et al. [4] for Grounded Visual Description and Visual Object Grounding. Our proposed AO-GVD model utilizes region embeddings that are aware of local spatio-temporal actor-object interactions and global semantic interactions. To achieve this, we have adopted the heterogeneous message passing modules from Chapter 2. Experimental results on ActivityNet Entities have shown that our Visual Context Module (which explicitly takes into account node and edge types on a sparse graph) performs comparably with powerful multi-head attention mechanisms (which operate on a fully-connected, homogeneous region graph) for grounding words in ground-truth or generated sentences. Combining multi-head attention with our Semantic Context Module outperformed the original GVD model in Grounded Visual Description.



GT: A woman is standing in a kitchen while holding a mop.
GVD: A woman is standing in a kitchen
GVD w/ symb: A woman is standing in a kitchen with a mop in her hand

GT: The man fills a white bowl with food.
GVD: The man in the kitchen takes the knife and the man in the black shirt takes the cone and the
GVD w/ symb: The man then grabs a piece of metal and begins to peel the object with the stick

GT: He then moves into a hand stand and jumps off the bar into the floor.
GVD: He dismounts , then dismounts and lands on the mat
GVD w/ symb: He dismounts , raising his arms into the air and lands on the mat

GT: A man is shown tight roping across a small pond and eventually falls in the water.
GVD: A man is standing in a lake
GVD w/ symb: A man is standing in a pool holding a rope

Figure 3.5: Video captioning results on sample video segments from the ActivityNet Entities validation set. Augmenting MHA with our Semantic Context Module module (SCM-VG) seems to yield richer captions, capturing more details about the objects in the video, even when there are mistakes in the described events. *GT*: Ground Truth. *GVD*: Grounded Video Description model with MHA. *GVD w/ symb (ours)*: GVD with SCM (5 frames shown from each segment). Best viewed zoomed in and in color.

3.4 Weakly-Supervised Generation and Grounding of Visual Descriptions with Conditional Generative Models

In Section 3.3, we proposed a visual grounding system that requires bounding box annotations for each groundable word during training. However, constructing datasets with such fine-grained bounding box annotations is rather time-consuming and costly. In this section, we propose a weakly-supervised visual grounding system that requires only video-caption pairs for training. In particular, we consider two tasks, as illustrated in Figure 3.6: (1) *Weakly-Supervised Visual Object Grounding (WS-VOG)*, where given an input image (or video) and its visual description, the goal is to *localize* the referred semantic entities in the visual input, and (2) *Weakly-Supervised Grounded Visual Description (WS-GVD)*, where given an input image (or video), we must jointly *generate* a natural language description and *localize* the generated words.

Most prior work has focused on learning how to align words with regions by learning how to correctly match images and videos to sentences [83, 119, 90, 9]. However, these matching-based approaches can only tackle the first task (*WS-VOG*), and cannot generate grounded visual descriptions. On the other hand, captioning-based approaches [4, 197] learn to ground words by learning to generate captions based on region proposals, thus they can tackle both tasks. For example, the GVD model [4] we discussed in Section 3.3.1 can be trained only with the captioning loss term and then the region attention coefficients can be used during inference for grounding each word. Nonetheless, exploiting soft attention as a grounding mechanism suffers from two major

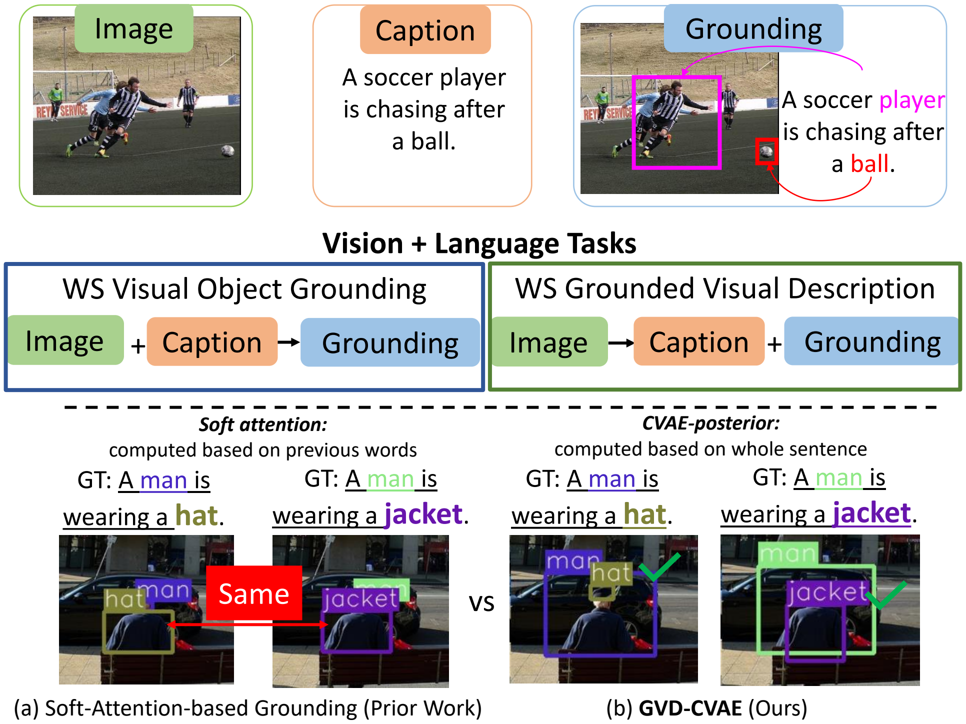


Image
Caption
A soccer player is chasing after a ball.
Grounding
A soccer player is chasing after a ball.
Vision + Language Tasks
WS Visual Object Grounding
Image + Caption → Grounding
WS Grounded Visual Description
Image → Caption + Grounding
Soft attention:
computed based on previous words
CVAE-posterior:
computed based on whole sentence
GT: A man is wearing a hat.
GT: A man is wearing a jacket.
GT: A man is wearing a hat.
GT: A man is wearing a jacket.
hat
man
Same
jacket
vs
man
hat
man
jacket
(a) Soft-Attention-based Grounding (Prior Work)
(b) GVD-CVAE (Ours)

Figure 3.6: Our proposed framework jointly models visual descriptions and word-to-region alignments conditioned on an input image (or video) and region proposals. Without using any bounding box annotations during training, it can tackle two tasks: Visual Object Grounding and Grounded Visual Description. Unlike prior work [4] that leverages soft attention for grounding and always predicts the same region for two words given the same visual input and partial caption context, our model can ground words by taking into account the full ground-truth or generated sentence.

limitations. First, despite being an effective, end-to-end learnable mechanism for summarizing relevant context, attention is not encouraged in any way to capture meaningful alignments and can result in poor grounding [76], unless it is supervised, as in Section 3.3. More importantly, each word is generated using attention computed from a query that summarizes the previously generated words (partial caption). Hence, the attention does not take into account the word to be grounded. For example, consider grounding the words *'hat'* and *'jacket'* given the sentences *"A man is wearing a hat"* and *"A man is wearing a jacket"*, respectively. As shown in Figure 3.6, existing attention-based grounding approaches will wrongly predict the same box for 'hat' and 'jacket', since the partial caption is the same.

To overcome these limitations, we propose a conditional generative model for the joint probability distribution of sentences and latent word-to-region alignments given an input image (or video) and a set of region proposals. That is, we account for the lack of ground-truth grounding annotations by introducing discrete latent variables that model word-to-region alignments. We parameterize our model with state-of-the-art visual encoders, language decoders and attention modules, and leverage Amortized Variational Inference [240, 241] to learn their parameters. The resulting Grounded Visual Description Conditional Variational Autoencoder (GVD-CVAE) allows us to both generate sentences, and also infer the latent word-to-region alignments by taking into account the *whole sentence, including the word to be grounded*. Hence, it can correctly ground the *hat* in the motivating example.

In summary, the contributions of the second part of this chapter are threefold. First, we introduce the GVD-CVAE, a novel conditional generative model of visual descriptions with a sequential discrete latent space and attention-based parameterization of the prior and approximate posterior alignment distributions. Second, we propose a training objective that encourages our CVAE model to learn latent variables that capture meaningful word-to-region alignments. Finally, we evaluate our method on three challenging image and video datasets and demonstrate that both our "prior" and "approximate

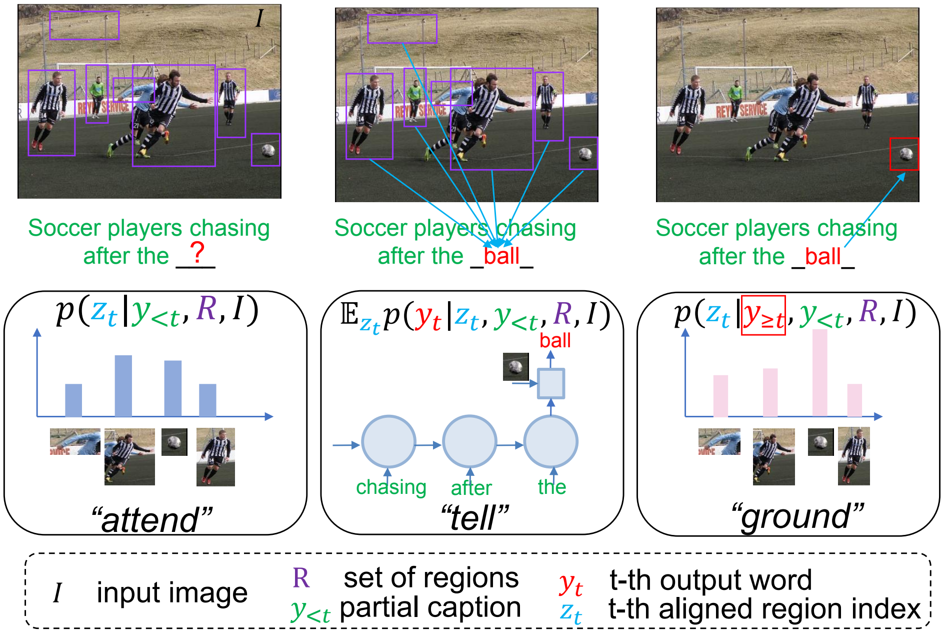

I
Soccer players chasing after the _?_
Soccer players chasing after the _ball_
Soccer players chasing after the _ball_
$p(z_t|y_{<t}, R, I)$
$\mathbb{E}_{z_t} p(y_t|z_t, y_{<t}, R, I)$
$p(z_t|\boxed{y_{\geq t}}, y_{<t}, R, I)$
ball
chasing
after
the
"attend"
"tell"
"ground"
I input image
R set of regions
$y_{<t}$ partial caption
y_t t-th output word
z_t t-th aligned region index

Figure 3.7: We propose a deep conditional generative model of visual descriptions that models each word-to-region alignment with a discrete latent variable $\mathbf{z}_t$. It is able to *attend* over the region proposals in an input image (or video), *tell* what it shows by marginalizing out the latent word-to-region alignments from the joint distribution and *ground* each word by leveraging the learned approximate posterior word-to-region alignment distribution.

posterior" alignment distributions improve upon soft attention. This leads to a 12% absolute improvement in Visual Object Grounding on Flickr30k Entities. Our model also achieves state-of-the-art or competitive grounding and captioning performance compared with a diverse family of state-of-the-art methods that are tailored to WS-VOG or WS-GVD.

3.4.1 GVD-CVAE: Attention-based Conditional Variational Autoencoder

As we described in Section 3.1 of this chapter, we are interested in grounding object words of ground-truth or generated visual descriptions in an input video (or image), and we treat this problem as a problem of word-to-region alignment by leveraging M candidate region proposals $R = \{\mathbf{r}_m\}_{m=1}^{M}$ extracted by off-the-shelf object detectors. Then, the localization problem is reduced to identifying the variable $\mathbf{z}_t \in \{0,1\}^M$, which denotes which region corresponds to the t-th word. However, we are now interested in training models using only weak supervision in the form of aligned pairs of images (or videos) and visual descriptions: $\{(I^{(n)}, Y^{(n)})\}_{n=1}^{N}$. Our key idea is to model the word-to-region alignments as latent variables in a deep conditional generative model. To this end, we propose a novel Grounded Visual Description Conditional Variational Autoencoder (GVD-CVAE). As illustrated in Figure 3.7, learning such a model allows us to leverage the posterior distribution of word-to-region alignments for grounding words *based on the entire sentence*, unlike attention-based grounding.

Let $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$ be the sequence of latent variables corresponding to alignments between words and regions, where $\mathbf{z}_t \in \{0,1\}^M$ is a binary discrete random variable with $z_{t,i} = 1$ when the i-th region proposal corresponds to the t-th word $\mathbf{y}_t$ and $z_{t,i} = 0$ otherwise. The joint conditional distribution of a caption Y and sequence of alignments Z, given the input video (or image) I and candidate regions R, can be factorized by using the chain rule on both Y and Z as follows:

$$\begin{aligned} p_\theta(Y, Z \mid R, I) &= p_\theta(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T, \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T \mid R, I) \\ &= \prod_{t=1}^{T} p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_{\leq t}, R, I) p_\theta(\mathbf{z}_t \mid \mathbf{y}_{<t}, \mathbf{z}_{<t}, R, I), \end{aligned} \tag{3.29}$$

where $\mathbf{y}_{<t} = [\mathbf{y}_1, \dots, \mathbf{y}_{t-1}]$ denotes the partial caption up to word $t-1$, and

similarly $\mathbf{z}_{<t} = [\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}]$ denotes the sequence of word-to-region alignments up until word $t-1$. We then simplify this joint distribution by making two assumptions: (1) The t-th word depends only on the region $\mathbf{z}_t$ given the partial caption $\mathbf{y}_{<t}$, and (2) the region-to-word alignments $\mathbf{z}_t$ for each word are conditionally independent of each other given the partial caption. With these assumptions, our joint probability distribution $p_\theta(Y, Z \mid R, I)$ becomes:

$$p_\theta(Y, Z \mid R, I) = \prod_{t=1}^{T} \overbrace{p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_t, R, I)}^{\text{language decoder}} \overbrace{p_\theta(\mathbf{z}_t \mid \mathbf{y}_{<t}, R, I)}^{\text{region prior}} . \tag{3.30}$$

Note that when modeling the first word $t = 1$, we assume that the previous word is a special $[BOS]$ token. Next, we describe how we parameterize our conditional generative model with deep networks, whose trainable weights are denoted with θ, as illustrated in Figure 3.8.

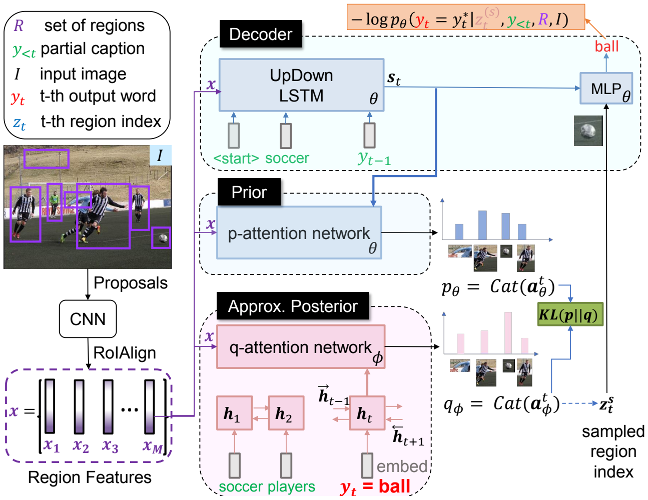


Figure 3.8: Our proposed GVD-CVAE architecture. The input image and proposals are fed through a *visual encoder* to produce region embeddings. The *prior word-to-region alignment* is computed as a function of only the previous words, while the *approximate posterior* is computed as a function of the full sentence. During training, a region is sampled from the approximate posterior and is fed to the *language decoder* that predicts the next word.

3.4.1.1 Visual Encoder

We use the same visual encoder as the one we described in Section 3.3.1. Recall that the visual encoder consists of pretrained backbone holistic models and trainable linear projections. Each video is encoded to a global video feature $\mathbf{v}$, a sequence of frame-level features $F = \{\mathbf{f}_l\}_{l=1}^{L}$, where l indexes the frames, and grounding-aware region representations $X = \{\mathbf{x}_m\}_{m=1}^{M}$. If the input is an image instead of a video, we can similarly describe it with a coarse image-level feature vector $\mathbf{v}$, fine-grained frame-level features $F = \{\mathbf{f}_l\}_{l=1}^{L}$, where l indexes the feature map spatial grid, and grounding-aware region representations [4], $X = \{\mathbf{x}_m\}_{m=1}^{M}$, that encode information about appearance, spatial position and object class knowledge transferred from an external dataset.

3.4.1.2 Language Decoder

The decoder $p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_t, R, I) = \mathrm{Cat}(g_\theta(\mathbf{s}_t, \mathbf{z}_t, X))$ is a categorical distribution over words on the vocabulary given the partial caption $\mathbf{y}_{<t}$, the word-to-region alignment $\mathbf{z}_t$, the region proposals R, and the visual input I. We parameterize this distribution with a Multilayer Perceptron (MLP) network:

$$g_\theta(\mathbf{s}_t, \mathbf{z}_t, X) = \mathrm{softmax}\left(W_c \tanh\left(W_h \left[\sum_{i=1}^{M} z_{t,i}\mathbf{x}_i; \mathbf{s}_t\right]\right)\right), \tag{3.31}$$

whose inputs are: (a) the state $\mathbf{s}_t \in \mathbb{R}^d$ of a trainable language model that summarizes $\mathbf{y}_{<t}$, R and I, and (b) the aligned region feature $\sum_{i=1}^{M} z_{t,i}\mathbf{x}_i \in \mathbb{R}^d$. Here, $[\cdot;\cdot]$ denotes concatenation and $W_h \in \mathbb{R}^{d\times 2d}, W_c \in \mathbb{R}^{|\mathcal{V}|\times d}$ are learnable weights.

Although $\mathbf{s}_t$ can be chosen to be the state of any standard language model [5, 239], we follow all prior work on grounded visual description [4, 197, 199, 198] and adopt a variant of the UpDown [5] LSTM model. As we discussed in detail earlier in this chapter in Sec. 3.3.1, this language model is composed of a word embedding layer (emb) and two LSTM [233] layers with hidden states $\mathbf{u}_t$ (Eq. 3.12) and $\mathbf{s}_t$ (Eq. 3.13), respectively. It also uses attention mechanisms $f_\theta(\cdot,\cdot)$ (defined in Eq. 3.14) and $k_\theta(\cdot,\cdot)$ (Eq. 3.15) over holistic visual features F and region features X respectively. Another simpler alternative that we explored uses the state $\mathbf{s}_t$ of a vanilla LSTM, which does not use the region features X:

$$\mathbf{s}_t = \mathrm{RNN}_\theta\left(\mathbf{s}_{t-1}, [\mathbf{v}; \mathrm{emb}(\mathbf{y}_{t-1})]\right). \tag{3.32}$$

3.4.1.3 Prior Word-To-Region Alignment Distribution

The prior distribution $p_\theta(\mathbf{z}_t \mid \mathbf{y}_{<t}, R, I) = \mathrm{Cat}(\alpha_\theta(\mathbf{s}_t, X))$ is a categorical distribution over possible word-to-region alignments. We choose to parameterize it with an additive attention mechanism [242] that uses the concatenation of the hidden state $\mathbf{s}_t$ of the UpDownLSTM from Eq. 3.13) and the region feature $\mathbf{x}_i$ as a query to compute region attention coefficients $\alpha_\theta(\mathbf{s}_t, X) \in \mathbb{R}^M$ that summarize the partial caption and visual input:

$$\alpha_\theta^{(i)}(\mathbf{s}_t, X) = \frac{\mathbf{v}_{ap}^T \tanh(W_{ap}[\mathbf{s}_t; \mathbf{x}_i])}{\sum_{j=1}^M \mathbf{v}_{ap}^T \tanh(W_{ap}[\mathbf{s}_t; \mathbf{x}_j])}, \tag{3.33}$$

where $\mathbf{v}_{ap} \in \mathbb{R}^d, W_{ap} \in \mathbb{R}^{d \times 2d}$ are learnable weights of the attention mechanism and d_a is a hyperparameter.

3.4.1.4 Variational Approximate Posterior Distribution

To learn the parameters of our conditional generative model we will leverage Amortized Variational Inference (AVI). Therefore, our model becomes a CVAE [241] with sequential discrete latent space and sentences as observations. In the CVAE framework, a variational distribution $q_\phi(Z \mid Y, R, I)$ is introduced to approximate the true posterior and is parameterized via a neural network with weights ϕ, also known as the "inference network". We experiment with various choices of the approximate posterior distribution.

Smoothing approximate posterior. Here, we choose to approximate the true posterior with a smoothing approximate posterior, which conditions each word-to-region alignment on the full sentence:

$$q_\phi(Z \mid Y, R, I) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t \mid \mathbf{y}_{\leq T}, R, I). \tag{3.34}$$

Then, we model the approximate posterior distribution of each word-to-region alignment as a multinomial distribution that is parameterized by the region attention coefficients $\alpha_\phi(\mathbf{h}_t, X) \in \mathbb{R}^M$ obtained via another attention network, the "q-attention-network", which receives a query $\mathbf{h}_t \in \mathbb{R}^d$ that summarizes the whole sentence and visual input, and the region features as keys. We experimented with various instantiations of this attention network, and in particular with the additive attention mechanism of Bahdanau et al. [242] and with the general dot product attention mechanism of Luong et al. [243]. For example, when using the general dot product attention mechanism, our approximate posterior distribution takes the following form:

$$\alpha_{\phi}^{(i)}(\mathbf{h}_t, X) = \frac{\mathbf{h}_t^T U \mathbf{x}_i}{\sum_{j=1}^{M} \mathbf{h}_t^T U \mathbf{x}_j}, \tag{3.35}$$

$$\mathbf{z}_t \mid Y, R, I \sim \mathrm{Cat}(\alpha_{\phi}(\mathbf{h}_t, X)), \tag{3.36}$$

where $U \in \mathbb{R}^{d \times d}$ is a learnable weight matrix. We observed that the choice of this attention mechanism is critical for the overall framework, and the optimal choice differs among datasets and models.

The attention query $\mathbf{h}_t$, which summarizes the whole sentence, is obtained by summing the forward and backward states of a BiLSTM network, whose inputs consist of the global feature $\mathbf{v}$ and ground-truth word $\mathbf{y}_t$ at each timestep:

$$\overrightarrow{\mathbf{h}_t} = \mathrm{BiLSTM}_{\phi}(\overrightarrow{\mathbf{h}_{t-1}}, [\mathrm{emb}(\mathbf{y}_t), \mathbf{v}]) \tag{3.37}$$

$$\overleftarrow{\mathbf{h}_t} = \mathrm{BiLSTM}_{\phi}(\overleftarrow{\mathbf{h}_{t+1}}, [\mathrm{emb}(\mathbf{y}_t), \mathbf{v}]) \tag{3.38}$$

$$\mathbf{h}_t = \overrightarrow{\mathbf{h}_t} + \overleftarrow{\mathbf{h}_t}, \tag{3.39}$$

where $\mathrm{emb}(\cdot)$ is the same word embedding layer as the one used in the language decoder (and thus is the only set of shared parameters between the deep conditional generative model and the approximate posterior).

Filtering approximate posterior. An alternative is to use a filtering approximate posterior, which conditions each word-to-region alignment on the partial caption up until the word to be grounded:

$$q_{\phi}(Z \mid Y, R, I) = \prod_{t=1}^{T} q_{\phi}(\mathbf{z}_t \mid \mathbf{y}_{\leq t}, R, I). \tag{3.40}$$

The q-attention network remains unmodified in this case, but since it needs to parameterize a distribution that is conditioned on just the partial caption, it receives as query the hidden state $\mathbf{h}_t$ of a simple LSTM (instead of a BiLSTM), i.e.,

$$\mathbf{h}_t = LSTM_\phi(\mathbf{h}_{t-1}, [\mathrm{emb}(\mathbf{y}_t), \mathbf{v}]) \tag{3.41}$$

$$\alpha_\phi^{(i)}(\mathbf{h}_t, X) = \frac{\mathbf{h}_t^T U \mathbf{x}_i}{\sum_{j=1}^{M} \mathbf{h}_t^T U \mathbf{x}_j}, \tag{3.42}$$

$$\mathbf{z}_t \mid \mathbf{y}_{\leq t}, R, I \sim \mathrm{Cat}(\alpha_\phi(\mathbf{h}_t, X)), \tag{3.43}$$

where $U \in \mathbb{R}^{d \times d}$ is a learnable weight matrix.

Object-aware approximate posterior. So far, we have designed deep inference networks, consisting of a (Bi)LSTM and an attention mechanism, for computing the parameters of the categorical approximate posterior word-to-region alignment distribution for each datapoint (video-caption pair). This network treats all words of the sentence in the same way. However, as we discussed earlier, our goal is to ground the groundable words of a sentence, i.e., object words belonging in the vocabulary of groundable words $\mathcal{V}_o$. This motivates the third variant of the approximate word-to-region alignment distribution, which takes into account the compatibility of each region with each groundable object word based on transferred object detector knowledge.

Recall that each grounding-aware region embedding $\mathbf{x}_i$ computed by our visual encoder, as defined in Eq. 3.9, contains information about the region's similarity to each one of the groundable object words. As we discussed, these similarity scores $M_s(\mathbf{o}) \in \mathbb{R}^{\mathcal{V}_o}$ (Eq. 3.8) are obtained by transferring object class knowledge [4] from an off-the-shelf object detector [93] trained on an external dataset [182]. Since our approximate word-to-region alignment distribution $q_\phi(\mathbf{z}_t \mid Y, R, I)$ for each word $\mathbf{y}_t$ is conditioned on the full sentence, including the word itself, we can explicitly utilize the similarity of each region to the groundable object word when computing the parameters of our approximate posterior distribution for the groundable words in the sentence.

More specifically, let $\mathbf{y}_t$ be a groundable word from the vocabulary $\mathcal{V}_o$ and let $c_t \in [1, \ldots, |\mathcal{V}_o|]$ denote its groundable object word index ($y_{t,c_t} = 1$). Then the object-aware approximate distribution takes the following form:

$$\mathbf{z}_t \mid Y, R, I \sim \mathrm{Cat}(\mathrm{softmax}(\tilde{\alpha}_\phi(\mathbf{h}_t, X) + \mathbb{1}[(\operatorname{argmax} \mathbf{y}_t) \in \mathcal{V}_o](O^T W_s^{(c_t)} + \mathbf{b}_s^{(c_t)}))), \tag{3.44}$$

where $\tilde{\alpha}_\phi(\mathbf{h}_t, X)$ are the unnormalized attention coefficients obtained via the q-attention network, $O = [\mathbf{o}_1; \ldots; \mathbf{o}_M] \in \mathbb{R}^{d_o \times M}$ is a matrix of region object features (extracted from the *fc7* layer of the object detector and also included in the grounding-aware region embedding defined in Eq. 3.9), $W_s^{(c_t)} \in \mathbb{R}^{d_0}$ is the (c_t)-th row of the semantic embedding matrix defined in Eq. 3.8, i.e., it is a weight row vector initialized with the pretrained object classifier corresponding to the external dataset object class that is closest to the object word $\mathbf{y}_t$, and $\mathbf{b}_s^{(c_t)}$ is the scalar bias corresponding to the c_t object word.

3.4.1.5 Approximate Inference

Visual Object Grounding. Given an input image or video and a ground-truth sentence Y, we address the VOG task by inferring the latent word-to-region alignment for each word of the sentence using the approximate posterior word-to-region alignment distribution:

$$\hat{b}_t = \mathbf{r}_j, \text{ where } j = \underset{i \in \{1,\ldots,M\}}{\operatorname{argmax}} \; q_\phi(z_{t,i} = 1 \mid Y, R, I). \tag{3.45}$$

We also experimented with using the prior word-to-region alignment distribution for grounding:

$$\hat{b}_t = \mathbf{r}_j, \text{ where } j = \underset{i \in \{1,\ldots,M\}}{\operatorname{argmax}} \; p_\theta(z_{t,i} = 1 \mid \mathbf{y}_{<t}, R, I). \tag{3.46}$$

Although this distribution suffers from the same limitation as soft-attention, namely it does not take into account the word being grounded, our experimental results suggest that it outperforms soft-attention. Notice that both grounding approaches have the same computational complexity as popular grounding methods [89, 4]: they require passing the sentence through a (Bi)LSTM and applying an attention mechanism over regions for each word.

We would like to clarify that our model assumes a single latent region for each groundable word both for images and videos. Therefore, given an input video and a textual description, each groundable word is localized with a bounding box in a potentially different frame of the video. However, we would often like to ground words in particular frames of the video. To do this, we use a heuristic, i.e., we choose the region at frame l with maximum q-attention coefficient (or p-attention coefficient (Eq. 3.33)) when grounding words given only the partial caption):

$$\hat{b}_{t,l} = \mathbf{r}_j, \quad \text{where } j = \underset{i \in \mathcal{J}_l}{\operatorname{argmax}} \, a_\phi^{(i)}(\mathbf{h}_t, X), \tag{3.47}$$

where $\mathcal{J}_l$ is the set of region indices extracted from frame l and $a_\phi(\mathbf{h}_t, X)$ is computed with the q-attention network in Eq. 3.35.

Grounded Visual Description. For the task of GVD, we follow a two-stage approach: first we generate a sentence and then we ground the generated words. Similar to the attention-based encoder-decoder model that we described in the first part of this chapter, we perform greedy decoding for sentence generation, i.e. we predict a word y_t^* at each timestep t and feed it as input to the next timestep. In particular, each next word can be predicted by using the marginal word distribution:

$$\widehat{\mathbf{y}}_t = \operatorname*{argmax}_{\mathbf{y}_t} \mathop{\mathbb{E}}_{\mathbf{z}_t \sim p_\theta(z_t \mid \mathbf{y}^*_{<t}, R, I)} p_\theta(\mathbf{y}_t \mid \mathbf{z}_t, \mathbf{y}^*_{<t}, R, I). \tag{3.48}$$

The marginal word distribution can be approximated via Monte Carlo sampling:

$$\widehat{\mathbf{y}}_t = \operatorname*{argmax}_{\mathbf{y}_t} \frac{1}{K} \sum p_\theta(\mathbf{y}_t \mid \mathbf{z}_t^{(k)}, \mathbf{y}^*_{<t}, R, I), \tag{3.49}$$

where $\left\{\mathbf{z}_t^{(k)}\right\}_{k=1}^{K}$ are K samples drawn according to $p_\theta(z_t \mid \mathbf{y}^*_{<t}, R, I)$. However, doing so is computationally expensive. Instead, we feed the expected value of $\mathbf{z}_t$:

$$\widehat{\mathbf{y}}_t = \operatorname*{argmax}_{\mathbf{y}} p_\theta(\mathbf{y}_t \mid \mathop{\mathbb{E}}_{\mathbf{z}_t \sim p_\theta} [\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I). \tag{3.50}$$

Observe that $\mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t] = a_\theta(\mathbf{s}_t, X) \in \mathbb{R}^M$, i.e. the expected value of $\mathbf{z}_t$ is equal to the attention coefficients computed by the p-attention network.

Note that $p_\theta(\mathbf{y}_t \mid \mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I)$ is a first-order Taylor approximation of the expectation in Eq. 3.48. If we choose to use a single layer MLP for word prediction (Eq. 3.31), then $p_\theta(\mathbf{y}_t \mid \mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I)$ is also valid lower

bound of $\mathbb{E}_{\mathbf{z}_t}\, p_\theta(\mathbf{y}_t \mid \mathbf{z}_t, \mathbf{y}^*_{<t}, R, I)$, since the single layer MLP is a convex function (composed of a linear mapping and the softmax function) of $\mathbf{z}_t$:

$$g_\theta(\mathbf{s}_t, \mathbf{z}_t, X) = \text{softmax}\left(W_c\left[\sum_{i=1}^{M} z_{t,i}\mathbf{x}_i; \mathbf{s}_t\right]\right), \tag{3.51}$$

$$\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_t, R, I \sim \text{Cat}(g_\theta(\mathbf{s}_t, \mathbf{z}_t, X)). \tag{3.52}$$

Using the expected value of $\mathbf{z}_t$ serves as a shortcut to avoid sampling, thus retaining the same computational complexity as discriminative encoder-decoder captioning methods:

$$\widehat{\mathbf{y}}_t = \text{argmax}\left[\text{softmax}\left(W_c\left[\sum_{i=1}^{M} a_\theta^{(i)}(\mathbf{s}_t, X)\mathbf{x}_i; \mathbf{s}_t\right]\right)\right]. \tag{3.53}$$

Given the generated sentence $\hat{Y}$, we can use the prior (Eq. 3.46) or approximate posterior (Eq. 3.45) word-to-region alignment distributions to ground the generated words.

3.4.1.6 Training

During training, we are given a dataset consisting of N i.i.d. pairs of images (or videos) and their visual descriptions, without grounding supervision. To train our GVD-CVAE, we minimize the following hybrid training objective over the parameters θ and ϕ (omitting the conditioning of all distributions on the visual input $I^{(n)}$ for readability):

$$\mathcal{L} = \frac{1}{N}\sum_{n,t} \lambda\mathcal{L}_{CVAE}(n,t) + (1-\lambda)\mathcal{L}_{CE}(n,t), \tag{3.54}$$

where $\mathcal{L}_{CE}(n,t) = -\log p_\theta(\mathbf{y}_t^{(n)} \mid \mathop{\mathbb{E}}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}_{<t}^{(n)}, R^{(n)})$ and

$$\mathcal{L}_{CVAE}(n,t) = \overbrace{\mathop{\mathbb{E}}_{\mathbf{z}_t \sim q_\phi}\left[-\log p_\theta(\mathbf{y}_t^{(n)} \mid \mathbf{y}_{<t}^{(n)}, \mathbf{z}_t, R^{(n)})\right]}^{\text{reconstruction loss}}$$
$$+ \beta \mathrm{KL}\left(q_\phi(\mathbf{z}_t \mid Y^{(n)}, R^{(n)}) \mid\mid p_\theta(\mathbf{z}_t \mid \mathbf{y}_{<t}^{(n)}, R^{(n)})\right). \quad (3.55)$$

Here, $KL(q(x) \mid\mid p(x))$ denotes the Kullback-Leibler divergence between two probability distributions $q(x)$ and $p(x)$ defined on the same probability space $\mathcal{X}$, and it is a non-symmetric measure of the difference between the two probability distributions. In particular it measures the information lost when $p(x)$ is used to approximate $q(x)$:

$$KL(q \mid\mid p) = \mathop{\mathbb{E}}_{x \sim q}\left[\ln\left(\frac{q(x)}{p(x)}\right)\right]. \quad (3.56)$$

Let us now dive into each one of the terms of this hybrid training objective.

Evidence Lower Bound Loss Term. To fit the parameters of the θ of the latent-variable model $p_\theta(Y, Z \mid R, I)$, we would ideally like to maximize the conditional log-likelihood of training examples $(Y^{(n)}, R^{(n)}, I^{(n)})$:

$$\max_\theta \log p_\theta(Y \mid R, I) = \max_\theta \log \mathop{\mathbb{E}}_Z \log p_\theta(Y, Z \mid R, I). \quad (3.57)$$

Based on our factorization of the joint probability distribution, this takes the form:

$$\max_\theta \ \log \ \mathop{\mathbb{E}}_Z \prod_{t=1}^{T} p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, R, I) p_\theta(\mathbf{z}_t \mid \mathbf{y}_{<t}, R, I). \quad (3.58)$$

In the case of conditionally independent, categorical prior word-to-region alignments, this is tractable, but we need to compute M softmaxes per word in the vocabulary. Depending on M and $|\mathcal{V}|$ (long video/large number of region proposals and/or large vocabulary), the cost can be prohibitive.

Instead, our method builds upon the framework of Conditional Variational Autoencoders. Hence, we approximate the true posterior with another simpler distribution $q_\phi(Z \mid Y, R, I) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t \mid Y, R, I)$ from a family of approximate distributions Q.

Let (Y, R, I) be an image-regions-caption triplet in training set. We want q_ϕ to be as "close" as possible to the true posterior $p(Z \mid Y, R, I)$:

$$KL(q(Z \mid Y, R, I) \mid\mid p(Z \mid Y, R, I)) = \mathop{\mathbb{E}}_{Z \sim q}\left[\log \frac{q(Z \mid Y, R, I)}{p(Z \mid Y, R, I)}\right] \tag{3.59}$$

$$= \mathop{\mathbb{E}}_{Z \sim q}[\log q(Z \mid Y, R, I)] - \mathop{\mathbb{E}}_{Z \sim q}[\log p(Z, Y \mid R, I)] + \log p(Y \mid R, I). \tag{3.60}$$

Given that the KL-divergence is non-negative, we get:

$$\log p(Y \mid R, I) \geq \underbrace{(\mathop{\mathbb{E}}_{Z \sim q}[\log p(Z, Y \mid R, I)] - \mathop{\mathbb{E}}_{Z \sim q}[\log q(Z \mid Y, R, I)])}_{ELBO} \tag{3.61}$$

This shows that the term on the right, which is called the Evidence Lower Bound (ELBO), is a lower bound on the log evidence, with the equality holding when the approximate posterior is equal to the true posterior. We therefore maximize the ELBO instead of maximizing the log-likelihood ($\log p(Y \mid R, I)$). Plugging in our factorization of the probability distribution (Eq. 3.30) and, for example our smoothing approximate posterior (Eq. 3.34), in the ELBO yields:

$$ELBO = \mathop{\mathbb{E}}_{Z\sim q}[\log p(Z,Y|R,I)] - \mathop{\mathbb{E}}_{Z\sim q}[\log q(Z \mid Y,R,I)] \tag{3.62}$$

$$= \mathop{\mathbb{E}}_{Z\sim q}\big[\log \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{z}_t,R,I)p(\mathbf{z}_t|\mathbf{y}_{<t},R,I)\big] - \mathop{\mathbb{E}}_{Z\sim q}\big[\log \prod_{t=1}^{T} q(\mathbf{z}_t|Y,R,I)\big] \tag{3.63}$$

$$= \mathop{\mathbb{E}}_{Z\sim q}\big[\sum_{t=1}^{T} \log p(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{z}_t,R,I)\big] - \mathop{\mathbb{E}}_{Z\sim q}\big[\sum_{t=1}^{T} \log \frac{q(\mathbf{z}_t \mid Y,R,I)}{p(\mathbf{z}_t \mid \mathbf{y}_{<t},R,I)}\big] \tag{3.64}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{z}_t\sim q_\phi(\mathbf{z}_t|Y,R,I)} \log p(\mathbf{y}_t|\mathbf{y}_{<t},\mathbf{z}_t,R,I) - \mathrm{KL}(q(\mathbf{z}_t|Y,R,I)||p(\mathbf{z}_t|\mathbf{y}_{<t},R,I)) \tag{3.65}$$

This quantity, which we would like to maximize, is exactly the negative of our $\mathcal{L}_{CVAE}$ loss term (Eq. 3.55 with $\beta = 1$), which we minimize. To derive Eq. 3.65 from Eq. 3.64, we used the following equalities:

$$\mathop{\mathbb{E}}_{Z\sim q_\phi}\left[\sum_{t=1}^{T} \log p(\mathbf{y}_t \mid \mathbf{y}_{<t},\mathbf{z}_t,R,I)\right] = \tag{3.66}$$

$$= \mathop{\mathbb{E}}_{\mathbf{z}_1,\ldots,\mathbf{z}_T\sim q_\phi(Z|Y,R,I)}\left[\sum_{t=1}^{T} \log p(\mathbf{y}_t \mid \mathbf{y}_{<t},\mathbf{z}_t,R,I)\right] \tag{3.67}$$

$$= \mathop{\mathbb{E}}_{\mathbf{z}_1\sim q_\phi(\mathbf{z}_1|Y,R,I)}\left[\cdots \mathop{\mathbb{E}}_{\mathbf{z}_T\sim q_\phi(\mathbf{z}_T|Y,R,I)}\left[\sum_{t=1}^{T} \log p(\mathbf{y}_t \mid \mathbf{y}_{<t},\mathbf{z}_t,R,I)\right]\right] \tag{3.68}$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{z}_t\sim q_\phi(\mathbf{z}_t|Y,R,I)}[\log p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t},\mathbf{z}_t,R,I)] \tag{3.69}$$

$$\mathop{\mathbb{E}}_{Z\sim q_\phi}\left[\sum_{t=1}^{T}\log\frac{q(\mathbf{z}_t \mid Y,R,I)}{p(\mathbf{z}_t \mid \mathbf{y}_{<t},R,I)}\right] \tag{3.70}$$

$$= \mathop{\mathbb{E}}_{\mathbf{z}_1,\ldots,\mathbf{z}_T\sim q_\phi(Z|Y,R,I)}\left[\sum_{t=1}^{T}\log\frac{q(\mathbf{z}_t \mid Y,R,I)}{p(\mathbf{z}_t \mid \mathbf{y}_{<t},R,I)}\right] \tag{3.71}$$

$$= \mathop{\mathbb{E}}_{\mathbf{z}_1\sim q_\phi(\mathbf{z}_1|Y,R,I)}[\cdots \mathop{\mathbb{E}}_{\mathbf{z}_T\sim q_\phi(\mathbf{z}_T|Y,R,I)}[\sum_{t=1}^{T}\log\frac{q(\mathbf{z}_t \mid Y,R,I)}{p(\mathbf{z}_t \mid \mathbf{y}_{<t},R,I)}]] \tag{3.72}$$

$$= \sum_{t=1}^{T}\mathop{\mathbb{E}}_{\mathbf{z}_t\sim q_\phi(\mathbf{z}_t|Y,R,I)}\left[\log\frac{q(\mathbf{z}_t \mid Y,R,I)}{p(\mathbf{z}_t \mid \mathbf{y}_{<t},R,I)}\right] \tag{3.73}$$

$$= \sum_{t=1}^{T}\mathrm{KL}(q(\mathbf{z}_t \mid Y,R,I)||p(\mathbf{z}_t \mid Y,R,I)) \tag{3.74}$$

Reweighting the KL-divergence Loss Term. Similar to prior work in generative modeling [226], we observe that optimizing the ELBO often results in an inference model that produces posteriors almost identical to the prior, i.e., the KL loss term goes to 0. In our case, this translates to word-to-region alignments that do not take into account the word to be grounded, which was the primary motivation for developing the GVD-CVAE. It can also be shown that when the KL term vanishes, the mutual information between the latent variables (latent region alignment) and our target (word), goes to zero as well (using ideas from [244]).

To mitigate this issue, we re-weight the KL loss term with a non-negative scalar factor β. We found that gradually increasing β with slope 0.5 up to a value $\beta_{clip} < 1$ (*clipped linear annealing* schedule) during training is effective for mitigating posterior collapse in most cases, as long as the right value for the hyperparameter β_{clip} is found. (Note that this is not a variation of the β-VAE [245], where β is introduced to disentangle the latent factors and is greater than 1). However, this simple β scheduler might fail for very expressive variants of our GVD-CVAE framework, such as combining the UpDown LSTM language decoder with a BiLSTM-based inference network.

An alternative approach is proposed by Shao et al. [223], who introduce a PI-controller [223] to anneal β, so that a desired KL-divergence value v_{kl} is reached. Their PI-Controller samples the KL-divergence value $\hat{v}_{kl}$ at each training step i and compares it with the desired KL-divergence value v_{kl}. The key idea is to use the error $e(i) = v_{kl}^{(i)} - \hat{v}_{kl}^{(i)}$ as feedback to a PI-Controller that tunes $\beta(i)$ in order to reduce the error. For example, when the error is large and positive, we would like the KL-diverenge to get larger, so we should use a smaller β. In particular, $\beta(i)$ is computed as follows [223]:

$$\beta(i) = \frac{K_p}{1 + \exp(e(i))} - K_i \sum_{j=0}^{i} e(j) + \beta_{min}, \tag{3.75}$$

where K_p, K_i, β_{min} are hyperparameters. The first term changes with the error $e(i)$ (P), while the second term changes with the integral of error (I). We would like to emphasize that this is not the standard instantiation of a PI-Controller and refer the reader to ControlVAE [223] for a more detailed explanation of the chosen instantiation. We found this approach to be the most effective for training all variants of our GVD-CVAE. To determine a range for the desired KL-divergence value v_{kl}, which is a hyperparameter of our GVD-CVAE when the PI-controller is used to anneal β, we recommend training a simple GVD-CVAE (e.g. with a simple LSTM language decoder (Eq. 3.32) and filtering approximate posterior (Eq. 3.40)) with the *clipped linear annealing beta* schedule, and after a model is trained successfully, the KL-divergence value at the end of training can be a good starting point for the hyperparameter search for v_{kl} when training GVD-CVAE models with the PI-Controller.

Cross-entropy Word Prediction Loss Term. Inspired by the original CVAE framework [241], which was trained with a hybrid loss, combining the ELBO with a prediction loss using samples from the prior distribution, we also optimize a hybrid loss that consists of the ELBO (sentence reconstruction and KL divergence terms) as well as a teacher-forcing sentence prediction objective. Instead of drawing samples from the prior distribution for the latter, we simply use its expected value, i.e, the p-attention coefficients. Recall that this is exactly the approximation that we employ to generate each word given the partially generated caption during inference. Thus, this loss term makes the prediction pipelines at training and testing consistent.

$$\mathcal{L}_{CE}(n,t) = -\log p_\theta(\mathbf{y}_t^{(n)} \mid \mathop{\mathbb{E}}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}_{<t}^{(n)}, R^{(n)}). \tag{3.76}$$

Our experiments demonstrate that jointly optimizing the CVAE loss (ELBO) and the cross-entropy word prediction loss ($\lambda = 0.5$) that is applied on word predictions obtained based on the p-attention-based weighted sum of region features, is essential for training our GVD-CVAE.

Optimization. To optimize the final hybrid objective (Eq. 3.54) using Stochastic Gradient Descent, we approximate the reconstruction loss term of the CVAE loss (Eq. 3.55) with a Monte-Carlo estimator, with S region samples $\mathbf{z}_t^{(s)}$ drawn from $q_\phi(\mathbf{z}_t \mid Y, R, I)$, where S is a hyperparameter:

$$\mathop{\mathbb{E}}_{\mathbf{z}_t \sim q_\phi} \left[-\log p_\theta(\mathbf{y}_t^{(n)} | \mathbf{y}_{<t}^{(n)}, \mathbf{z}_t, R^{(n)}) \right] \approx -\frac{1}{S} \sum_{s=1}^{S} \log p_\theta(\mathbf{y}_t^{(n)} | \mathbf{y}_{<t}^{(n)}, \mathbf{z}_t^{(s)}, R^{(n)}). \quad (3.77)$$

Although it is straight-forward to sample from the categorical distribution, we cannot use the reparameterization trick [240] to backpropagate gradients from the decoder loss all the way to the parameters ϕ of the inference model through the sampling, as our latent variables are discrete. A recently popular approach to handle this difficulty is to sample from a continuous approximation of the discrete categorical distribution. In particular, we sample from the Gumbel-Softmax [246, 247] distribution:

$$\mathbf{z}_t^{(s)} = \mathrm{softmax}((a_\phi(\mathbf{h}_t, X) + \mathbf{g})/\tau), \quad (3.78)$$

where $\mathbf{g} \in \mathbb{R}^M$ is a vector of i.i.d. samples drawn from a Gumbel$(0,1)$ distribution, i.e., $\mathbf{g} = -\log(\log(\mathbf{u})), \mathbf{u} \sim \text{Uniform}(0,1)$ and τ is a temperature hyperparameter that controls the smoothness of the samples. Samples from this distribution converge to one-hot samples from the categorical distribution when $\tau \to 0$.

3.4.2 GVD-MCVAE: Sequential Grounding

In our proposed GVD-CVAE model, we assumed that the word-to-region alignments for each word in a sentence are conditionally independent of each other given the partial caption, the visual input and region proposals. However, not modeling the dependence between the grounding of different words in a sentence could result in all words grounded to the same region.

In this subsection, we extend our GVD-CVAE approach to model the dependency between the grounding of words in a sentence. We assume that (a) the t-th word depends only on the region $\mathbf{z}_t$ given the partial caption $\mathbf{y}_{1:t-1}$, and (b) the region-to-word alignment $\mathbf{z}_t$ for the t-th word depends only on the grounding $\mathbf{z}_{t-1}$ of the previous word (Markov assumption). Under these assumptions, our joint probability distribution $p_\theta(Y, Z \mid R, I)$ becomes:

$$p_\theta(Y, Z \mid R, I) = \prod_{t=1}^{T} \overbrace{p_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_t, R, I)}^{\text{language decoder}} \overbrace{p_\theta(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{y}_{<t}, R, I)}^{\text{region prior}} \tag{3.79}$$

Note that the only difference with the conditional probability distribution of the GVD-CVAE (Eq. 3.30), is in the prior word-to-region alignment distribution, which is now additionally conditioned on the previous word-to-region alignment. We will now describe how we modify the parameterization of the prior distribution and our choice of the approximate posterior distribution for this model, which we call GVD-MCVAE (Grounded Visual Description - Markov Conditional Variational Autoencoder).

The prior distribution $p_\theta(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{y}_{<t}, R, I)$ is a categorical distribution over possible word-to-region alignments. We parameterize it with an attention mechanism that yields region attention coefficients $\alpha_\theta([\mathbf{s}_t; \sum_i z_{t-1,i}\mathbf{x}_i], \mathbf{x}) \in \mathbb{R}^M$ based on a query $[\mathbf{s}_t, \sum_i z_{t-1,i}\mathbf{x}_i]$ that summarizes the partial caption, visual input and previously aligned region:

$$\alpha_\theta^{(i)}([\mathbf{s}_t; \sum_i z_{t-1,i}\mathbf{x}_i], \mathbf{x}) \propto \mathbf{w}^T \tanh(W_a[\mathbf{s}_t; \sum_i z_{t-1,i}\mathbf{x}_i; \mathbf{x}_i]), \tag{3.80}$$

$$\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{y}_{<t}, R, I \sim \mathrm{Cat}(\alpha_\theta(\mathbf{s}_t, \sum_i z_{t-1,i}\mathbf{x}_i, \mathbf{x})), \tag{3.81}$$

where $\mathbf{w}, W_a$ are learnable weights. When modeling the first word of each sentence $t = 1$, the query consists only of the state $\mathbf{s}_t$ of the language model. In practice, we add a dummy region to the set of region proposals, with its region embedding set to a zero vector, and we assume $z_{0,0} = 1$.

Similarly, we modify the design of our approximate variational posterior distribution by employing an attention mechanism that at each timestep takes into account the previous word to region alignment $\mathbf{z}_{t-1}$. For example, for the case of a filtering approximate posterior distribution, we have:

$$q_\phi(Z \mid Y, R, I) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t \mid \mathbf{y}_{\leq t}, \mathbf{z}_{t-1}, R, I), \tag{3.82}$$

where $q_\phi(\mathbf{z}_t \mid \mathbf{y}_{\leq t}, \mathbf{z}_{t-1}, R, I)$ is a categorical distribution that is parameterized by the attention coefficients $\alpha_\phi([\mathbf{h}_t; \sum_j z_{t-1,j}\mathbf{x}_j], \mathbf{x}) \in \mathbb{R}^M$:

$$\mathbf{z}_t \mid \mathbf{z}_{t-1}, Y, R, I \sim \text{Cat}(\alpha_\phi([\mathbf{h}_t; \sum_j z_{t-1,j}\mathbf{x}_j], \mathbf{x})). \tag{3.83}$$

3.4.3 Experimental Evaluation

To evaluate our proposed deep conditional generative models on the *WS-VOG* and *WS-GVD* tasks, we use three challenging benchmark datasets: the Flickr30k Entities [248] image dataset and the ActivityNet Entities [4] and YouCook2 BB [89] video datasets. These datasets are suitable for evaluating our method since they provide RGB data paired with visual descriptions and ground-truth bounding boxes for the objects that each noun/pronoun in the sentences refers to. Since we are operating on the weakly-supervised grounding regime, we ignore bounding box annotations during training, and we use them only to evaluate our method.

We trained and evaluated multiple variations of our latent-variable model of sentences, which stem from different choices for (a) the language decoder, (b) the approximate posterior word-to-region alignment distribution, (c) the usage of transferred object classifier knowledge in the parameterization of the approximate posterior, (d) the prior word-to-region alignment distribution, and (e) whether the visual encoder yields contextual region embeddings. Table 3.3 gives an overview of the variants of our deep conditional generative model.

After presenting the datasets and implementation details in Section 3.4.3.1, we begin our experimental evaluation by comparing the grounding performance given both ground-truth and generated sentences for the multiple variants of our GVD-CVAE framework on the Flickr30kEntities dataset in Section 3.4.3.2. We also compare our GVD-CVAE with strong baselines, such as the discriminative soft-attention-based encoder-decoder model of Zhou et al. [4], which we reviewed in Section 3.3.1. Experimental results clearly demonstrate that, while using the same visual encoder and language decoder, leveraging the variational approximate posterior word-to-region alignment distribution outperforms using soft-attention coefficients for grounding. Interestingly, even grounding based on our learned *prior* word-to-region alignment distribution improves upon the soft-attention baseline by a significant margin in both tasks. In Section 3.4.3.3 we continue our discussion by comparing our best performing GVD-CVAE model variant with state-of-the-art methods in the three datasets, including methods that are tailored towards only one of the

Table 3.3: Overview of different variants of our GVD-CVAE framework. They correspond to different choices of (a) the language decoder, (b) the approximate posterior word-to-region alignment distribution, (c) the usage of transferred object classifier knowledge in the parameterization of the approximate posterior (*Obj. Cls.*), (d) the prior word-to-region alignment distribution, and (e) whether the visual encoder yields contextual region embeddings (*Obj. Int.*).

		Approximate Posterior			
Model	Decoder	Cond.	Obj. Cls.	Prior	Obj. Int.
GVD-CVAE-L-Fw	LSTM	$z_t \| y_{\leq t}$	✗	$z_t \| \mathbf{y}_{<t}$	✗
GVD-CVAE-L-FwBw	LSTM	$z_t \| y_{\leq T}$	✗	$z_t \| \mathbf{y}_{<t}$	✗
GVD-CVAE-L-FwBwGrd	LSTM	$z_t \| y_{\leq T}$	✓	$z_t \| \mathbf{y}_{<t}$	✗
GVD-CVAE-U-Fw	UpDown	$z_t \| y_{\leq t}$	✗	$z_t \| \mathbf{y}_{<t}$	✗
GVD-CVAE-U-FwBw	UpDown	$z_t \| y_{\leq T}$	✗	$z_t \| \mathbf{y}_{<t}$	✗
GVD-CVAE-U-FwBwGrd	UpDown	$z_t \| y_{\leq T}$	✓	$z_t \| \mathbf{y}_{<t}$	✗
GVD-MCVAE-L-Fw	LSTM	$z_t \| z_{t-1}, y_{\leq t}$	✗	$z_t \| z_{t-1}, \mathbf{y}_{<t}$	✗
AO-GVD-CVAE-L-Fw	LSTM	$z_t \| y_{\leq t}$	✗	$z_t \| \mathbf{y}_{<t}$	✓

two tasks, e.g., methods that can only tackle *WS-VOG* and cannot both generate visual descriptions and ground the generated words. In Section 3.4.3.4 we present ablation studies on the effect of (a) each term of the training objective, (b) the β scheduler for mitigating posterior collapse, (c) the attention mechanism used to parameterize the approximate posterior distribution, and (d) the various hyperparameters, such as the number of samples, sampling temperature, and β_{clip}. Last, we provide qualitative results for both tasks and both image and video data in Section 3.4.3.5.

3.4.3.1 Datasets and Implementation Details

Datasets. We use the following three challenging benchmark datasets:

– *Flickr30k Entities (F30k):* This is a large-scale image dataset, originally annotated with phrase-to-region alignments [248]. To evaluate our results on object grounding (rather than phrase grounding), we follow the setup from Zhou et al. [4] to convert each noun phrase (e.g. *her brown hat*) associated with each bounding box to a single groundable object, such as *hat*. This results in $|\mathcal{V}_o| = 480$ groundable words out of the $|\mathcal{V}| = 8,639$ words comprising the vocabulary. We use the standard dataset split with 29k/1k/1k images in the training, validation and testing sets, respectively.

– *ActivityNet Entities (ANet):* As we discussed in Section 3.3.5.1, this is a large-scale video dataset, containing $52k$ video segments annotated with a caption each. Following the original setup [4], we use a vocabulary of $4,905$ words, 431 of which are groundable. Each groundable word in a sentence is associated with a bounding box in a frame of the video where it can be clearly observed.

– *YouCook2:* This is a video dataset containing YouTube cooking videos with video segments paired with captions and bounding box annotations [89] at 1 fps for 67 object classes. We use the same training/validation/test split as [9].

All our models are evaluated with grounding and captioning metrics as detailed in Section 3.3.5.

Implementation details. For the F30k and Anet datasets, our GVD-CVAE receives as inputs the region proposals, region features and image/video global features from Zhou et al. [4], with 100 region proposals per frame/image. For YouCook2, we use 20 region proposals and the features extracted by Shi et al. [9]. Hyperparameters such as learning rate, β_{clip}, attention mechanisms, number of samples, are chosen based on the validation sets of F30k and

YouCook2. Since annotations for the testing set of ANet are not public and the evaluation server is closed at the time of submission, we follow [249] and report results on the validation set. To do so, we train a model with hyperparameters selected based on the F30k validation set. All other hyperparameters, such as layer sizes, are in general adopted from prior work [4, 9]. Additional training and implementation details for the models in Tables 3.7, 3.8 and 3.10 are reported next.

– *Flickr30k Entities Inputs:* We use the same region proposals and features as Zhou et al. [4][4]. For each image, we use a Faster R-CNN [72] detector with the ResNext-101 [238] backbone pretrained on Visual Genome [182] to obtain region proposals. In particular, we retain the top 100 region proposals per frame, based on their detection confidence score. Each region is described by a $2,048$-dimensional feature vector extracted from the $fc6$ layer of the ResNext-101. Following GVD [4], we use a global feature vector of size $2,048$ to describe the image being captioned. We use a vocabulary of $8,639$ words including `UNK` (the symbol for rare words not included in the vocabulary) and `EOS` (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch, same as in GVD [4].

– *Flickr30k Entities Model:* The pre-extracted region features, image convolutional features and global image feature are transformed into X, F and $\mathbf{v}$ by our trainable encoder (which mirrors the encoder of GVD [4]). In particular, the region embedding consists of the concatenation of: a linear projection

[4] https://github.com/facebookresearch/grounded-video-description/tree/flickr_branch

of the $fc6$ region feature (initialized with the $fc7$ layer weights of the object detector), a 300-dimensional trainable embedding of the 4-dimensional position of the bounding box coordinates, and a 481-dimensional vector of object classification scores obtained by applying a trainable object classification layer on top of the $fc7$ feature. After normalizing these 3 components with layernorm, they are concatenated and passed through a linear projection that projects to a lower-dimensional space of dimensionality 1,024. This serves as our *grounding-aware region embedding* [4] $\mathbf{x}_i$. Similarly, the convolutional features and global image feature are projected with two linear transformation layers to a lower-dimensional space of dimensionality 1,024, yielding F and $\mathbf{v}$, respectively.

We use the GVD-CVAE-U-FwBwGrd variant of our GVD-CVAE. In particular, our decoder has at its core a two layer (hierarchical) LSTM of hidden size 1,024. The convolutional attention $f_\theta(\cdot,\cdot)$ is an additive attention mechanism of attention size 512, which takes in the convolutional feature map F and determines by the hidden state $\mathbf{u}_t$ how significant each feature map column should contribute to generate a word ($f_\theta^{(l)}(\mathbf{u}_t, F)$). The same holds for the region attention $k_\theta(\cdot,\cdot)$. Our p-attention network is an additive attention mechanism with attention size 512. Our full inference model uses the grounding aware approximate posterior and consists of a Bi-LSTM with hidden size 1,024 and a q-attention network with dot-product attention mechanism ($\alpha_\phi^{(i)}(\mathbf{h}_t, X) \propto \mathbf{h}_t^T U \mathbf{x}_i$).

– *Flickr30k Entities Training:* We train our model for 40 epochs with the Adam [171] optimizer, having an initial learning rate of 2e−4 decayed by a factor 0.8 every

3 epochs. Our batch size is 40 images, the number of Monte Carlo samples is $S = 10$, and the Gumbel-Softmax temperature is $\tau = 0.8$. Note that we start training with $\lambda = 0$ for 20 epochs and then add the ELBO loss and jointly optimize the cross-entropy and ELBO losses ($\lambda = 0.5$). For annealing β, we use the PI-Controller [223], with $Ki = -0.0001$, $v_{kl} = 0.06$, $Kp = 0.01$. Hyperparameters were either borrowed by GVD or were chosen based on Box accuracy on the validation set. We apply dropout with probability 0.5 to fully-connected layers. All layers are trained from scratch, except for the backbones yielding the initial region and image features. Ground-truth captions are truncated to 20 words during training and testing.

– *Flickr30k Entities Evaluation:* We evaluate our model on weakly-supervised object grounding and grounded visual description on the validation and testing sets. We have chosen to evaluate the checkpoint at the end of training, instead of the checkpoint that achieves the best metrics. We use the GVD metrics and evaluation scripts for evaluating: captioning and grounding [5]. This yields the reported result in Table 3.7.

– *ActivityNet Entities:* We use the same architecture and hyperparameters for our visual encoder and language decoder as described in Section 3.3.5. For our GVD-CVAE, we used the architecture and hyperparameters selected on the Flickr30k Entities validation set, except for the q-attention network, which is implemented as additive attention in this case. We also adjusted the learning rate for a larger batch size and reduced the number of epochs (for faster

[5] https://github.com/facebookresearch/grounded-video-description/blob/44411533ea967244867a6b186a9b5cebba476015/eval_grd_flickr30k_entities.py

training). Namely, we set a batch size of 60 videos, a learning rate of 3e$-$4 and trained for 30 epochs, reporting validation results with the model obtained at the end of training. Following [4], we uniformly sample 10 frames from each video segment during training and testing. Ground-truth captions are truncated to 20 words during training and testing.

– *ActivityNet Entities Evaluation:* We evaluate our model (the checkpoint at the end of training) on weakly-supervised object grounding and grounded captioning on the validation set. Unfortunately, the official CodaLab evaluation server[6] is currently closed. We use the official metrics and evaluation scripts for evaluating: captioning[7] and grounding[8].

– *YouCook2 Inputs:* We use the same region proposals and features as Shi et al. [9]. For each frame, we use a Faster R-CNN [72] detector with VGG-Net [37] backbone pretrained on Visual Genome [182] to obtain region proposals. In particular, we retain the top 20 region proposals, based on their detection confidence score. Each region is described by a 4,096-dimensional feature vector extracted from the $fc7$ layer of the VGG-Net. We also combine that region feature with a 300-dim trainable embedding of the bounding box coordinates (including the normalized frame index). We also use a global feature vector of size 3,072 describing the video segment to be captioned, which is obtained by averaging the temporal sequence of frame-wise appearance and motion features from [239]. Following GVD [4] the global feature vector is augmented

[6] https://competitions.codalab.org/competitions/20537

[7] https://github.com/LuoweiZhou/densevid_eval_spice/blob/bbab10c202e956266031a0dd6c791cba25b58e59/evaluate.py

[8] https://github.com/facebookresearch/ActivityNet-Entities/blob/aa5cd28383e5e9c63e875ada54057591a71509d9/scripts/eval_grd_anet_entities.py

with a 50-dimensional embedding of the segment positional information (i.e., total number of segments, segment index, start time and end time). We note here that MIL-based methods in this dataset do not use that global feature vector. We use a vocabulary of 1,009 words including UNK (the symbol for rare words not included in the vocabulary) and EOS (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch (in contrast to Shi et al. [9], who use pre-trained GloVE word embeddings for the groundable words).

– *YouCook2 Model:* The pre-extracted region and global video features are transformed into X and $\mathbf{v}$ by our trainable encoder, i.e. a pair of two linear transformation layers that project features to a lower-dimensional space of dimensionality 1,024. Our decoder has at its core a single layer LSTM of hidden size 1,024. Our p-attention network is an additive attention mechanism with attention size 512. Our inference model consists of a BiLSTM with hidden size 1,024 and a q-attention network with an additive attention mechanism of size 512.

– *YouCook2 Training:* We train our model for 40 epochs with the Adam [171] optimizer, having an initial learning rate of 1e$-$4 decayed by a factor of 0.8 every 3 epochs. Our batch size is 80 video segments and $S = 10$, $\tau = 0.8$, and $\lambda = 0.5$. The latter were chosen based on Box accuracy on the validation set. For annealing β, we use the PI-Controller [223], with $Ki = -0.0001$, $v_{kl} = 0.1$, $Kp = 0.01$. We apply dropout with a probability of 0.5 on fully-connected layers. All layers are trained from scratch, except for the backbones yielding the initial region and video features. Following [89, 9], we randomly sample

5 frames from each video segment during training, while we use all frames (extracted at 1fps) during testing. Ground-truth captions are truncated to 20 words during training and whole captions are used during testing (maximum sentence length 46 words).

– *YouCook2 Evaluation:* We evaluate our model (the checkpoint at the end of training) on the validation and testing sets using the same experimental setup and metrics as in NAFAE [9][9]. We use the CVAE prior distribution to ground each groundable word in each frame. We made this choice, since CVAE-p outperformed the CVAE approximate posterior (CVAE-q) in the validation set of this dataset.

3.4.3.2 Comparison of Models

Are the regions localized via our learned word-to-region alignment distributions better than those localized via soft-attention-based baselines? We begin by comparing the grounding performance of our GVD-CVAE (and in particular the GVD-CVAE-U-FwBwGrd variant) with three soft-attention-based baselines (p-attention, GVD, GVD-Grd) on the validation sets of Flickr30k Entities (images) and ActivityNet Entities (videos).

The first baseline (p-attention) consists of our language decoder and p-attention network trained by minimizing the cross-entropy sentence generation loss (by setting $\lambda = 0$ in our hybrid objective defined in Eq. 3.54). Once a model is trained, grounding is performed by selecting the region at each frame with the highest p-attention coefficient $a_\theta^{(i)}(\mathbf{s}_t, X)$. These attention coefficients

[9] https://github.com/jshi31/NAFAE/blob/master/lib/datasets/youcook_eval.py

are computed with the p-attention network given the hidden state of the top LSTM $\mathbf{s}_t$ as the query. It should be clear that this p-attention baseline is the exact soft-attention counterpart of our variational GVD-CVAE.

The other baselines are based on the popular attention-based encoder-decoder captioning model [4] (Section 3.3.1), trained with the teacher-forcing language generation cross-entropy loss. For a fair comparison, we ensure that our p-attention baseline and our GVD-CVAE exactly mirror the inputs and the visual encoder/language decoder modules of the GVD model. As discussed earlier in this chapter, grounding given this model can be performed: (a) by selecting the region with maximum region attention coefficient $k_\theta^{(i)}(\mathbf{u}_t, X)$ given the partial caption $\mathbf{y}_{<t}$ (Eq. 3.20, GVD), or (b) by combining the attention coefficients with region-to-class similarity scores based on the word $\mathbf{y}_t$ to be grounded for the *VOG* task (Eq. 3.22, GVD-Grd). Note that both the p-attention and GVD baselines ground words with soft-attention coefficients computed based on the partial caption, and they only differ in the query vector that is used to compute the region attention coefficients ($\mathbf{s}_t$ or $\mathbf{u}_t$).

In Table 3.4, we compare our GVD-CVAE's ability to ground objects in ground-truth or generated sentences with these powerful, discriminative baselines. We observe that even grounding based on our learned *prior* word-to-region alignment distribution (GVD-CVAE-p) improves upon both soft-attention baselines (*p-attention* and GVD) by a significant margin in both benchmarks and tasks, despite similarly capturing only the history of previous words. Focusing on the Flickr30k Entities dataset, the GVD-CVAE-p improves grounding accuracy given ground-truth sentences (Box Accuracy) from 19%

Table 3.4: Comparison of grounding performance between the p-attention, GVD and GVD-Grd models (baselines) and our GVD-CVAE on the validation sets of F30k and ANet. We report the box accuracy metric for evaluating grounding given ground-truth sentences and the $F1_{all}$ metric for evaluating grounding of object words in generated sentences. GVD-CVAE-p (GVD-CVAE-q) denotes using our learned prior (approximate posterior) alignment distribution for grounding.

Dataset	Method	Box Acc.	$F1_{all}$
F30k (Image)	p-attention (Ours)	17.0	4.2
	GVD [4]	22.0	4.4
	GVD-Grd [4]	25.9	4.4
	GVD-CVAE-p (Ours)	29.6	6.2
	GVD-CVAE-q (Ours)	**33.4**	**7.3**
ANet (Video)	p-attention (Ours)	11.6	3.1
	GVD [4]	14.9	3.7
	GVD-Grd [4]	21.3	3.7
	GVD-CVAE-p (Ours)	19.4	4.8
	GVD-CVAE-q (Ours)	**24.2**	**6.1**

to 29%, although the two models differ only in their training, and in particular whether the CVAE loss is used ($\lambda = 0.5$). The reason for this improvement is that our prior distribution is encouraged during training to "look ahead" when sampling a region to generate a word, by mimicking the approximate posterior alignment distribution which has access to future words. Using the latter for grounding conditioned on the full sentence further improves results (from 29.6% to 33.4%), verifying our intuition that leveraging the word to be grounded in its language context can help us better localize the word. Importantly, our GVD-CVAE-p improves the Box Accuracy of the strong GVD baseline from 22% to 29% on the same dataset. Additionally, our GVD-CVAE-q achieves an absolute accuracy gain of 7.5% on Box Accuracy over the GVD-Grd discriminative baseline which also takes into account the word to be

Table 3.5: Comparison of our deep conditional generative models on the Flickr30k Entities validation set. *Obj. Cls.* denotes inference model with transferred object class knowledge.

		Approximate Posterior		Box Acc.		$F1_{all}$	
Model	Decoder	Cond.	Obj. Cls.	p	q	p	q
GVD-CVAE-L-Fw	L	$z_t \| y_{\leq t}$	✗	30.3	31.8	6.7	7.4
GVD-CVAE-L-FwBw	L	$z_t \| y_{\leq T}$	✗	27.5	31.3	5.7	6.6
GVD-CVAE-L-FwBwGrd	L	$z_t \| y_{\leq t}$	✓	30.2	34.8	6.9	7.5
GVD-CVAE-U-Fw	U	$z_t \| y_{\leq t}$	✗	26.3	31.4	6.0	7.4
GVD-CVAE-U-FwBw	U	$z_t \| y_{\leq T}$	✗	25.1	32.3	5.4	7.0
GVD-CVAE-U-FwBwGrd	U	$z_t \| y_{\leq t}$	✓	29.6	33.4	6.2	7.3
GVD-MCVAE-L-Fw	L	$z_t \| z_{t-1}, y_{\leq t}$	✗	26.7	27.4	5.2	6.5

grounded, demonstrating the benefits of our conditional generative modeling. These conclusions are further confirmed by our comparison with baseline models on the video dataset of ActivityNet Entities, where we can observe the same trends.

How does the choice of the language decoder, the deep conditional generative model, and the variational posterior affect grounding performance? Table 3.5 demonstrates the grounding performance given ground-truth sentences or sentences generated by our model obtained for different variants of our GVD-CVAE model.

Interestingly, our GVD-CVAE achieves top grounding performance even when using a simple LSTM in the decoder (GVD-CVAE-L variants) instead of the more expressive hierarchical UpDown model, that also uses two extra attention mechanisms. In fact, the GVD-CVAE-L-FwBwGrd variant yields a Box Accuracy of 34.8% which outperforms the GVD-CVAE-U-FwBwGrd model which has an accuracy of 33.4. This demonstrates the effectiveness of our latent-variable modeling. We will still use the UpDown language model for our comparisons with state-of-the-art methods on ActivityNet Entities and Flickr30k Entities, since they also use the UpDown decoder, but we will focus on the simpler LSTM model for the rest of our experiments.

Another observation is that explicitly adding transferred information about object class distributions in the inference model (object-aware approximate posterior) improves grounding given ground-truth sentences (Box Accuracy) when the approximate posterior distribution is used. For example, the GVD-CVAE-L-FwBwGrd model improves upon the GVD-CVAE-L-FwBw model by 3%, 31.3% → 34.8%, which is in line with our expectation. Note that the GVD-CVAE-L-FwBwGrd model also outperforms the baseline of using the transferred object detector knowledge to ground each object word with its closest object classifier from the Visual Genome dataset, which yields 29%. Even more interestingly, the prior alignment distribution of the model trained with the object-aware smoothing approximate posterior improves over the one of the model trained with just the smoothing approximate posterior. This demonstrates that knowledge from the inference model is distilled to the prior during training via the KL divergence loss, resulting in a model that is looking at better localized regions *while generating* descriptions based on the prior and decoder modules.

When using the UpDown language model, taking the full sentence into account via a BiLSTM, i.e. using a smoothing posterior $q_\phi(\mathbf{z}_t|\mathbf{y}_{\leq T})$, leads to better grounding compared to only seeing the sentence up to the current word $y_{\leq t}$ with an LSTM (e.g., improving Box Acc. from 31.4% to 32.3%). The difference between GVD-CVAE-L-Fw and GVD-CVAE-L-FwBw models is smaller when using the LSTM language model, and we observe that in both cases the grounding of generated sentences does not improve.

Last, we report preliminary results with the GVD-MCVAE model in the last row. We observe that the model that captures dependencies between word-to-region alignments of groundable words performs slightly better than the model that captures dependencies between consecutive words, as expected. However, we observe that the GVD-MCVAE-L-Fw performs worse than the GVD-CVAE-L-FW model, despite this additional modeling of pairwise dependencies between word-to-region alignments. One possible explanation might be that we simply transferred hyperparameter values (including the training setup) from the GVD-CVAE model, however these might not be optimal for the GVD-MCVAE model, which differs both in architecture and training objective. We plan to conduct further experiments to determine an optimal experimental setup for this variant. Furthermore, we were not able to train a GVD-MCVAE without the approximate posterior collapsing to the prior with a bilinear dot-product attention mechanism for the q-attention network (as the one we use for the GVD-CVAE), and it was critical to use an additive attention mechanism.

What is the effect of using contextual region embeddings on weakly supervised grounding performance in videos? For all the experiments reported so far for both our GVD-CVAE model and the baselines, we have been representing each region proposal with the grounding-aware region embedding.

This embedding only captures local cues, such as the object appearance, the bounding box position and the transferred object class knowledge. Here, we experiment with context-aware region embeddings, i.e., embeddings that take into account the spatio-temporal interactions among the detected regions. In particular, we study two approaches for learning contextual region embeddings: (a) a Transformer model with multi-head self-attention applied on the set of region proposals, (b) our V-HetGAT model from Chapter 2 that models spatio-temporal interactions among actor and object regions based on a visual spatio-temporal graph (described in Section 3.3.2). We report results in Table 3.6. On the one hand, we observe that using our V-HetGAT model for computing the contextual embeddings improves upon multi-head attention in this weakly-supervised regime for the baseline GVD model. Recall that in the fully-supervised training setup there was no significant variation in performance between these two approaches. Additionally, the variant of GVD-CVAE model with context-aware region embeddings outperforms all the GVD and GVD-Grd baselines, following the trend of the rest of our variants. On the other hand, these results, including results from Zhou et al. [4], suggest that using local region embeddings leads to better grounding performance. We conjecture that the context-aware region representations which are varying during training confuse the attention mechanisms, which without grounding supervision fail to appropriately attend to the right region. This leads to a significant drop in performance from 24.2 to 16.4 when we combine our V-HetGAT with the GVD-CVAE-U-FwBwGrd model.

Table 3.6: Weakly-supervised Visual Object Grounding with contextual region embeddings on ActivityNet Entities [4] validation set. Choices for region representation (Region Repr): (*Local*) grounding-aware region featuresm (*MHA*): region-interaction-aware region features obtained via multi-head self-attention applied on grounding-aware region featuresm, (*V-HetGAT*): actor-object interaction-aware region embeddings obtained via applying our V-HetGAT model on grounding-aware region features.

Method	Region Repr.	Box Acc.
GVD [4]	Local	14.9
GVD-Grd [4]	Local	21.3
GVD [4]	MHA	2.4
GVD	V-HetGAT	11.0
GVD-Grd [4]	MHA	19.7
AO-GVD-CVAE-U-FwBwGrd (CVAE-p)	V-HetGAT	16.1
AO-GVD-CVAE-U-FwBwGrd (CVAE-q)	V-HetGAT	16.4
GVD-CVAE-U-FwBwGrd (CVAE-p)	Local	19.4
GVD-CVAE-U-FwBwGrd (CVAE-q)	Local	24.2

3.4.3.3 Comparison with the State of the Art

As shown in Table 3.7, our GVD-CVAE improves weakly-supervised object grounding by 12% compared to the GVD method (21% to 33.8%) on the F30k image dataset. Thus, it sets the state-of-the-art *VOG* result, and closes the gap with the fully-supervised GVD approach (41.4%). It also generates more grounded captions (higher $F1_{all}$ and $F1_{loc}$ scores) than all other methods, given the same features from Zhou et al. [4]. We even outperform methods using Scene Graphs [6] for grounding [250]. Note that the $F1_{all}$ scores obtained by both our CVAE-p (6.43) and CVAE-q (6.88) distributions outperform Cyclical [197] (4.85) and DPA [198] (4.79). This suggests that modeling alignments as latent variables works better than applying attention regularization techniques during training. Despite generating more grounded captions, our

Table 3.7: Comparison with state-of-the-art methods on the Flickr30k Entities test set. The performance of the fully-supervised GVD model (Sup.) is reported as an upper-bound to the weakly-supervised approaches. Types of model inputs during inference: region proposals extracted and encoded following **G**VD [4] or B**U**TD [5], or **S**cene-graphs [6]. * denotes models trained using auxiliary image-to-text matching models [7]. SCST denotes models fine-tuned via Reinforcement Learning [8]. As it can be seen, results in the third block are obtained with different inputs, and thus they are not directly comparable to ours. We report average results for our GVD-CVAE after 5 random runs.

		VOG	GVD					
			Captioning				Grounding	
	Feat	Acc	B@4	M	C	S	$F1_{all}$	$F1_{loc}$
GVD [4] (Sup.)	G	41.4	27.3	22.5	62.3	16.5	7.55	22.2
GVD [4]	G	21.4	26.9	22.1	60.1	16.1	3.88	11.7
GVD-Grd [4]	G	25.5	26.9	22.1	60.1	16.1	3.88	11.7
Cyclical [197]	G	-	26.6	22.3	60.9	16.3	4.85	13.4
DPA [198]	G	-	27.6	22.6	62.7	16.7	4.79	15.5
SCAN-SCST [199]*	G	-	28.0	22.6	66.2	17.0	6.53	15.8
BUTD [5]	U	24.2	27.3	21.7	56.6	16.0	-	-
DPA [198]	U	-	27.2	22.3	60.8	16.3	5.45	15.3
Sub-GC [250]	S	-	28.5	22.3	61.9	16.4	5.98	16.5
SCAN-SCST [199]*	U	-	**30.1**	22.6	**69.3**	16.8	**7.17**	17.5
GVD-CVAE	G	**33.7**	24.0	21.3	55.3	15.7	6.70	**19.2**
GVD-CVAE-SCST	G	31.6	29.8	**23.1**	67.6	**17.2**	6.94	17.6

method has lower captioning metrics than SoTA methods, some of which apply reinforcement learning (RL). However, our language model can also be finetuned with a CIDEr-based SCST loss [8] (GVD-CVAE-RL), leading to competitive captioning metrics. In particular, we finetune the parameters of our decoder and prior networks with SCST using CIDEr as the reward. Since the goal of this experiment was to show that our decoder can be finetuned with RL to match the performance of SoTA models in captioning, we chose

Table 3.8: Comparison with state-of-the-art methods on the ActivityNet Entities validation set. We report average results for our GVD-CVAE after 5 random runs.

	VOG	GVD					
		Captioning				Grounding	
	Acc	B@4	M	C	S	$F1_{all}$	$F1_{loc}$
GVD (Sup.) [4]	35.7	2.59	11.2	47.5	15.1	7.1	24.1
MIL-based							
NAFAE [9]	19.5	-	-	-	-	-	-
STVG [251]	21.1	-	-	-	-	-	-
SCL [249]	23.8	-	-	-	-	-	-
Captioning-based							
GVD [4]	14.9	2.28	10.9	45.6	**15.0**	3.7	12.7
GVD-Grd [4]	21.3	2.28	10.9	45.6	15.0	3.7	12.7
Cyclical [197]	-	**2.45**	**11.1**	**46.4**	14.8	4.7	15.8
GVD-CVAE	**23.9**	1.90	10.4	41.8	13.3	**5.8**	**21.7**

to finetune a simpler model (hence the small reduction in weakly-supervised grounding from 33.8 to 31.6). In particular, we used the GVD-CVAE-L-Fw model that was trained with our hybrid loss until epoch 38 and then finetuned with SCST until epoch 60, with learning rate $5e-5$ and batch size 48.

Results on the ANet video dataset (Table 3.8) show similar trends. Our GVD-CVAE yields better metrics when grounding ground-truth or generated sentences. It also outperforms video-tailored, video-to-text matching models, such as NAFAE [9]. Although powerful, these models cannot tackle the grounded visual description task. Since we evaluate only on the validation set, we did not select the model with best CIDEr score, or tune the learning rate based on it. This led to slightly inferior captioning metrics compared to [4] and [197], who used the validation set for selecting a model to be evaluated

Table 3.9: Comparison with state-of-the-art methods on the YouCook2 test set following the experimental setup of Shi et al. [9]. We compare with methods that exploit various tasks for weakly-supervised learning (WSL): captioning (C) or matching with Multiple Instance Learning (M). Our captioning-based method is competitive with advanced MIL-based methods for weakly-supervised video object grounding and can additionally perform grounded captioning. Obj. Int.: modeling inter-object spatio-temporal interactions, e.g. using self-attention. Frm. Sim.: modeling word-to-frame similarity to better handle frames where the groundable word is occluded. Reg. Sim.: modeling similarity among grounded regions across frames for a groundable word.

		Method details			Box accuracy (%)		Query accuracy (%)	
	Task	OI	FS	RS	macro	micro	macro	micro
Upper Bound					62.41	-	-	-
DVSA-frm [87, 9]	M				37.55	44.16	39.31	46.14
Zhou [89, 9]	M	✓	✓		35.08	42.42	36.69	44.34
NAFAE [9]	M		✓	✓	40.71	46.33	42.45	48.41
STVG [251]	M	✓	✓		41.63	47.02	43.40	48.98
SCL [249]	M	✓			**42.80**	**48.60**	**44.61**	**50.61**
GroundR [196, 9]	C				19.94	-	-	-
GVD-CVAE (Ours)	C				38.85	44.62	40.71	46.63

on the now closed test server.

We also compare our method to MIL-based grounding approaches in the YouCook2 test split. As it can be seen in Table 3.9, although our method performed better or comparably to all video-to-text-matching methods on the ANet video dataset, it is lagging behind NAFAE [9] by around 2% in Box Accuracy on the YouCook2 dataset. A possible explanation is that, while in ANet grounding is evaluated on a single frame, in YouCook2 grounding predictions are evaluated in every frame. Therefore, the MIL-based methods that model the consistency between the localized regions at each frame are able to perform better. We believe that extending our GVD-CVAE to model relationships between the grounded regions at each frame will improve these

Table 3.10: Grounding performance comparison on YouCook2 validation set following the experimental setup of Shi et al. [9].

	Box accuracy (%)		Query accuracy (%)	
	macro	micro	macro	micro
Upper Bound	62.42	68.56	65.55	70.32
GroundR [196, 9]	19.63	-	-	-
DVSA-frm [87, 9]	36.90	44.26	38.48	46.27
DVSA-vid [87, 9]	36.67	43.62	38.20	45.60
MCOG [89, 9]	35.69	43.04	37.26	44.99
NAFAE [9]	39.54	46.41	41.29	48.52
STVG [251]	39.90	46.80	41.36	48.74
SCL [249]	**41.94**	**48.46**	**43.46**	**50.45**
GVD-CVAE (Ours)	38.85	45.91	40.54	48.01

Table 3.11: Impact of various training objectives on weakly-supervised object grounding. Performance measured via Box accuracy (%) on the F30k validation set.

Training objective	CVAE-p	CVAE-q
ELBO	3.29	3.16
CE + ELBO	25.22	23.99
CE + ELBO + β anneal	26.07	25.61
CE + ELBO + β anneal + clip	26.31	28.88
CE + ELBO + PI Controller	**29.27**	**31.71**

metrics, and we leave that to future work. For completeness, we also provide results on the validation set of YouCook2 in Table 3.10.

3.4.3.4 Ablation Studies

What is the effect of the proposed training objective? We first train our GVD-CVAE with the vanilla CVAE loss (ELBO), i.e., with $\lambda, \beta = 1$. Without any of our proposed modifications, this results in a very low grounding performance, as can be seen in the first row of Table 3.11. By adding the cross-entropy loss term that penalizes word predictions based on soft region context determined by the p-attention network (CE+ELBO), we are able to improve upon the soft-attention baseline of 22%. However, looking into the learning curves in Figure 3.9, reveals that the KL loss term has vanished by the end of training, suggesting that the model's posterior has collapsed to the prior and the approximate posterior alignment does not additionally take into account the word being grounded. Applying known solutions to KL vanishing, such as linearly annealing the β hyperparameter from 0 to 1 (CE+ELBO+β anneal), does not solve the problem. Instead, our proposed clipped linear annealing schedule leads to overall better grounding of 28.9% (KL term ≈ 0.06). Alternatively, after we determine a desirable value for the KL term, we can use the PI-Controller [223] to schedule β over training, which we found to be less sensitive to changes in architecture and requires minimal calibration. Note that in this ablation we used a single LSTM language decoder and an LSTM in the inference model for faster experimentation (GVD-CVAE-L-Fw).

What is the effect of the training objective hyperparameters (number of region samples, sampling temperature, β_{clip}) on the grounding performance? First, we train and test our GVD-CVAE-L-Fw model with varying number of samples $S = \{1, 5, 10, 15\}$ from the variational posterior for approximating the expectation of the sentence reconstruction term of the ELBO 3.55. Although it is common practice to train VAEs and CVAEs with a single sample from the variational posterior, results in Table 3.12 suggest that increasing the number of samples from 1 to 10 improves grounding performance when either the prior or approximate posterior word-to-region alignment distributions are

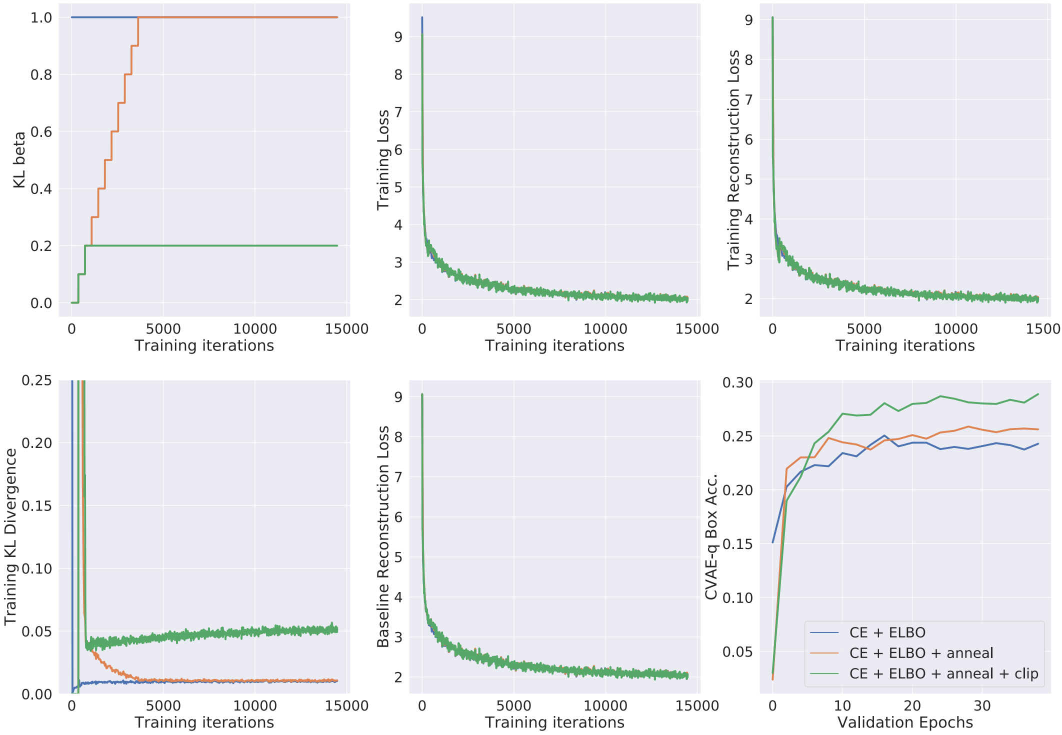


Figure 3.9: Comparison of learning curves when training the GVD-CVAE-L-Fw model variant for three schedules of the β hyperparameter on the Flickr30k training/validation sets. The clipped linear annealing schedule results in higher KL divergence (the approximate posterior does not collapse to the prior) and in higher grounding accuracy.

used. For example, the performance of CVAE-q increases from 26.75% to 28.88%. This improvement was expected, since more samples lead to a smaller error in the approximation of the expectation. However, further increasing the number of samples not only increases the memory requirements of our framework, but it might also degrade performance as shown for the case of 15 samples. Results also suggest that the sampling temperature, which is a hyperparameter of the Gumbel-Softmax continuous approximation from which we sample our latent variables $\mathbf{z}_t$, is important to tune. We use a temperature $\tau = 0.8$ for all our models in all three datasets. Last, we observe

Table 3.12: Ablation study on the Flickr30k Entities validation set. (const) denotes using a constant $\beta = \beta_{clip}$. We report results with the GVD-CVAE-L-Fw variant of our model. CVAE-p denotes box accuracy (%) obtained using the learned prior alignment distribution, while CVAE-q using the approximate posterior.

Ablation	β_{clip}	S	λ	τ	CVAE-p	CVAE-q
	0.2	10	0.5	0.8	26.31	**28.88**
Varying number of region samples	0.2	1	0.5	0.8	25.62	26.75
	0.2	5	0.5	0.8	27.19	27.93
	0.2	15	0.5	0.8	25.58	27.68
Varying temperature	0.2	10	0.5	0.5	25.95	26.84
Varying β scheduler	1.0	10	0.5	0.8	26.07	25.61
	0.1	10	0.5	0.8	26.03	28.23
	0.3	10	0.5	0.8	26.75	28.26
	1.0 (const)	10	0.5	0.8	25.22	23.99
	0.2 (const)	10	0.5	0.8	26.90	27.90

that the grounding performance does not vary significantly for varying values of β_{clip} in this dataset, as long as the β scaling factor is annealed linearly up to a small value smaller than 1. Annealing β up to 1, or using a small, but constant value for β significantly reduces performance (from 28.88% to 25.61% and 27.90%, respectively).

Limitations. Similar to all other proposal-based approaches to visual grounding, our model performance is limited by the quality of the region proposals. Also, we applied the same model for image and video object grounding to demonstrate its generality and effectiveness, without taking advantage of several inductive biases in the video domain, such as the visual similarity between grounded regions in consecutive frames.

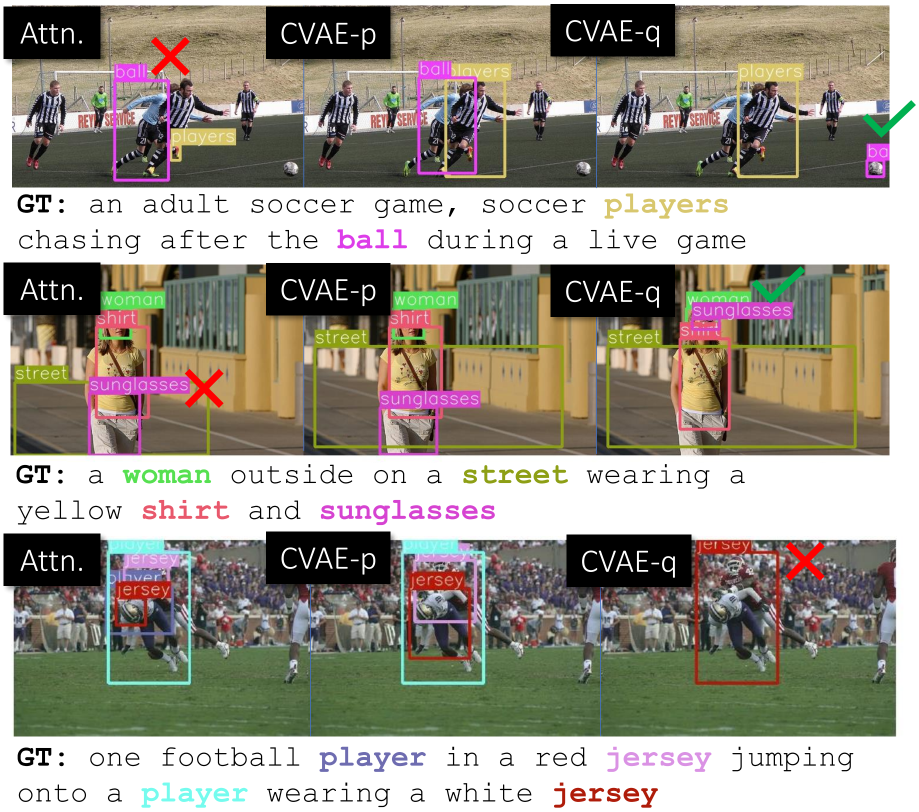
3.4.3.5 Qualitative Results

Finally, we show and discuss qualitative image grounding results on Flickr30k Entities (Figures 3.10, 3.13 and 3.14), and video grounding results on ActivityNet Entities (Fig. 3.15). We also show captions generated by our GVD-CVAE on Flickr30k Entities validation images (Fig. 3.16). We would like to emphasize that to obtain these qualitative results we used the the GVD-CVAE-L-Fw model variant trained with a clipped linear annealing schedule for β. We conclude, with grounding results on the two video datasets in Figures 3.11 and 3.17.

Computational Runtime. All models were implemented in Python using Pytorch and are based on the Grounded Video Description (`https://github.com/facebookresearch/grounded-video-description`) open-source code. Given pre-extracted video and region features, a forward pass through our model for performing grounding on 20 ActivityNet videos (10 frames sampled from each, $M = 1000$) takes 0.7 seconds at a single Tesla K80 GPU. We train our models on 4 GPUs and training lasts from around 6 to 24 hours depending on the dataset (training on the ActivityNet Entities video datasets lasts longer than training on the Flickr30k Entities image dataset).

3.4.4 Conclusion

In the second half of this chapter, we have proposed a novel grounded visual description CVAE for weakly-supervised learning of word-to-region aligments. We have designed a variety of deep conditional generative models of



GT: an adult soccer game, soccer **players** chasing after the **ball** during a live game

GT: a **woman** outside on a **street** wearing a yellow **shirt** and **sunglasses**

GT: one football **player** in a red **jersey** jumping onto a **player** wearing a white **jersey**

Figure 3.10: Qualitative weakly-supervised visual object grounding results on the Flickr30k Entities validation set. For each ground-truth caption, we show grounding results obtained by (a) the soft-attention baseline, (b) our prior, and (c) our approximate posterior alignment distributions. We observe that knowing the words to be grounded improves grounding of small objects. The third row shows a failure case, in which our CVAE-q predicts the same bounding box for all groundable words. Best viewed zoomed in and in color.

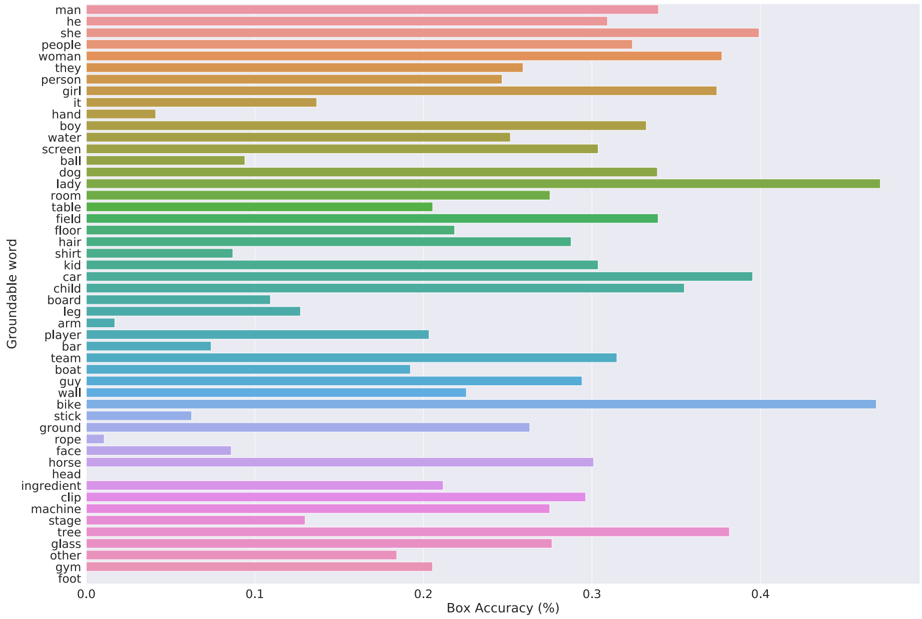


Figure 3.11: Weakly-supervised object grounding accuracy obtained by our GVD-CVAE-L-Fw model for the 50 most frequent classes in the ground-truth sentences of the ActivityNet Entities validation set.

sentences and explored the impact of architectures and conditional independence assumptions on the grounding performance. We have demonstrated the generality and effectiveness of our model by evaluating it on both image and video datasets. In particular, we have shown how leveraging the latent alignment distributions (both the *prior* word-to-region alignment distribution that only looks at previous words, and the *approximate posterior* distribution that looks at the full sentence) of our model significantly outperforms soft attention for grounding given ground-truth or generated sentences. Our novel approach also yielded competitive results under multiple metrics in both grounding and grounded video description while comparing against methods optimized for one of the two tasks. Notably, our GVD-CVAE has set the state-of-the-art *VOG* result on Flickr30k Entities, improving over grounding accuracy by 12% over GVD, and has reduced the gap with the fully-supervised GVD approach, which we discussed in the first half of this chapter.

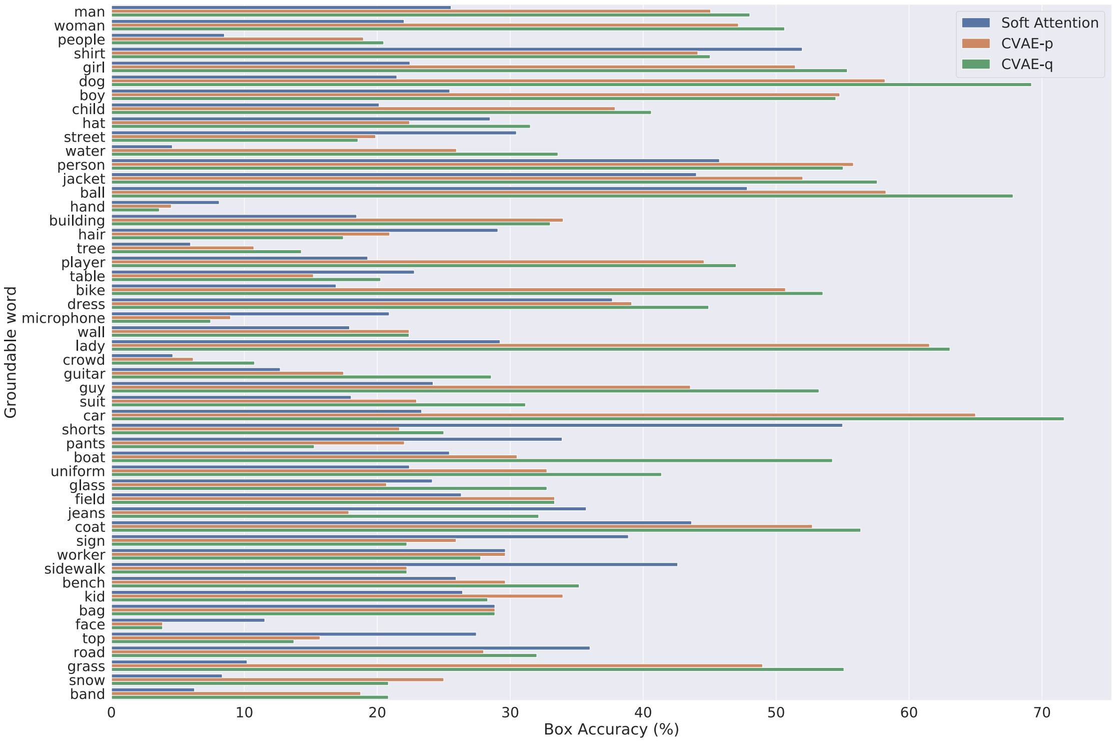
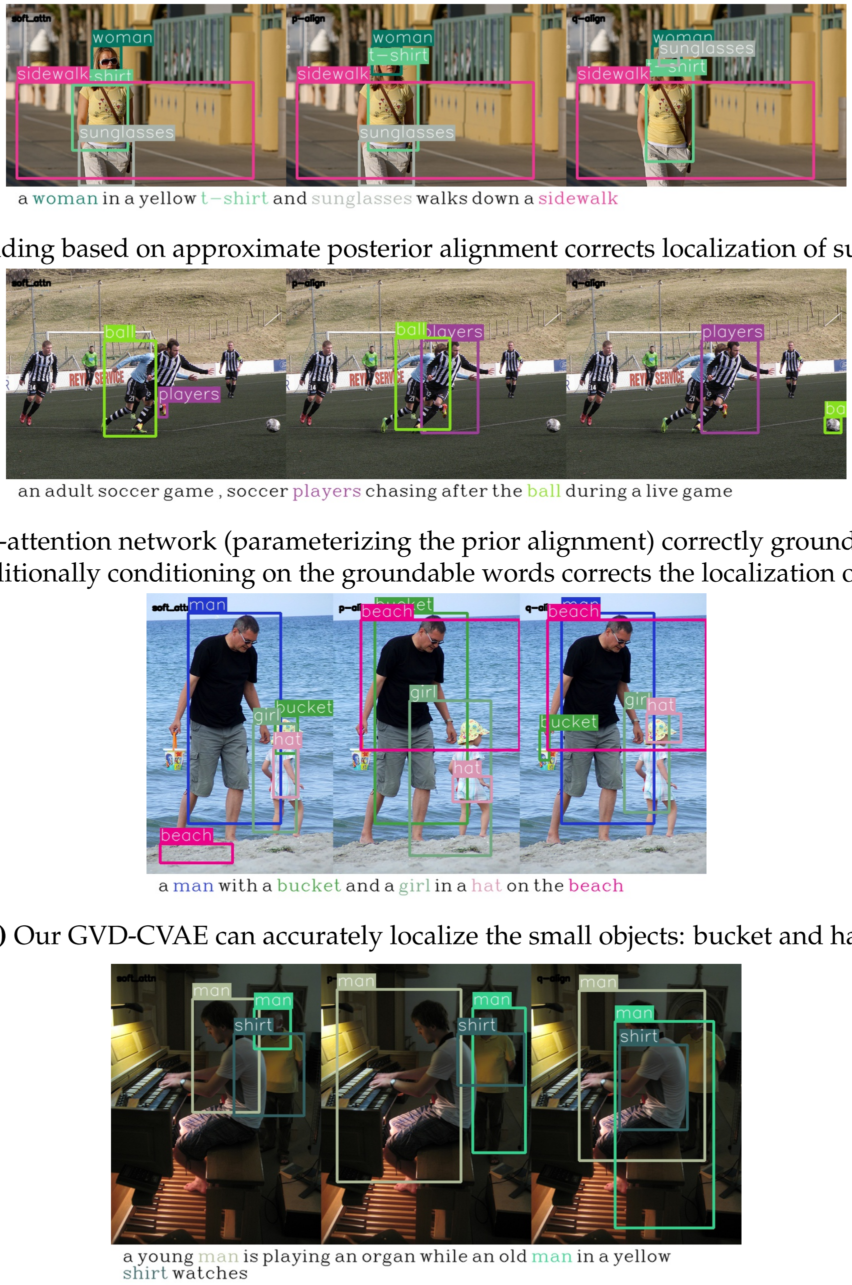


Figure 3.12: Comparison of weakly-supervised object grounding accuracy between the soft-attention and the prior and approximate posterior alignment distributions of our GVD-CVAE-L-Fw model for the 50 most frequent classes in the ground-truth sentences of the Flickr30k Entities validation set. Grounding with our GVD-CVAE leads to an improvement in most classes. We observe a large improvement in grounding based on gender when knowing the word to be grounded (CVAE-q), for example improving grounding of *girl*, *boy*, *lady*, *guy*, *woman*, *man*. Also, well-defined objects, such as *bike*, *car*, *ball*, *dog* are better grounded. Soft attention is better in some clothing classes, such as *shirt*, *short*, *pants*. All the compared methods fail to accurately ground objects like *face* and *hand*, which are both small and challenging to disambiguate from other faces/ hands. Furthermore, groups of objects, such as *crowd* and *people*, are challenging to ground with our one-to-one word- to-region correspondence assumption.


soft_attn
p−align
q−align
woman
sidewalk
shirt
t−shirt
sunglasses
a woman in a yellow t−shirt and sunglasses walks down a sidewalk
ball
players
ba
an adult soccer game , soccer players chasing after the ball during a live game
man
bucket
girl
hat
beach
a man with a bucket and a girl in a hat on the beach
man
shirt
a young man is playing an organ while an old man in a yellow shirt watches

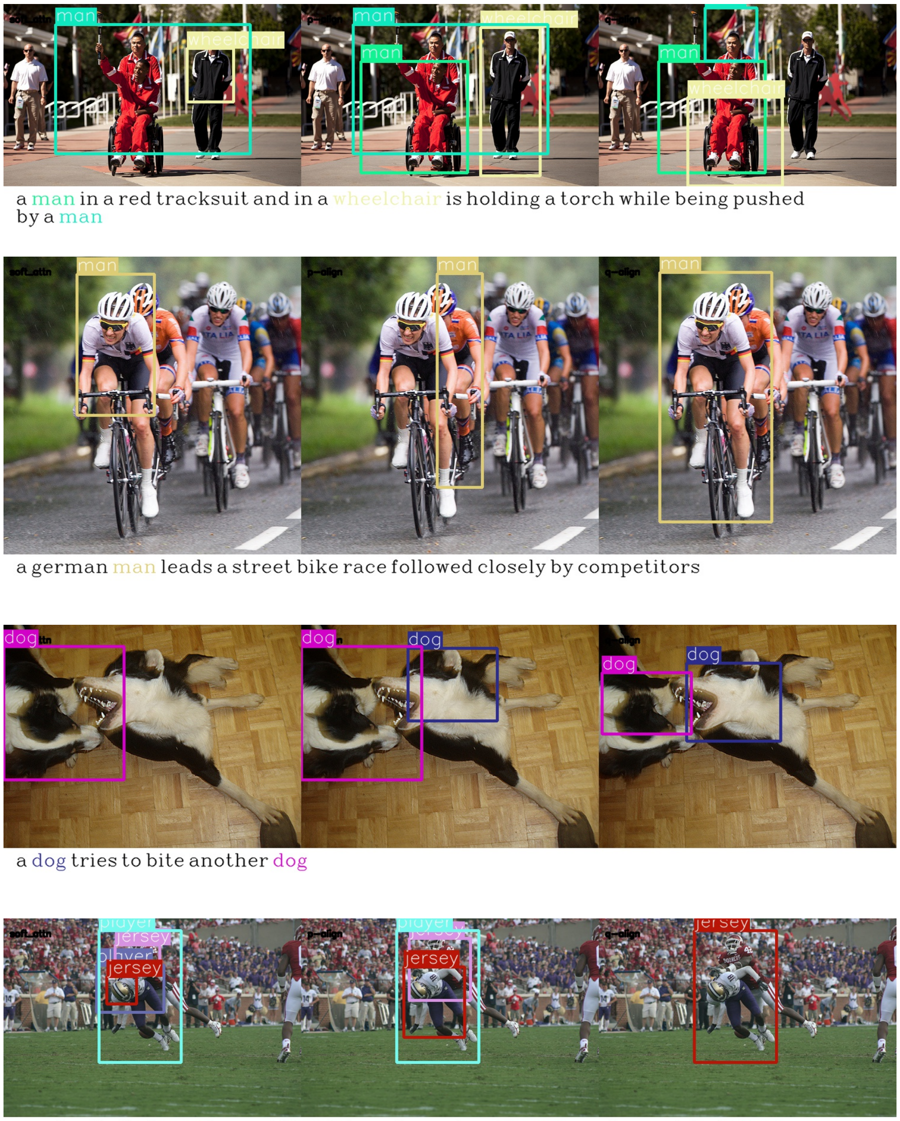
(a) Grounding based on approximate posterior alignment corrects localization of sunglasses.

(b) Our p-attention network (parameterizing the prior alignment) correctly grounds players, while additionally conditioning on the groundable words corrects the localization of the ball.

(c) Our GVD-CVAE can accurately localize the small objects: bucket and hat.

(d) Failure case: Our approximate posterior alignment fails to disambiguate between the two men and the two shirts. In contrast, our prior alignment which grounds based on: "[...] in a yellow", accurately localizes the shirt. Grounding based on whole phrases (yellow shirt) instead of individual words might help mitigate this issue.

Figure 3.13: Qualitative comparison of weakly-supervised object grounding results obtained by the baseline and our GVD-CVAE on images from Flickr30k Entities. For each caption, we show three copies of each image with grounding results obtained by the soft-attention baseline, our prior and posterior alignment distributions.



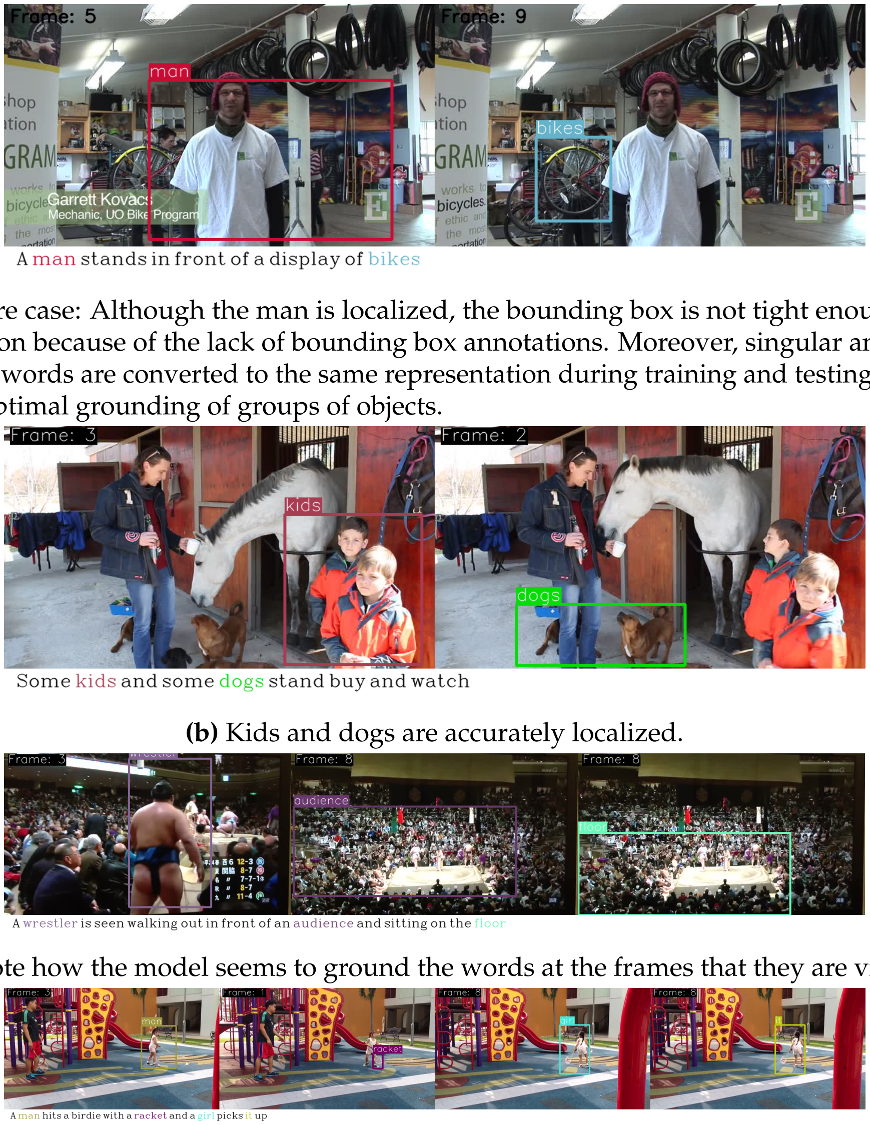
a man in a red tracksuit and in a wheelchair is holding a torch while being pushed by a man

a german man leads a street bike race followed closely by competitors

a dog tries to bite another dog

one football player in a red jersey jumping onto a player wearing a white jersey

Figure 3.14: More weakly-supervised object grounding results on images from Flickr30k Entities. For each caption, we show three copies of each image with grounding results obtained by the soft-attention baseline, our prior and posterior alignment distributions, respectively. Our GVD-CVAE (middle and right columns) can disambiguate between multiple instances of the same class (man, dog, etc.). The bottom figure shows a failure mode of our inference model, which might select the same region for all words.


Frame: 5
man
Frame: 9
bikes
A man stands in front of a display of bikes
Frame: 3
kids
Frame: 2
dogs
Some kids and some dogs stand buy and watch
Frame: 3
wrestler
Frame: 8
audience
Frame: 8
floor
A wrestler is seen walking out in front of an audience and sitting on the floor
Frame: 3
man
Frame: 1
racket
Frame: 8
girl
Frame: 8
it
A man hits a birdie with a racket and a girl picks it up

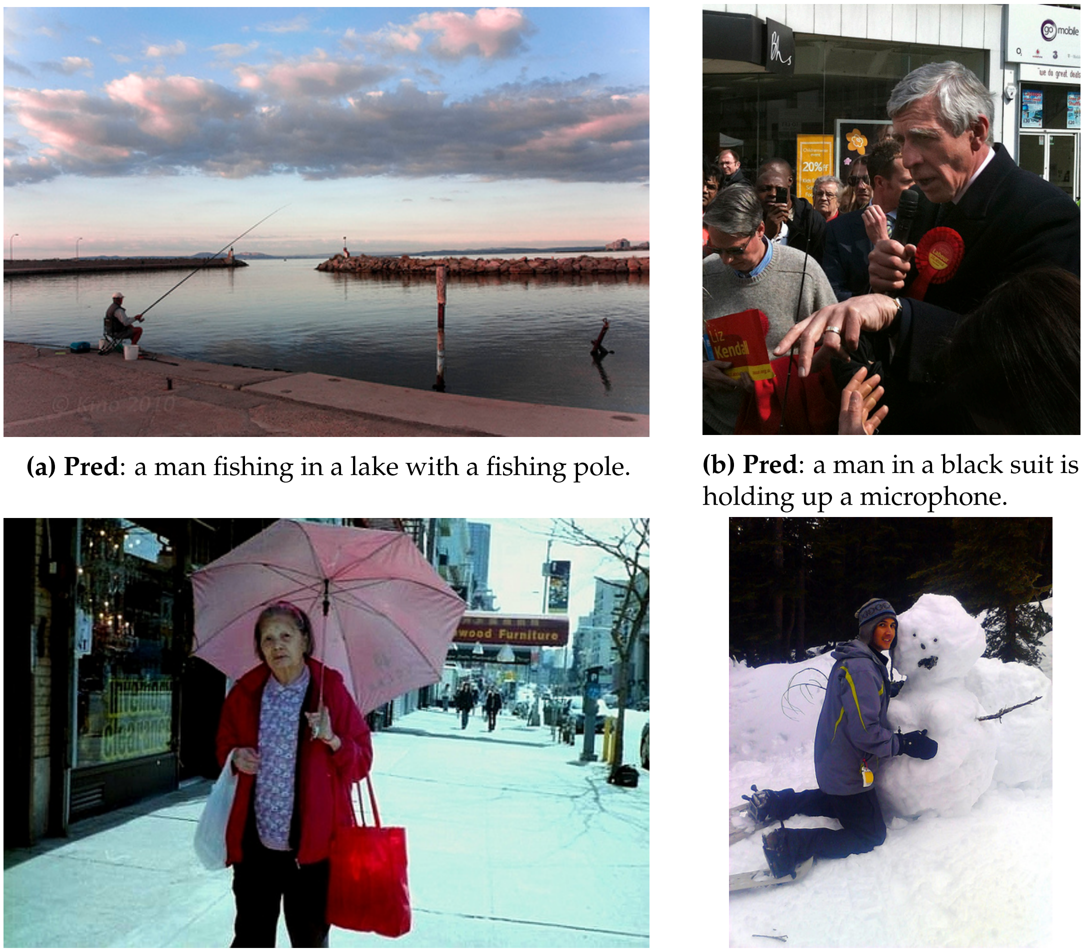
(a) Failure case: Although the man is localized, the bounding box is not tight enough. This is common because of the lack of bounding box annotations. Moreover, singular and plural forms of words are converted to the same representation during training and testing, leading to sub-optimal grounding of groups of objects.

(b) Kids and dogs are accurately localized.

(c) Note how the model seems to ground the words at the frames that they are visible.

(d) Failure case: the model fails to ground the correct racket and the man. Modeling the dependencies between the regions grounding each word in the sentence might help mitigate such issues.

Figure 3.15: Qualitative weakly-supervised object grounding results obtained on videos from the ActivityNet Entities validation set. For each groundable word in a ground-truth caption, we show the aligned region that is the mode of the approximate posterior distribution (region with the maximum q-attention network coefficient over all regions in the 10 equally-spaced frames).



(a) Pred: a man fishing in a lake with a fishing pole.

(b) Pred: a man in a black suit is holding up a microphone.

(c) Pred: a woman in a red coat is walking down the street holding a red umbrella.

(d) Pred: a person in a snow snowsuit is standing in the snow.

Figure 3.16: Generated image captions by our GVD-CVAE for images from the Flickr30k Entities validation set. Generated captions, although having a simple syntactic structure, tend to refer to objects in the image (fishing pole, umbrella, microphone, snowsuit, etc.).

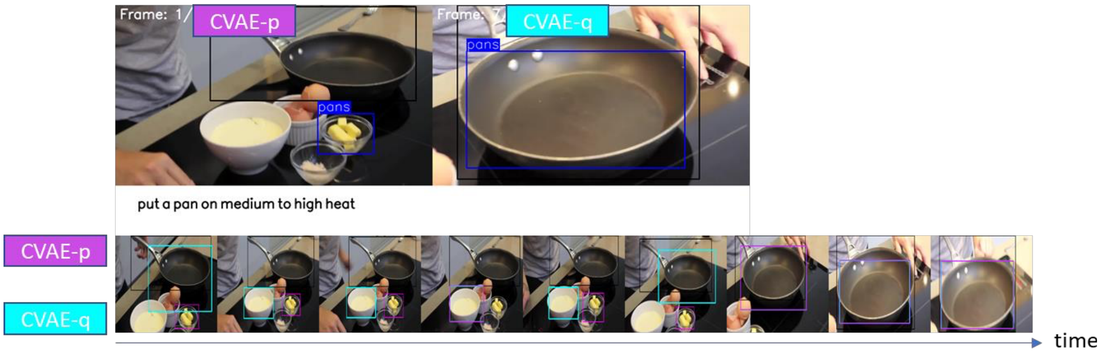


Figure 3.17: Qualitative weakly-supervised object grounding results in YouCook2. Both our approximate posterior and prior word-to-region alignments assume there is a single latent region out of all available regions in all video frames. When looking into this inferred region, the approximate posterior (CVAE-q) seems to be better able to ground each groundable word than the prior (CVAE-p). However, the regions with maximum q-attention (or p-attention) coefficients in each frame are not consistent. This could be the reason why CVAE-p happens to perform better than CVAE-q in this dataset. Ensuring grounded regions are consistent among frames (for example by finding region correspondences as a post-processing step) is an interesting avenue for future improvements of the GVD-CVAE model. Also note that this limitation is shared with the baseline GVD model, which also might not attend in regions of every frame to generate a caption word.

Chapter 4

Actor-Centric Deep Methods for Real-Time Activity Detection in Extended Videos

Chapters 2 and 3 focused on videos that are spatially-centered around a couple of actors involved in an event. In this chapter, we revisit the temporal activity detection problem from Chapter 2, but we relax the assumption of spatially-centered videos around a few, human adult actors. Our goal is to design a system for real-time detection of activities in videos that are *extended both in time and space* [12, 10], i.e., long untrimmed videos that capture multiple actors of various types (people, vehicles) performing multiple activities in various regions of indoor or outdoor scenes. We first detail the challenges associated with real-time activity detection in extended videos, motivating the need for decomposing the video into salient spatio-temporal sub-volumes and discuss prior work on extracting such sub-volumes. Then, we present our region-based framework for actor-centric activity detection in extended videos, by leveraging tracks of actor and object detections. Our framework consists of two stages: the first stage extracts spatio-temporal volumes of interest (*tubelets*) that are spatially-localized around an actor, while the second stage processes each one of those tubelets in parallel to temporally detect activities in them. Last, we present experimental results on the MEVA dataset [10] to validate our framework in terms of speed and performance.

4.1 Motivation and Overview

Powered by deep convolutional networks that process whole video frames and large datasets with rich human annotations, modern video understanding systems are capable of accurately detecting hundreds of human action classes in benchmark datasets [44, 252]. However, many of these popular datasets hide the inherent complexity of action recognition, by either focusing on trimmed videos with a single actor performing a single activity [252] or videos capturing activities performed by a few actors [174, 253, 254], occupying mostly foreground pixels. They also contain only activities performed by humans. The performance of state-of-the-art frameworks is indeed shown to degrade as (a) the number of actors in a scene increases [94], (b) their scale decreases [94], and (c) the complexity of activities increases [254]. Moreover, most activity recognition methods are not suitable for processing extended videos in real time. These limitations affect the ability to deploy these systems for real-time activity detection in extended videos containing a large number of actors (e.g., an average of around 30 actors) of varying types and scales, including tiny actors, performing multiple activities of varying durations [10].

Existing approaches for activity detection in extended videos narrow down the visual search space by identifying video sub-volumes, such as cuboids [255], action tubes [256], or actor tracks [257], that might contain activities. A cuboid is a sequence of bounding boxes with the same spatial coordinates, thus it can be used to crop a valid sub-video and can be fed as input to modern action recognition models. However, the rigid cuboid shape does not necessarily capture the versatile nature of actions. In contrast, action tubes are flexible spatio-temporal sub-volumes capturing relevant spatial contextual cues, but they are typically very short and fail to capture long-term temporal context. Actor tracks are ideal for capturing such temporal context, but might be impractical for real-time activity detection in extended videos for two reasons. First, in typical surveillance videos, such as videos of crowded parking lots, there is a large number of person and vehicle tracks. It is infeasible to process all these tracks under the real-time action recognition constraint. Second, it is not trivial to combine tracks in order to obtain the relevant visual context for detecting various types of activities, such as activities involving a single actor, interaction between actors or actor-object interactions.

In this chapter, we propose an *actor-centric framework* for real-time action detection of complex human and vehicle activities of varying spatio-temporal scales in extended surveillance videos. As illustrated in Figure 4.1, our framework is composed of two main modules: tubelet generation and temporal activity detection per tubelet.

The object-detection-based tubelet generation module decomposes an extended video into a collection of action-agnostic *actor-centric tubelets of interest*. Each actor-centric tubelet consists of an *actor tracklet* and a *context*

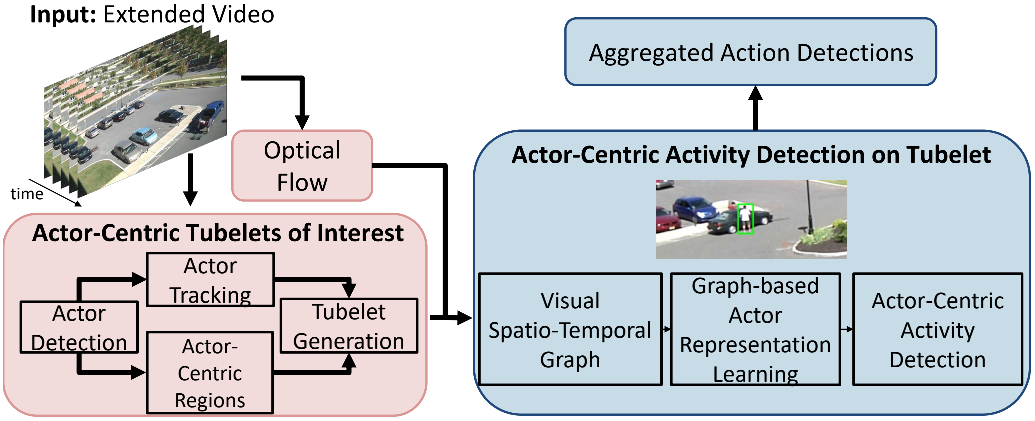

Input: Extended Video
time
Optical Flow
Actor-Centric Tubelets of Interest
Actor Detection
Actor Tracking
Actor-Centric Regions
Tubelet Generation
Aggregated Action Detections
Actor-Centric Activity Detection on Tubelet
Visual Spatio-Temporal Graph
Graph-based Actor Representation Learning
Actor-Centric Activity Detection

Figure 4.1: Overview of our proposed actor-centric framework for complex activity detection in extended videos. It consists of two main components: (a) actor-centric tubelet generation and (b) activity detection per tubelet. The first component generates spatio-temporal tubelets of interest, which are associated with a single primary actor (person or vehicle) and capture all the relevant spatio-temporal visual context (scene cues, interacting objects, etc.). The second component predicts the activities performed by an actor over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions. Details for their implementation are provided in Sec. 4.3.

tubelet. The former is a sequence of bounding boxes of variable size that contain the actor (human or vehicle), and the latter is a sequence of bounding boxes of constant size that captures adaptive, long-range spatio-temporal context for recognizing the activities of that actor. Overall, this module helps us localize activities in space on an actor-level and also reduces the number of regions that need to be processed in order to detect activities, reducing our overall processing time.

Each actor-centric tubelet is then passed to the second module, which detects the activities performed by an actor over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions. A popular approach for actor-centric action detection applies action classifiers on top of local actor features pooled from an intermediate feature map of a 3D CNN model [254, 95]. However, these local actor features do not capture the rich spatio-temporal interactions of the actor with other actors and objects within the tubelet. We model these interactions with a visual spatio-temporal graph, whose nodes correspond to detected actors and objects in the tubelet and whose edges encode different types of potential interactions, and obtain *context-aware actor features* by applying the Visual ST-MPNN [103] (introduced in Chapter 2) on this heterogeneous spatio-temporal graph. Our actor-centric activity detection module is trained only with actor-level supervision, without requiring annotations of relevant objects. Finally, activity detections from all tubelets are aggregated to generate the output set of activity detections for the input video.

In summary, the contributions of this work are three-fold. First, we introduce an *actor-centric framework* for real-time activity detection in extended security videos. Second, we propose an object-detection-based approach for generating action-agnostic actor-centric tubelets of interest that capture an adaptive spatio-temporal context for recognizing the activities of the corresponding actor. Third, we encode spatio-temporal actor-object interactions within each optical flow tubelet with a visual spatio-temporal graph and leverage state-of-the-art Graph Neural Networks [103] for obtaining context-aware, discriminative actor representations. We evaluate the proposed approach on the MEVA (Multiview Extended Video with Activities) dataset [10] and the ActEV21 Sequestered Data Leaderboard and obtain competitive activity detection results compared to published methods in terms of both speed and performance.

4.2 Related Work

Action Detection in Extended Videos. Most prior work on action detection focuses on long, untrimmed videos with activities performed by a few adult actors. Approaches that temporally detect activities by processing whole frames with convolutional networks, such as the RC3D [69], without determining spatio-temporal regions that might contain activities, have been shown not to be able to handle extended videos [10]. Thus, we focus our brief review of related work on approaches that first identify candidate spatial locations of activities. Activities are localized either frame-by-frame by leveraging person detections [95, 96, 94], or via spatio-temporal volumes, like short tubes [97, 98, 99, 100, 101] or tracks [102]. However, these approaches become impractical for detecting activities in extended surveillance videos, not only because they are not able to detect vehicle activities, but also because they will typically result in a large number of proposals, hurting run-time performance.

Detecting complex activities in extended, multi-person videos [12] is a more challenging and computationally demanding task, which requires narrowing down the visual search space by identifying regions that might contain activities. Our proposed approach, that leverages actor tracklets to spatially localize activities, is inspired by early work which tracked moving objects [258, 259] obtained by object detectors [257] or background subtraction [260, 261], and represented those tracks with hand-crafted, global representations. However, we lift simplifying assumptions of such prior works, such as activities being only human-vehicle interactions and a single activity happening in each region at a time [257], or videos being temporally pre-segmented [262]. Furthermore, we combine actor tracklets with tubelets [263], which allows us to capture adaptive, dynamic spatio-temporal context. Our work is also complementary to recent approaches that employ global deep representations of cuboids [255, 264, 265] or short tubes [256], and offers additional benefits, e.g., modeling of spatio-temporal interactions and long-term temporal context, as well as localization of the actors.

Interaction-based Region Representation Learning. Modeling spatio-temporal interactions between actors and objects has a long history in video understanding [266, 267, 128, 59]. However, most of prior work has focused on modeling interactions between regions with undirected graphical models in a discrete label space [110, 111, 268], where the regions were represented with hand-crafted features. Instead, the focus of our work is to leverage such interactions for learning context-aware actor representations (continuous features). Our activity detection model builds upon recently developed deep architectures called Graph Neural Networks (GNNs) [145], which enable representation learning on graph-structured data. Although GNNs have recently been applied to video understanding [113, 96, 142, 82, 117], they have not been explored for activity detection in extended videos. This chapter adapts the Visual ST-MPNN [103], which we introduced in Chapter 2 as a GNN tailored to representation learning on heterogeneous spatio-temporal graphs, to the task of actor-centric activity detection on tubelets and replaces appearance actor/object features with local motion features.

4.3 Actor-Centric Activity Detection

This section presents our proposed actor-centric framework for human and vehicle activity detection in extended videos. The overview of our framework is illustrated in Figure 4.1. An extended video is decomposed into basic units, called actor-centric tubelets of interest. Each tubelet is associated with an actor tracklet and ideally captures all the relevant spatio-temporal visual context (scene cues, interacting objects, etc.) for recognizing the activities of the actor. For the purposes of activity detection in extended surveillance videos, we consider humans and vehicles as actors, since the activities of interest include atomic human activities (e.g., *person closes facility door*), group human activities (e.g., *person embraces person*), human-vehicle interactions (e.g., *person closes trunk*) and atomic vehicle activities (e.g., *vehicle turning left*). Our action recognition module encodes the rich spatio-temporal visual context in spatio-temporal actor-object visual graphs and learns context-aware actor representations with the Spatio-Temporal Message Passing Neural Network (ST-MPNN). In the following, we first define the actor-centric tubelet. Then, we describe in details our approach for (a) actor-centric tubelet generation and (b) supervised temporal multi-label action recognition per tubelet. Finally, we discuss how to post-process the time series of action scores per tubelet in order to output final action detections in the input extended video.

4.3.1 Actor-Centric Tubelets of Interest

An actor-centric tubelet of interest is defined as a tuple of two bounding box sequences of the same temporal length: (a) an actor tracklet, i.e. a sequence of actor bounding boxes linked by an actor tracker, and (b) a context tubelet, i.e. a sequence of bounding boxes of constant height and width that contain the actor in addition to relevant spatial context. Formally, given an extended video with spatio-temporal dimensions (H, W, T), each actor-centric tubelet, denoted τ_i, is described as: $\tau_i = (t_s^i, t_e^i, \mathcal{B}_a^i, \mathcal{B}_c^i)$, where t_s^i is the start frame, t_e^i is the end frame, $\mathcal{B}_a^i$ is the actor tracklet, and $\mathcal{B}_c^i$ is the context tubelet. Both actor tracklet and context tubelet are sequences of bounding boxes of length $L = t_e - t_s + 1 \leq T$ denoted as $\mathcal{B}_a = [(x_0^a, y_0^a, w_0^a, h_0^a), \ldots, (x_L^a, y_L^a, w_L^a, h_L^a)]$ and $\mathcal{B}_c = [(x_0^c, y_0^c, w^c, h^c), \ldots, (x_L^c, y_L^c, w^c, h^c)]$, respectively, such that for each frame t the actor bounding box is included in the context bounding box and the context boxes have constant height and width, i.e.:

$$0 \leq x_t^c \leq x_t^a < x_t^a + w_t^a \leq x_t^c + w^c \leq W - 1, \tag{4.1a}$$

$$0 \leq y_t^c \leq y_t^a < y_t^a + h_t^a \leq y_t^c + h^c \leq H - 1. \tag{4.1b}$$

The actor-centric tubelet of interest has the following desirable properties: (1) it captures long-term temporal context of the actor's actions, since it is associated with an actor tracklet of arbitrary length, (2) it includes long-range spatial context, which complements the actor's appearance for recognizing the actor's activities (since each tubelet can have a different height and width), (3) it defines a valid sub-video with constant height and width, which can be fed to any modern backbone deep neural network for feature extraction, and (4) it can be annotated with unambiguous ground-truth activities at each timestep (given actor-level annotations). We should emphasize that our tubelet is not an action proposal, since it can be associated with zero or multiple actor activities.

Rather, it is a sub-volume of interest that is likely to contain activities and is focused on a single actor, similar to videos in most benchmark datasets.

Object-detection-based tubelet generation. Our actor-centric tubelet generation method filters out tracks that are not likely to contain an activity (such as parked vehicles) or are secondary to other actor tracks (such as vehicles involved in person-vehicle interactions for which the primary actor is the person). It also determines an adaptive spatial extent for each actor-centric tubelet based on interactions. It achieves this by relying only on object detections without requiring training with action spatio-temporal annotations. In particular, it consists of four stages: object detection, actor tracking, actor-centric region of interest extraction, and tubelet generation, as illustrated in Fig. 4.1.

– *Object Detection:* We initialize our tubelet generation pipeline by detecting objects (person or vehicle) per frame with the Faster R-CNN [175] off-the-shelf object detector, which was trained on the external MSCOCO [176] image dataset. Note that this object detector requires no supervision from action datasets, thus it may detect objects that are not performing an action.

– *Actor Tracking:* We track detections from each actor class (person or vehicle) using the SORT [269] off-the-shelf tracker, which predicts a trajectory using a Kalman filter and matches tracks to detections using a simple IoU metric. Tracking not only provides the basis for linking regions of interest across time to generate actor-centric tubelets, but also helps fill in missing object detections.

– *Actor-Centric Region of Interest Extraction:* The goal of this step is to (a) find actors at each frame that are likely to be involved in activities and (b) identify other actors they might be interacting with. This information will be used to filter out track segments that are not likely to contain activities, such as static vehicles without any people in their vicinity, thus reducing the number of regions fed to our activity detection module with minimal impact on the recall. It will also aid in determining the adaptive, spatial context that is relevant for recognizing the activities of each actor. We use a rule-based approach to find Regions of Interest (ROIs) per frame, where each region corresponds to one out of 5 potential types of ROIs and is associated with a primary actor detection. Such regions are automatically extracted from actor detections by associating them with hand-crafted rules based on scale-normalized distance thresholds, as described below:

1. Type 1: Any detected person is a region of interest of Type 1.
2. Type 2: Any moving vehicle is a region of interest of Type 2. We consider a vehicle to be moving if the scale-normalized distance of two consecutive bounding box centers $\text{dist}(b^v_{i,t}, b^v_{i,t-1})$ exceeds a threshold θ_{vd}:
$$\text{dist}(b^v_{i,t}, b^v_{i,t-1}) \geq \theta_{vd}. \tag{4.2}$$
3. Type 3: Any group of person detections that are close enough (their pairwise distance does not exceed a threshold θ_{pp} are considered to be a region of interest of Type 3:
$$\text{dist}(b^h_{i,t}, b^h_{j,t}) \leq \theta_{pp}. \tag{4.3}$$

The bounding box of this region of interest is the union box of the individual person bounding boxes.

4. Type 4: Any person-vehicle pair where the distance of the person and the vehicle does not exceed a threshold θ_{pv} and the vehicle is not moving is a region of interest of Type 4:

$$\mathrm{dist}(b^h_{i,t}, b^v_{j,t}) \leq \theta_{pv}, b^h_{i,t} \notin \text{Type2} \tag{4.4}$$

   The bounding box of this region of the interest is the union box of the person and vehicle bounding boxes.

5. Type 5: Any vehicle that has been recently active, i.e., vehicles that were (will be) moving or were (will be) associated with an actor detection within a look-back (look-ahead) window of K frames, is considered to be a region of interest of Type 5. This rule aims to cover cases where neither the person is visible, nor the vehicle is moving, but an activity such as *exiting the vehicle* might be occurring:

$$b^v_{i,t} \notin \text{Type2} \cup \text{Type4}, \tag{4.5}$$

$$\exists k \in [-K, K] \text{ s.t. } b^v_{i,t+k} \in \text{Type2} \cup \text{Type4}. \tag{4.6}$$

An intuitive illustration of the five types of actor-centric RoIs and their corresponding primary actors, as well as the rules used for their construction, is shown in Figure 4.2. Note that an actor detection can be the primary actor of zero, one or multiple actor-centric RoIs. For example, a person can be associated with multiple nearby people and vehicles. We would like to

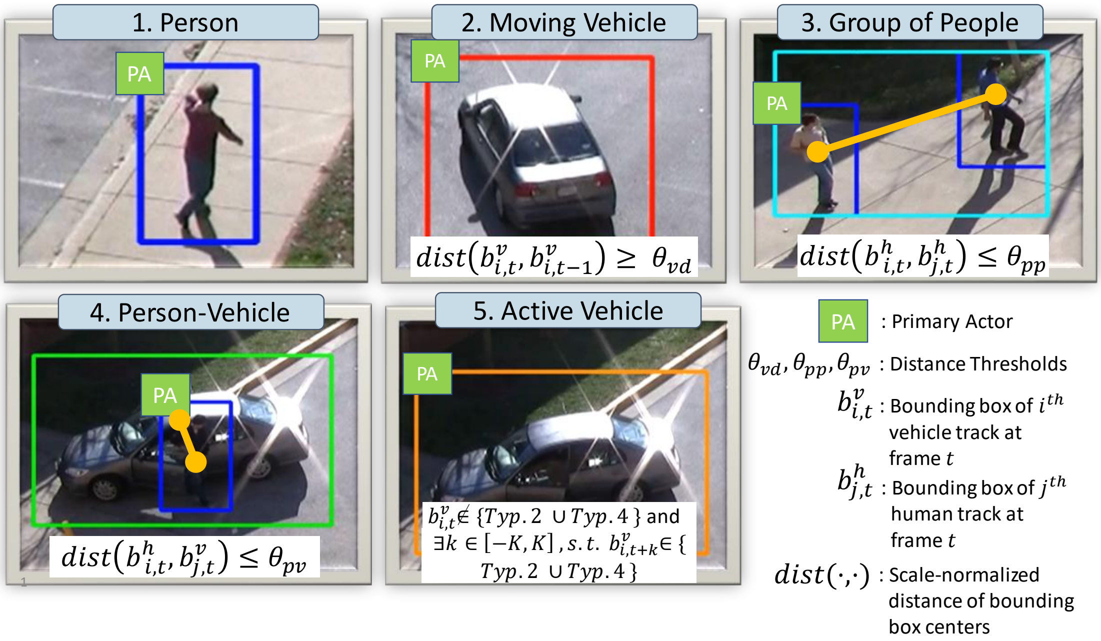

1. Person
PA
2. Moving Vehicle
PA
$dist(b^v_{i,t}, b^v_{i,t-1}) \geq \theta_{vd}$
3. Group of People
PA
$dist(b^h_{i,t}, b^h_{j,t}) \leq \theta_{pp}$
4. Person-Vehicle
PA
$dist(b^h_{i,t}, b^v_{j,t}) \leq \theta_{pv}$
5. Active Vehicle
PA
$b^v_{i,t} \notin \{Typ.2 \cup Typ.4\}$ and $\exists k \in [-K, K], s.t.\ b^v_{i,t+k} \in \{Typ.2 \cup Typ.4\}$
PA : Primary Actor
$\theta_{vd}, \theta_{pp}, \theta_{pv}$: Distance Thresholds
$b^v_{i,t}$: Bounding box of i^{th} vehicle track at frame t
$b^h_{j,t}$: Bounding box of j^{th} human track at frame t
$dist(\cdot,\cdot)$: Scale-normalized distance of bounding box centers

Figure 4.2: Types of regions of interest with their associated primary actors for sample frames from the VIRAT [12] dataset. For example, regions of Type 5 correspond to recently active vehicles, i.e., vehicles that were (will be) moving or were (will be) associated with an actor detection within a look-back (look-ahead) window of K frames.

emphasize some limitations of these hand-crafted rules. The *moving vehicle definition* assumes a static camera, while the *group of people definition* results in overlapping regions of interest, one for each person.

– *Tubelet Generation:* Given the actor-centric RoIs extracted per frame, we are now ready to describe the generation of actor-centric tubelets. First, we construct a context bounding box for each actor detection that is the primary actor of at least one actor-centric RoI. This context box is constructed by computing the union of all RoIs which have this actor as their primary actor. Leveraging the extracted actor tracks, context bounding boxes associated with the same primary actor instance are linked over time to construct an actor-centric tubelet of interest, with the sequence of context boxes generating the *context tubelet* $\mathcal{B}_c$, and the sequence of primary actor bounding boxes generating the *actor tracklet* $\mathcal{B}_a$. We would like to emphasize that in contrast to track-based methods, our actor-centric tubelets do not necessarily include a whole actor track, but only track segments that contain actor detections that are primary actors of RoIs. For example, instead of predicting activities for each timestep of a tracked vehicle, we only predict activities for the temporal segments that this vehicle is either moving or people are about to enter/exit it. Still, all detections of this vehicle can serve as context for other tubelets.

Context tubelet post-processing. The generated context tubelets might have an irregular shape with sudden changes in the size of the consecutive bounding boxes, e.g., because the number of interacting actors varies with time or because of errors in the association of actors due to occlusions. To alleviate this issue, we enlarge each context bounding box of the tubelet so that they have the same height and width, with its dimension being determined by the largest bounding box of the tubelet. A final refinement step ensures that the tubelet consists of a smoother sequence of context bounding boxes. In particular, a Savitzky-Golay [270] filter is used to estimate smoothed values of the bounding box centers. Then, the top-left context bounding box coordinates are updated accordingly without modifying the tubelet dimensions.

4.3.2 Actor-Centric Activity Detection on Tubelet

Once an extended video is decomposed into a set of actor-centric tubelets of interest, our system seeks to temporally detect the activities performed by the actor of each tubelet. Our proposed structured activity detection module, illustrated in Figure 4.3, builds upon the VS-ST-MPNN (V-HetGAT variant) that we presented in Chapter 2. It encodes spatio-temporal interactions between actors and objects in a visual graph and learns graph-structure-aware actor embeddings that can be used to recognize activities.

Visual spatio-temporal graph. Let $\tau = (t_s, t_e, \mathcal{B}_a, \mathcal{B}_c)$ be an extracted tubelet with length $L = t_s - t_e + 1$. We represent it with a visual spatio-temporal, attributed graph $G^v = (\mathcal{V}^v, \mathcal{E}^v)$, which consists of a set $\mathcal{V}^v$ of actor nodes and object nodes, and a set of visual edges $\mathcal{E}^v$. Actor nodes correspond to the bounding boxes of the primary actor tracklet $\mathcal{B}_a$ of the tubelet, while object nodes correspond to other object detections within the context tubelet $\mathcal{B}_c$, including other visible humans and vehicles. The graph is built by adding directed, typed edges that connect nodes. In particular, an edge connecting node j to node i is associated with an edge type $r(i, j)$. We use three edge types: *object-to-actor spatial* ($r(i, j) = 0$) and *actor-to-object spatial* ($r(i, j) = 1$) edges connect actor and object nodes in the same frame, while *actor-to-actor temporal* ($r(i, j) = 2$) edges connect actors across frames. We constrain temporal edges to connect only nodes of the same type between consecutive frames. We initialize graph node and edge attributes using the same procedure that we described for instantiating the visual subgraph on regular videos for temporal activity detection in Section 2.5.2.3. The only difference is that for the case of extended videos we chose to represent videos with optical flow features. Our choice was motivated by the experimental observations of Gleason et al. [264], who demonstrated that using optical flow features for activity detection in surveillance videos, outperforms RGB features, potentially because optical flow features capture better salient motion cues even in low-resolution areas. All graph node attributes $\mathbf{h}_i^{(0)}$ are initialized with ROI-pooled features from a feature map that is obtained by passing a cropped optical flow tubelet through a flow I3D network [44]. Similarly, edge attributes $\mathbf{q}_{ij}^{(0)}$ are initialized with the relative spatial location of the connected nodes.

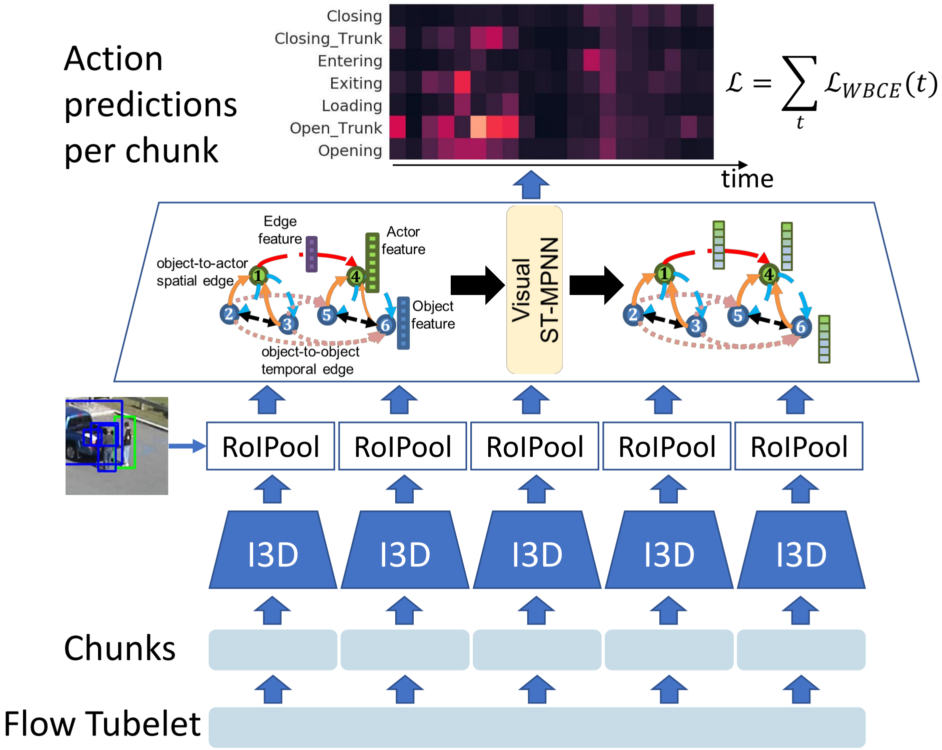


Figure 4.3: Graph-based activity detection module for actor-centric activity detection in a tubelet.

Graph-based actor representation learning. Given the input visual st-graph, our Visual Context Module iteratively refines the local node and edge features with spatio-temporal contextual cues. As we described in detail in Section 2.3, at each iteration l, we (1) compute scalar visual edge weights using edge-type-specific attention mechanisms; (2) compute a message along each edge using the attention-based scalar edge weight, the current embeddings of the connected nodes and the edge; (3) update the embedding of every node by aggregating messages from incoming edges with an update function; and (4) update the attribute of every edge by using the message that was computed alongside it. Importantly, the message passing functions are parameterized with learnable weights that depend on the edge type. After L^v layers of the spatio-temporal MPNN (or equivalently L^v rounds of node and edge updates), we obtain refined, visual context-aware node and edge features.

Temporal activity detection. Let $\mathbf{x}_t$ be the context-aware node feature that corresponds to the tubelet's primary actor bounding box at time t. A linear classifier is applied on $\mathbf{x}_t$ to predict scores for C action classes at time t:

$$\tilde{\mathbf{y}}_t = W_{cls}\mathbf{x}_t + \mathbf{b}_{cls} \in \mathbb{R}^C, \quad t = 1, ..., L, \tag{4.7}$$

where $W_{cls} \in \mathbb{R}^{C \times d}$ and $\mathbf{b}_{cls} \in \mathbb{R}^C$ are learnable parameters. Since an actor might be performing multiple activities at the same time, we treat the problem as a multi-label per-frame action classification problem, passing scores $\tilde{\mathbf{y}}_t$ through a sigmoid activation function to yield the predicted action probabilities $\hat{\mathbf{y}}_t \in [0, 1]^C$.

The output of the previous step is a sequence of probabilities for each activity $a \in \{0, \ldots, C-1\}$ for each tubelet timestep t. To obtain final temporal detections for an activity a within the tubelet, we need to convert the action scores sequence to a set of temporal segments with start, end times and associated confidence scores. To achieve this, we first apply a median filter to the predicted probabilities for activity a $[\hat{y}^a_0, \ldots, \hat{y}^a_{L-1}]$ to obtain the smoothed activity probabilities $[s^a_0, \ldots, s^a_{L-1}]$. We then initialize activity detections at the local maxima of the smoothed action score time-series $[s^a_0, \ldots, s^a_{L-1}]$. The temporal boundaries of an activity detected at local maximum location t_k, with score $s^a_{t_k}$, are extended by including previous and future timesteps until their action score falls below a relative threshold $\theta \cdot s^a_{t_k}$, where $\theta < 1$ is a hyperparameter. In this way, we can detect activities of arbitrary lengths and we can handle several instances of the same activity performed by the tubelet's primary actor, such as consecutive *turning left* activities corresponding to the same vehicle tracklet.

Training. Our actor-centric activity detection module is trained with actor-level annotations associated with the primary actor of each tubelet. Given the ground-truth activity annotations for the primary actor of a tubelet, the ST-MPNN network is trained jointly with the action classifiers by using a Weighted Binary Cross-Entropy (WBCE) loss per class:

$$\mathcal{L}_{WBCE}(y^a_t, \hat{y}^a_t) = \beta_a y^a_t \log \hat{y}^a_t + (1 - y^a_t) \log(1 - \hat{y}^a_t), \tag{4.8}$$

where $y^a_t \in \{0, 1\}$ is the ground-truth label for timestep t and action a, and $\hat{y}^a_t \in [0, 1]$ is our model's prediction. To handle the class imbalance, we apply a weighting factor β_a to positive examples of each class a, which is determined based on the inverse class frequency.

4.4 Experiments

4.4.1 Datasets

We validate our method on the MEVA dataset and the ActEV 2021 Sequestered Data Leaderboard. The **MEVA dataset** [10] consists of 5-minute videos capturing indoor and outdoor scenes. There is an ongoing effort for annotating MEVA videos with actor-level annotations of 37 activity classes by Kitware and the community. We use Kitware annotations[1] for 784 of these videos for training our activity detection module and 172 for constructing an internal validation set for our ablation studies. The **ActEV 2021 SDL**[2] consists of sequestered surveillance videos, which are not publicly available. Evaluating a method on this dataset requires submitting an activity recognition system that is compatible with the ActEV Command Line Interface (CLI) protocol and temporally detects instances of 37 activites. The submitted system is then executed on test servers provided by NIST and scores are reported on the public leaderboard.

4.4.2 Metrics

The activity detection performance of our system is evaluated with the official metrics of the ActEV SDL evaluation: (a) the probability of missed detection at fixed time-based false alarm per minute (Pmiss@0.02tfa), and partial area under the Detection Error Tradeoff curve (nAUDC@0.2tfa). These metrics are calculated by finding correspondences between system activity detections

[1] https://gitlab.kitware.com/meva/meva-data-repo/-/tree/master/annotation/DIVA-phase-2/MEVA/kitware-meva-training

[2] https://actev.nist.gov/sdl

and ground-truth activity instances via the Hungarian algorithm. A ground-truth activity instance is considered to be missed if it does not overlap with a system detection for at least one second. The time-based false alarm is the proportion of time the system detected an activity when there was none. Temporally fragmented activity detections that occur during the duration of a ground-truth activity instance do not increase T_{fa} unless the detections overlap temporally. For details on the computation of these metrics, we refer the reader to the official evaluation plan [271].

For achieving a good performance under these metrics, our system needs to accurately detect activities, while at the same time it needs to minimize the Time-based False Alarm. We used the official scorer[3] for evaluating the system on our internal MEVA validation set. Metrics were computed per video and we report their average.

4.4.3 Implementation Details

Tubelet generation. Our actor detections correspond to Person and Vehicle (bicycle, car, motorcycle, bus, truck) object detections with confidence score above 0.5. The SORT tracker [269] is used to separately track people and vehicles. Tracks are terminated after not being associated with an actor detection for 64 frames. Afterwards, regions of interest are identified in each frame by associating actor detections with hand-crafted rules, which are based on thresholds of scale-normalized distances, as introduced in Fig. 4.2: $\theta_{pp} = 6000$, $\theta_{pv} = 5000$, $\theta_{vv} = 500$, and an active vehicle look-ahead/look-back window

[3] https://github.com/usnistgov/ActEV_Scorer

of $K = 256$ frames.

Activity detection module. For activity detection on each tubelet, we first crop the tubelets from an optical flow representation of the input extended video. Optical flow is extracted from resized and downsampled RGB frames with the TVL1 algorithm following the same setup as in [264]. To build the visual graph, we first apply an optical flow I3D network [44], which was trained for action classification on MEVA cuboids and shared by the authors of [264], on consecutive 2-second non-overlapping chunks of the input flow tubelet. In this way, we obtain a feature map with a temporal stride of 8 frames for each chunk. We then instantiate the graph on top of the primary actor detections and 10 most confident object detections (with score above 0.1) at the corresponding tubelet frames. Note that we store the centre coordinates of all object detections for a frame of the original extended video in a KD-tree data structure, which enables efficient rectangle range queries. We can then efficiently retrieve all object detections whose centre lies within a tubelet bounding box at a given frame. The initial node features for actors/objects are pooled from the Mixed 4f 3D feature map of the flow I3D for each detected region using RoIAlign [175]. These features are refined to include context by performing 3 rounds of node/edge refinement with the Visual ST-MPNN [103], resulting in context-aware 512-dimensional embeddings of actor regions that are fed to action classifiers. The action detection threshold θ is set to 0.8 and median window size is 25 frames (3 chunks).

Training. We jointly train the Visual ST-MPNN and action classifiers on 7151 tubelets extracted from MEVA training videos for 150 epochs using a batch size of 10 tubelets (with a maximum length of 30 seconds). Given ground-truth actor-level annotations, we assign a ground-truth activity to the primary actor of a tubelet at a given frame if its detected bounding box overlaps with the corresponding ground-truth actor with $IoU > 0.5$. We use the Adam [171] optimizer, with an initial learning of $1e^{-4}$.

CLI system. The system submitted to the ActEV SDL is customized to run on a hardware consisting of 4 GPUs with 128GB RAM. It is implemented as a pipeline consisting of several stages with each stage producing an output to be used by the later stages. The stages can be enumerated as follows:

1. Optical Flow Extraction,
2. Object Detection and Actor Tracking,
3. Tubelet Generation,
4. I3D Feature Extraction, and
5. ST-MPNN Processing.

Each stage is parallelizable and spawns several subprocesses/workers which work on multiple videos/chunks simultaneously. Among the stages, stage 3 is CPU-intensive and the rest are GPU-intensive. The pipeline processes the entire test set in batches of 96 videos. Each stage maintains a processing queue of 96 videos and any idle workers consume videos from this queue until the entire video batch has been processed. The number of workers for each of the stages are: 48, 24, 96, 8, and 8 respectively. In all stages except stage-3, we are limited by the GPU memory and hence cannot increase the

Table 4.1: Temporal detection results on the ActEV 2021 Known Facility SDL as of November 1st 2021. We report the Pmiss@0.02tfa and nAUDC@0.2tfa metrics. Lower nAUDC and pmiss values indicate a superior performance since they are related to missing an activity.

System	nAUDC	pmiss@0.02tfa	Rel. Time
Cuboids [264]	0.476	-	0.725
Gabriella [256]	0.438	-	0.362
Dense Prop. [272]	0.423	-	-
CMU-DIVA	**0.163**	**0.3424**	0.413
UCF	0.232	0.3793	0.751
UMD	0.262	0.4544	0.380
IBM-Purdue	0.281	0.4942	0.631
Visym Labs	0.283	0.4620	0.721
UMD-Columbia	0.305	0.4716	0.516
UMCMU	0.323	0.5297	0.464
Purdue	0.332	0.5853	0.131
BUPT-MCPRL	0.799	0.9281	**0.123**
MINDS_JHU (Ours)	0.483	0.6649	0.967

number of workers anymore. The system submitted to the ActEV SDL slightly differs from the system evaluated on our internal validation set: (a) the object detector is applied on the video with a stride of 4 frames for faster processing, while repeating the bounding box detections in between to accommodate for the skipped frames, and (b) we keep at most 200 actor-centric tubelets from each input video, after ranking them based on motionness cues.

4.4.4 Experimental Results

Comparison with the state of the art. Table 4.1 compares the activity detection performance of our method with recently published work and other submitted

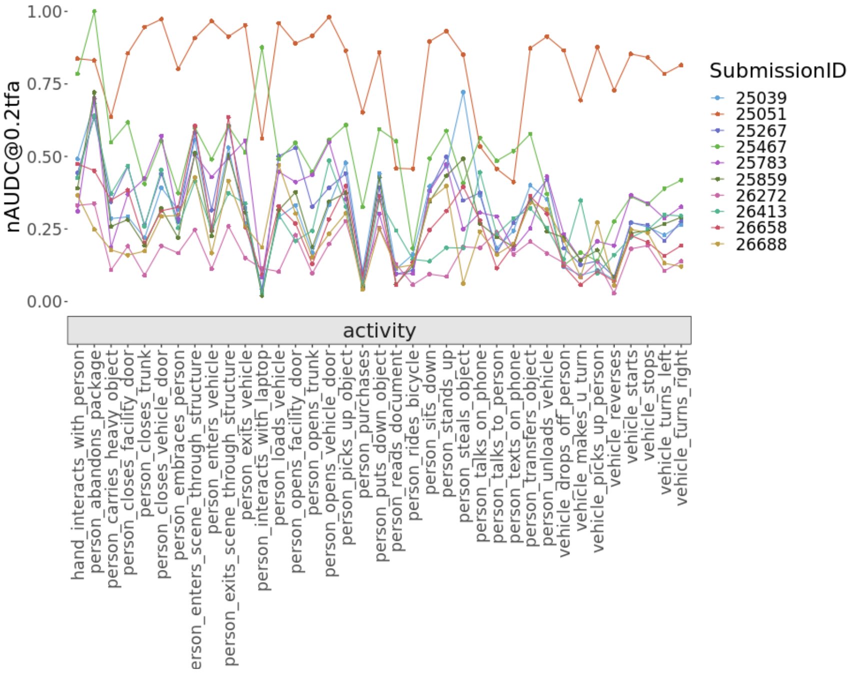


Figure 4.4: Per-class nAUDC scores for systems on the ActEV 2021 SDL. Our system ID is 25467 (light green).

systems on the ActEV 2021 SDL Known Facility Leaderboard[4] as of November 2021. Our actor-centric framework for real-time activity detection achieves activity detection performance that is close to other published methods [256, 264, 272] (rows 1-3). Notably, it achieves this metric despite only training the GNN and action classifiers of our framework using actor-level annotations, in under 3 hours using a single Titan XP GPU (given the extracted visual graph), while relying on off-the-shelf, pretrained networks for object detection and flow feature extraction. When compared to system submissions on the ActEV SDL Challenge, which might utilize additional training datasets, end-to-end training, and model ensembles, our system lags behind most of them. However, as we can see in Fig. 4.4, our system (ID: 25467) performs on par

[4] https://actev.nist.gov/sdl#tab_leaderboard

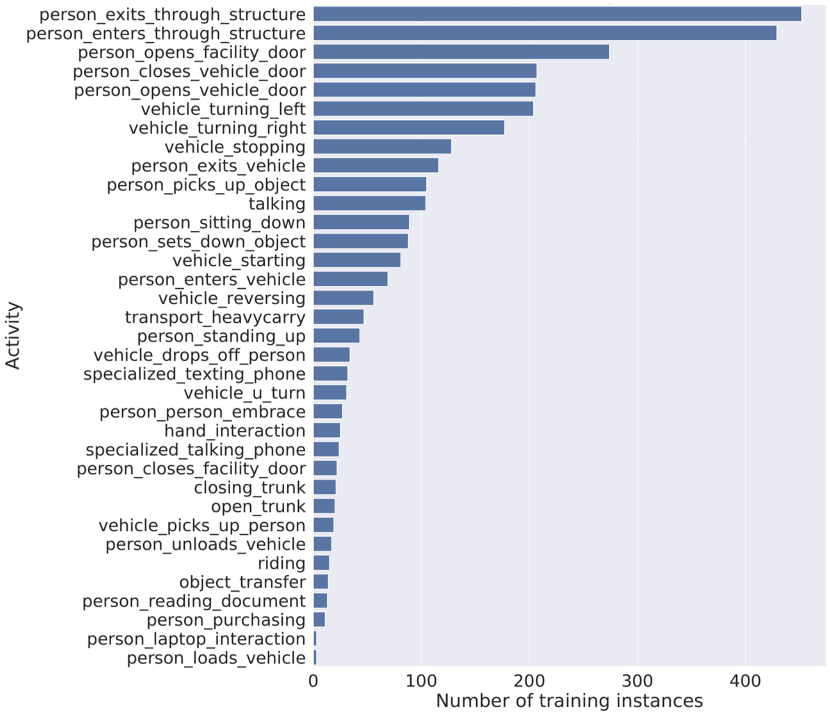


Figure 4.5: Number of training instances per activity used to train our system submitted to the ActEV 2021 SDL.

with other methods for a wide range of activities, such as person-vehicle interactions (*vehicle drops-off person*) and vehicle activities (*vehicle u-turn*), while performing significantly worse on *person abandons package* and *person interacts with laptop*, for which we used only a few annotated instances, as illustrated in Fig. 4.5. Our overall performance could be improved by including more samples of these activities in our training set and by fine-tuning our object detector on surveillance data. Furthermore, the I3D could be fine-tuned jointly with the ST-MPNN.

Ablation analysis. We now discuss a variety of ablation studies of different components of our framework. In Table 4.2, we compare the total number of actor regions that are included in actor tracks with the number of regions that

Table 4.2: Impact of tubelet generation method on the number of actor regions that are fed to the activity detection module. *Tracks*: baseline tubelets spanning each actor track of an extended video. *ACToIs*: our proposed Actor-Centric Tubelets of Interest. Results are reported on our internal validation set of Kitware-annotated MEVA videos.

Tubelet type	Nb. Actor RoIs
Tracks	6783972
ACToIs	**2553404**

Table 4.3: Activity recall of our proposed Actor-Centric Tubelets of Interest on our MEVA training and validation sets. Recall $R@T$ is computed by considering an activity instance as retrieved when at least one tubelet's primary actor overlaps with the ground-truth actor with IoU > 0.5 for at least T consecutive frames.

	$R@30$	$R@8$	$R@1$
ACToIs (train)	69.0	81.7	85.0
ACToIs (val)	67.9	81.5	84.0

are the primary actors of our actor-centric tubelets. As we can see, our tubelet generation method prunes a large number of tracked actor detections that are unlikely to contain activities and only feeds 37% of the actor regions to the activity detection module. This helps our model perform real-time activity detection. Despite pruning a large number of actor regions, our generated tubelets retrieve a large number of ground-truth activities (around 80%), as shown in Table 4.3. The primary cause for missed activity detections are object detection failures of the off-the-shelf, pretrained object detector. In Table 4.4, we first experiment with two different action classification models to determine the best architecture for our system. In particular, we compare our context-aware feature obtained by applying the Visual ST-MPNN on our visual graph with a baseline feature that is obtained from locally-extracted

Table 4.4: Ablation experiments on our internal MEVA validation set for different design choices of the activity detection module. *Local Actor Feat*: baseline approach that recognizes actor activities based on locally-extracted actor features. *Context-aware Actor Feat*: our proposed approach that learns context-aware actor features with the Visual ST-MPNN. *Dynamic Duration*: generating activity detections of varying durations. *Fixed Duration*: generating activity detections of fixed duration (6 sec) around local maxima.

Feat	Duration	nAUDC	pmiss@0.04	pmiss@0.02
Local	Dynamic	0.558	0.556	0.680
Context-Aware	Dynamic	0.531	0.501	0.663
Context-Aware	Fixed	**0.492**	**0.469**	**0.565**

actor features passed through a trainable two-layer Multi-layer Perceptron of hidden size 1024. Refining the local actor features with the GNN improves performance, verifying our intuition that spatio-temporal actor-object interactions are crucial for detecting activities. Furthermore, we compare generating activity detections of fixed duration (6 seconds) around each local maximum of the score time-series per activity class, instead of adaptively extending the detection to the past and future by a relative score threshold. Surprisingly, fixed duration activity detections lead to a better performance. This can be attributed to the employed detection metrics, which consider a ground-truth activity to be detected as long as it has an overlap of 1 second with a system detection.

4.5 Conclusion

In this chapter, we have introduced an actor-centric framework for detecting complex human and vehicle activities of varying spatio-temporal scales in extended surveillance videos. The basic idea was to decompose an extended video into a collection of actor-centric tubelets of interest, which capture long-range spatial and temporal context for an actor. Then, the problem was reducted to processing each tubelet and predicting its primary actor's activities over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions within the tubelet. We have validated a proof-of-concept of that system on the MEVA dataset, including the external, sequestered Known Facility leaderboard, where our system performed competitively with previously published methods, and yielded promising results for numerous activities compared to other systems submitted at the ActEV SDL'21 challenge. Last, our system has been designed in a modular fashion, making it amenable to improvements. For instance, the current off-the-shelf object detection, tracking and feature extraction backbones can be easily replaced by state-of-the-art networks, such as DETR [192], Joint Detection and Embedding (JDE) multiple-object tracker [273], and TANet [274], respectively.

Chapter 5

Conclusion and Future Work

The goal of this thesis was to leverage actors, objects and their interactions, in order to develop novel technology that can automate the detection of complex activities and associated interacting entities in extended videos.

We approached this problem of designing such a computer vision system with a "divide-and-conquer" strategy. Our main intuition was that it should be easier for the computer vision algorithm to understand the activities that occur in an extended video if it focuses on smaller and shorter subvideos, such as a spatio-temporal subvolume that is centered around a couple of interacting people and vehicles. Hence, we proposed dividing the original extended video in such smaller subvideos and analyzing each one of them in parallel. With that strategy in mind, we focused on the following video understanding tasks: (a) temporal activity detection in untrimmed videos that are spatially-centered around a few actors, (b) grounded visual description and visual object grounding given video segments corresponding to an event, and (c) activity detection in extended videos.

In Chapter 2 we focused on the task of temporal activity detection, which we addressed by learning region-based frame representations that are used to predict the activities at each timestep. We proposed novel heterogeneous Message Passing Neural Networks, composed of a Visual Context Module and a Semantic Context Module, for representation learning on heterogeneous graphs, which encode visual and semantic interactions among actor and object regions in a video. The key idea was to learn different message functions for different edge types, and to take into account edge attributes, such as relative geometric relations between regions, in order to compute context-aware region embeddings. Experimental evaluation showed that by jointly learning these region embeddings with activity recognition networks, e.g., classifiers, our framework outperformed baselines using local region features or contextual embeddings obtained by Graph Convolutional Neural Networks applied on a homogeneous visual graph. Our proposed VS-ST-MPNN model also improved upon prior Graph Neural Networks in terms of sub-activity detection performance on the CAD-120 dataset, setting a new state of the art. Moreover, combining region-based activity predictions with predictions based on holistic, clip-level RGB features led to state-of-the-art temporal activity detection performance on the Charades dataset, significantly outperforming two-stream, holistic approaches that utilize both RGB and Optical Flow inputs. Furthermore, qualitative analysis showed that the learned model automatically attends to relevant contextual objects when aggregating relevant visual context for activity prediction at each timestep, and that richer interactions (e.g., more types of interactions or long-term interactions) encoded in the input graph lead to better embeddings learned by our model.

In Chapter 3, we moved beyond predicting a label from a limited set of predefined activity classes for each activity that is temporally detected in a video. Instead, our goal was to joinly describe temporally localized events with a natural language description - which conveys richer details about the visual content - as well as localize each referred semantic entity in the video input. We explored various models to tackle this task with varying levels of supervision. In the first half of the chapter, we introduced an extension of the fully-supervised GVD model originally proposed by Zhou et al. [4] for Grounded Visual Description and Visual Object Grounding. Our proposed AO-GVD model utilizes region embeddings that are aware of local spatio-temporal actor-object interactions and global semantic interactions. To achieve this, we adapted the heterogeneous message passing modules from Chapter 2. We evaluated this model on the ActivityNet Entities video dataset, and demonstrated the benefits of combining multi-head self-attention (Transformers) with our Semantic Context Module. In the second half of the chapter, we explored the more challenging problem of learning how to align words in sentences with visual regions based only on video-caption pairs and made advances in the weakly-supervised visual object grounding and grounded visual description tasks. We proposed a novel grounded visual description CVAE for weakly-supervised learning of word-to-region alignments. We designed a variety of deep conditional generative models of sentences and explored the impact of architectures and conditional independence assumptions on the grounding performance. To demonstrate the generality and effectiveness of our model we evaluated it on two tasks and both image and video datasets. We showed that leveraging the learned latent alignment distributions (both the *prior* word-to-region alignment distribution that only looks at previous words, and the *approximate posterior* distribution that looks at the full sentence) of our model significantly outperforms soft attention for grounding given ground-truth or generated sentences. Our novel approach also yielded competitive results under multiple metrics in both grounding and grounded video description while comparing against methods optimized for one of the two tasks.

In Chapter 4, we relaxed the assumption that videos are spatially-centered around a few main actors, and tackled the activity detection problem in extended surveillance videos. We introduced an actor-centric framework for detecting complex human and vehicle activities of varying spatio-temporal scales in extended surveillance videos. The basic idea was to decompose an extended video into a collection of actor-centric tubelets of interest, which capture long-range spatial and temporal context for an actor. Then, the problem was reduced to processing each tubelet and predicting its primary actor's activities over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions within the tubelet. We evaluated our system on the challenging MEVA dataset, and we were able to detect activities in real-time, yielding promising results, despite the simple design and training of our system.

To conclude this thesis, we now discuss some directions for future research motivated by limitations that remain despite our advances in region-based video understanding.

Moving beyond 2D actor and object bounding boxes. Although the graph-based representation learning approaches and word-to-region alignment models that we proposed in this thesis are agnostic to the type of input regions, throughout this thesis we assumed that regions correspond to 2D bounding boxes of actors and objects. In other words, we represented videos in terms of 2D region proposals, and grounded words to bounding boxes corresponding to semantic entities. While this was a reasonable decomposition for the types of videos considered in this thesis (capturing daily indoor activities or outdoor scenes), there are many cases where object region proposals might not be the best semantic units. First, the performance of our region-based approaches is limited by the quality of detected regions. For instance, object detectors might fail to detect partially-occluded objects, which are commonly encountered in crowded scenes, or unfamiliar objects that were not seen during training, such as various cooking ingredients and types of household objects. In other cases, videos might only capture a single person and the goal might be to recognize their gestures or facial expressions. Decomposing a video in terms of actors and objects would not be helpful in this case.

In these settings, it would be advantageous to decompose videos in terms of more fine-grained semantic units, such as: (a) object parts (e.g., body parts or vehicle doors/wheels etc.) [49], or (b) class-agnostic semantic segments [275]. For instance, recent works in the image domain have started exploring the dense grounding of object words and phrases to semantic segments of the image, in a task called *Panoptic Narrative Grounding* [276]. An interesting direction for better handling occlusions and modeling geometric relationships among objects, would be to leverage recent advances that extract 3D bounding box proposals from 3D point clouds [277] and build visual spatio-temporal graphs on top of these 3D structures.

Joint activity detection and grounding by language specification. In this thesis, we detected instances of activities from a closed-set of activity labels. In real applications, one might be interested in detecting particular events, such as detecting all instances of a "man wearing a blue hat enters a building" and localizing the referred actors and objects. This would remove the need to construct a fixed vocabulary of activity labels a priori, and it would allow computer vision systems to recognize activities that they have never seen before. Building upon the insights developed in this thesis, one can envision a *coarse-to-fine strategy* for tackling this task: (a) first select which tubelet (spatio-temporal subvolume) best matches the textual description of the desired event, and (b) then ground the referred semantic entities within the tubelet with the latent-variable sentence models that we developed in this thesis. To address the first step, i.e., matching tubelets with sentences, one can leverage state-of-the-art coarse-level image-to-text (or video-to-text) matching frameworks, such as CLIP [278]. These frameworks typically pretrain transformer models with contrastive objectives using a large amount of image-caption pairs, in order to learn a joint embedding space for sentences and visual inputs.

Domain generalization. In the experiments conducted in this thesis, we assumed that the test videos are drawn from the same distribution of the training set. Although one would expect computer vision systems to be able to learn the semantics of each activity and be able to recognize it in any setting, such as in surveillance videos, videos captured by hand-held devices, or animated movies, this is not the case for most state-of-the-art systems including ours. The reason is that there exists a large distribution shift across these domains on a pixel level. Even training with surveillance videos from a particular facility (collection of buildings and scenes) and testing on videos from a different facility [279] significantly degrades our activity detection system's performance compared to training and testing on videos from the same facility (e.g., the nAUDC@0.2tfa metric, related to the probability of missing an activity instance, increases from 48% to 63%). Addressing this so-called unsupervised domain adaptation problem, would also allow researchers to train models on easily-generated, automatically-annotated synthetic data obtained from simulations.

One solution towards region-based video understanding frameworks that generalize across domains, would be to leverage robust region proposals from domain adaptive object detectors [280], and then follow a semi-supervised training approach, such as [281], during which we first pretrain our region-based framework on source data with labels (e.g., simulated data), and then apply the pretrained framework on a target dataset to obtain pseudo-labels, which can be used to fine-tune the model in that new domain.

The computer vision algorithms proposed in this thesis were designed with the goal of automatic video understanding in unconstrained videos. The unique computer vision challenges that we addressed - detection of fine-grained, co-occurring activities, lack of fine-grained bounding box annotations, extended videos with multiple types of actors - were significant obstacles in the deployment of video understanding systems in real-life applications. We therefore hope that this thesis will ultimately contribute to better assistive robotic agents, and public safety systems.

Bibliography

[1] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal Robotics Research*, 32(8):951–970, 2013.

[2] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526, 2016.

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[4] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *European Conference on Computer Vision*, September 2018.

[8] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[9] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *IEEE Winter Applications of Computer Vision Conference*, 2021.

[11] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *European Conference on Computer Vision*, pages 401–417, 2018.

[12] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[13] YouTube usage statistics. https://blog.youtube/press/.

[14] Surveillance cameras installed in the us. https://www.theverge.com/2019/12/9/21002515/surveillance-cameras-globally-us-china-amount-citizens.

[15] Niels Haering, Péter L Venetianer, and Alan Lipton. The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19(5):279–290, 2008.

[16] Adriana Tapus, Maja J Mataric, and Brian Scassellati. Socially assistive robotics [grand challenges of robotics]. *IEEE robotics & automation magazine*, 14(1):35–42, 2007.

[17] Isidoros Rodomagoulakis, Nikolaos Kardaris, Vassilis Pitsikalis, Effrosyni Mavroudi, Athanasios Katsamanis, Antigoni Tsiami, and Petros Maragos. Multimodal human action recognition in assistive human-robot interaction. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2702–2706. IEEE, 2016.

[18] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42, 1990.

[19] Karl Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image understanding*, 59(1):94–115, 1994.

[20] Lee W Campbell and Aaron F Bobick. Recognition of human body motion using phase space constraints. In *Proceedings of IEEE international conference on computer vision*, pages 624–630. IEEE, 1995.

[21] Dariu M Gavrila and Larry S Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proceedings cvpr ieee computer society conference on computer vision and pattern recognition*, pages 73–80. IEEE, 1996.

[22] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, volume 92, pages 379–385, 1992.

[23] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402. IEEE, 2005.

[24] Ramprasad Polana and Randal Nelson. Detecting activities. *Journal of Visual Communication and Image Representation*, 5(2):172–180, 1994.

[25] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[26] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, volume 3, pages 726–726. IEEE Computer Society, 2003.

[27] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[28] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[29] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[30] Rizwan Chaudhry, Avinash Ravichandran, Gregory D. Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939, 2009.

[31] Avinash Ravichandran, Rizwan Chaudhry, and René Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems.

In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1651–1657. IEEE, 2009.

[32] Avinash Ravichandran, Rizwan Chaudhry, and Rene Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):342–353, 2012.

[33] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[34] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[36] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, 2015.

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[40] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy tradeoffs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[41] Colin Lea, René Vidal, and Gregory D Hager. Learning convolutional action primitives for fine-grained action recognition. In *IEEE international conference on robotics and automation*, pages 1642–1649. IEEE, 2016.

[42] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52. Springer, 2016.

[43] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1012, 2017.

[44] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017.

[45] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *IEEE International Conference on Computer Vision*, October 2019.

[46] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE International Conference on Computer Vision*, pages 6824–6835, 2021.

[47] Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR 2011*, pages 3177–3184. IEEE, 2011.

[48] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S. Davis. Representing videos using mid-level discriminative patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[49] Effrosyni Mavroudi, Lingling Tao, and René Vidal. Deep Moving Poselets for Video Based Action Recognition. In *IEEE Winter Applications of Computer Vision Conference*, pages 111–120, 2017.

[50] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual Action Recognition with R*CNN. In *IEEE International Conference on Computer Vision*, pages 1080–1088, 2015.

[51] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *European Conference on Computer Vision*, pages 106–122, 2018.

[52] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1365–1372. IEEE, 2009.

[53] Michalis Raptis and Leonid Sigal. Poselet key-framing: A model for human activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2650–2657, 2013.

[54] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action Recognition with Actons. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 3559–3566, December 2013.

[55] Du Tran and Lorenzo Torresani. EXMOVES: Mid-level Features for Efficient Action Recognition and Video Analysis. *International Journal of Computer Vision*, 119(3):239–253, April 2016.

[56] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto. Discovering discriminative action parts from mid-level video representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1242–1249, 2012.

[57] Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese. Action recognition by hierarchical mid-level action elements. In *Proceedings of the IEEE international conference on computer vision*, pages 4552–4560, 2015.

[58] Yu-Gang Jiang, Zhenguo Li, and Shih-Fu Chang. Modeling scene and object contexts for human action retrieval with few examples. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(5):674–681, 2011.

[59] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[60] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Neural Information Processing Systems*, 30, 2017.

[61] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1961–1970, 2016.

[62] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017.

[63] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling

of actions in complex videos. *International Journal of Computer Vision*, 126(2):375–389, 2018.

[64] A. J. Piergiovanni and Michael S. Ryoo. Temporal gaussian mixture layer for videos. In *International Conference on Machine learning*, 2019.

[65] Praveen Tirupattur, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Modeling multi-label action dependencies for temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1460–1470, 2021.

[66] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016.

[67] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016.

[68] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European conference on computer vision*, pages 768–784. Springer, 2016.

[69] H. Xu, A. Das, and K. Saenko. R-C3d: Region Convolutional 3d Network for Temporal Activity Detection. In *IEEE International Conference on Computer Vision*, pages 5794–5803, 2017.

[70] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[71] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019.

[72] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[73] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[74] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017.

[75] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2017.

[76] Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille. Attention correctness in neural image captioning. In *AAAI Conference on Artificial Intelligence*, 2017.

[77] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, 2018.

[78] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, pages 771–787, 2018.

[79] Peter Carbonetto, Nando De Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. *European Conference on Computer Vision*, 2004.

[80] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2012–2019, 2009.

[81] Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. Generating descriptions with grounded and co-referenced people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4979–4989, 2017.

[82] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[83] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2017-January, pages 5253–5262, 11 2017.

[84] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *European Conference on Computer Vision*, volume 12348 LNCS, 2020.

[85] Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In *Annual Meeting of the Association for Computational Linguistics*, volume 1, 2013.

[86] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *European Conference on Computer Vision*, volume 8689 LNCS, 2014.

[87] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676, 4 2017.

[88] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh,

and Ajay DIvakaran. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *IEEE International Conference on Computer Vision*, volume 2019-October, pages 2601–2610, 10 2019.

[89] Luowei Zhou, Nathan Louis, and Jason J. Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *British Machine Vision Conference*, 2019.

[90] De An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding 'it': Weakly-supervised reference-aware visual grounding in instructional videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[91] Yang Wang and Minh Hoai. Pulling actions out of context: Explicit separation for effective combination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7044–7053, 2018.

[92] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. *Neural Information Processing Systems*, 32, 2019.

[93] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[94] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware RCNN: A baseline for action detection in videos. In *European Conference on Computer Vision*, pages 440–456, 2020.

[95] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for AVA. *CoRR*, abs/1807.10066, 2018.

[96] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *European Conference on Computer Vision*, pages 318–334, 2018.

[97] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.

[98] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H.S. Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference*, 2016.

[99] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip H.S. Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. *IEEE International Conference on Computer Vision (ICCV), 2017*, 2017.

[100] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *IEEE International Conference on Computer Vision*, Oct 2017.

[101] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (T-CNN) for action detection in videos. In *IEEE International Conference on Computer Vision*, 2017.

[102] Guilhem Chéron, Anton Osokin, Ivan Laptev, and Cordelia Schmid. Modeling spatio-temporal human track structure for action localization. *CoRR*, abs/1806.11008, 2018.

[103] Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation learning on visual-symbolic graphs for video understanding. In *European Conference on Computer Vision*, volume 12374 LNCS, 2020.

[104] Effrosyni Mavroudi and René Vidal. Weakly-supervised generation and grounding of visual descriptions with conditional generative models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[105] Effrosyni Mavroudi, Prashast Bindal, and René Vidal. Actor-centric tubelets for real-time activity detection in extended videos. In *IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2022.

[106] Elena Kokkoni, Effrosyni Mavroudi, Ashkan Zehfroosh, James C Galloway, Renè Vidal, Jeffrey Heinz, and Herbert G Tanner. Gearing smart environments for pediatric motor rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 17, 2020.

[107] Carolina Pacheco, Effrosyni Mavroudi, Elena Kokkoni, Herbert G Tanner, and René Vidal. A detection-based approach to multiview action classification in infants. In *IEEE International Conference on Pattern Recognition*, 2020.

[108] Effrosyni Mavroudi, Divya Bhaskara, Shahin Sefati, Haider Ali, and

René Vidal. End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding. In *IEEE Winter Applications of Computer Vision Conference*, 2018.

[109] Daphne Koller, Joseph Weber, Timothy Huang, Jitendra Malik, G Ogasawara, B Rao, and S Russell. Towards robust automatic traffic scene analysis in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[110] Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[111] Xiaoyang Wang and Qiang Ji. Video event recognition with deep hierarchical context model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[112] Song-Chun Zhu, David Mumford, et al. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2(4):259–362, 2007.

[113] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *European Conference on Computer Vision*, pages 413–431, 2018.

[114] Hao Huang, Luowei Zhou, Wei Zhang, and Chenliang Xu. Dynamic Graph Modules for Modeling Higher-Order Interactions in Activity Recognition. In *British Machine Vision Conference*, 2019.

[115] Y. Yuan, X. Liang, X. Wang, D. Yeung, and A. Gupta. Temporal Dynamic Graph LSTM for Action-Driven Video Object Detection. In *IEEE International Conference on Computer Vision*, pages 1819–1828, 2017.

[116] Pallabi Ghosh, Yi Yao, Larry Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. In *IEEE Winter Applications of Computer Vision Conference*, 2020.

[117] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding. In *IEEE International Conference on Computer Vision*, pages 8117–8126, October 2021.

[118] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic Graph Reasoning Meets Convolutions. In *Neural Information Processing Systems*, pages 1853–1863. Curran Associates, Inc., 2018.

[119] X. Chen, L. Li, L. Fei-Fei, and A. Gupta. Iterative Visual Reasoning Beyond Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[121] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520 – 527, 2007.

[122] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936. IEEE, 2009.

[123] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[124] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2010.

[125] Yuanhao Chen, Long Zhu, Chenxi Lin, Hongjiang Zhang, and Alan L Yuille. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. *Advances in neural information processing systems*, 20, 2007.

[126] Benjamin Z. Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song Chun Zhu. I2t: Image parsing to text description. In *Proceedings of the IEEE*, volume 98, 2010.

[127] Jake Porway, Qiongchen Wang, and Song Chun Zhu. A hierarchical and contextual model for aerial image parsing. *International journal of computer vision*, 88(2):254–283, 2010.

[128] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *IEEE International Conference on Computer Vision*, 2011.

[129] Bangpeng Yao and Li Fei-Fei. Action recognition with exemplar based 2.5d graph matching. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *European Conference on Computer Vision*, pages 173–186, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[130] Marcin Marszalek and Cordelia Schmid. Constructing Category Hierarchies for Visual Recognition. In *European Conference on Computer Vision*, ECCV '08, pages 479–491, Berlin, Heidelberg, 2008. Springer-Verlag.

[131] M. Marszalek and C. Schmid. Semantic Hierarchies for Visual Object Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[132] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–136, 2010.

[133] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-Scale Object Classification Using Label Relation Graphs. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 48–64, 2014.

[134] C. Lee, W. Fang, C. Yeh, and Y. F. Wang. Multi-label Zero-Shot Learning with Structured Knowledge Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1576–1585, 2018.

[135] Ruiyu Li, Makarand Tapaswi, Renjie Liao, Jiaya Jia, Raquel Urtasun, and Sanja Fidler. Situation recognition with graph neural networks. In *IEEE International Conference on Computer Vision*, 2017.

[136] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rossenberg, and L. Fei-Fei. Learning semantic relationships for better action retrieval in images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1100–1109, 2015.

[137] Shayan Modiri Assari, Amir Roshan Zamir, and Mubarak Shah. Video classification using semantic concept co-occurrences. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[138] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):352–364, 2018.

[139] N. I. Nauata Junior, H. Hu, G. Zhou, Z. Deng, Z. Liao, and G. Mori. Structured Label Inference for Visual Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[140] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4772–4781, 2016.

[141] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep

Learning on Spatio-Temporal Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[142] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[143] Andrei Nicolicioiu, Iulia Duta, and Marius Leordeanu. Recurrent space-time graph neural networks. In *Neural Information Processing Systems*, 2019.

[144] Mostafa S Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *European Conference on Computer Vision*, pages 721–736, 2018.

[145] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine learning*, pages 1263–1272, 2017.

[146] C. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf. Attend and Interact: Higher-Order Object Interactions for Video Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018.

[147] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine learning*, volume 2, page 4, 2021.

[148] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, page 457–466, 2018.

[149] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[150] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[151] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710, 2020.

[152] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2017.

[153] Elvin Isufi, Fernando Gama, and Alejandro Ribeiro. Edgenets: Edge varying graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[154] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Neural Information Processing Systems*, pages 9225–9235, 2018.

[155] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[156] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. G3raphground: Graph-based language grounding. In *IEEE International Conference on Computer Vision*, pages 4281–4290, 2019.

[157] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *IEEE International Conference on Computer Vision*, 2019.

[158] Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. A graph-based framework to bridge movies and synopses. In *IEEE International Conference on Computer Vision*, 2019.

[159] Weijiang Yu, Jingwen Zhou, Weihao Yu, Xiaodan Liang, and Nong Xiao. Heterogeneous graph learning for visual commonsense reasoning. In *Advances in Neural Information Processing Systems 32*, pages 2769–2779. Curran Associates, Inc., 2019.

[160] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[161] ChenHan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid Knowledge Routed Modules for Large-scale Object Detection. In *Neural Information Processing Systems*, pages 1552–1563, 2018.

[162] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[163] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[164] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision*, 2021.

[165] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[166] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2015.

[167] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, pages 801–818, 2018.

[168] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[169] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[170] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Neural Information Processing Systems*, pages 3111–3119, 2013.

[171] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[172] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[173] Ninghang Hu, Gwenn Englebienne, Zhongyu Lou, and Ben Kröse. Latent hierarchical model for activity recognition. *IEEE Transactions on Robotics*, 31(6):1472–1482, 2015.

[174] Haroon Idrees, Amir R. Zamir, Yu Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155, 2017.

[175] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[176] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.

[177] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Neural Information Processing Systems*, pages 568–576. Curran Associates, Inc., 2014.

[178] A. Piergiovanni and M. S. Ryoo. Learning Latent Super-Events to Detect Multiple Activities in Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018.

[179] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[180] A. Dave, O. Russakovsky, and D. Ramanan. Predictive-Corrective Networks for Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2067–2076, 2017.

[181] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous Temporal Fields for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5650–5659, 2017.

[182] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A.

Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[183] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Object referring in videos with language and human gaze. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4129–4138, 2018.

[184] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10668–10677, 2020.

[185] Sijie Song, Xudong Lin, Jiaying Liu, Zongming Guo, and Shih-Fu Chang. Co-grounding networks with semantic attention for referring expression comprehension in videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1355, 2021.

[186] Ludan Ruan, Jieting Chen, Yuqing Song, Shizhe Chen, and Qin Jin. Team ruc_aim3 technical report at activitynet 2021: Entities object localization. *CoRR*, abs/2106.06138, 2021.

[187] Peter Gorniak and Deb Roy. Understanding complex visually referring utterances. In *Proceedings of the HLT-NAACL 2003 workshop on Learning word meaning from non-linguistic data*, pages 14–21, 2003.

[188] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded

attribute learning. In *International Conference on Machine learning*, pages 1435–1442, 2012.

[189] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3558–3565, 2014.

[190] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqground). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4175–4184, 2019.

[191] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[192] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229, 2020.

[193] Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwing. Interpretable and globally optimal prediction for textual grounding using image concepts. *Advances in Neural Information Processing Systems*, 30, 2017.

[194] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual

grounding. In *IEEE International Conference on Computer Vision*, October 2019.

[195] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *IEEE International Conference on Computer Vision*, pages 1780–1790, 2021.

[196] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, 2016.

[197] Chih-Yao Ma, Yannis Kalantidis, Ghassan AlRegib, Peter Vajda, Marcus Rohrbach, and Zsolt Kira. Learning to generate grounded visual captions without localization supervision. In *European Conference on Computer Vision*, 2020.

[198] Fenglin Liu, Xuancheng Ren, Xian Wu, Shen Ge, Wei Fan, Yuexian Zou, and Xu Sun. Prophet attention: Predicting attention with future attention. In *Neural Information Processing Systems*, 2020.

[199] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. More grounded image captioning by distilling image-text matching model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4776–4785, 8 2020.

[200] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show,

attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015.

[201] Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of attention for image captioning. In *IEEE International Conference on Computer Vision*, 2017.

[202] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. In *Neural Information Processing Systems*, volume 2018-December, 2018.

[203] Shiv Shankar and Sunita Sarawagi. Posterior attention models for sequence to sequence learning. In *International Conference on Learning Representations*, 2019.

[204] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Meeting of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[205] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neural Information Processing Systems*, pages 13–23, 2019.

[206] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 2020.

[207] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Neural Information Processing Systems*, volume 2015-January, 2015.

[208] Anirudh Goyal, Alessandro Sordoni, Marc Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Neural Information Processing Systems*, volume 2017-December, 2017.

[209] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI Conference on Artificial Intelligence*, 2017.

[210] Hong Min Chu, Chih Kuan Yeh, and Yu Chiang Frank Wang. Deep generative models for weakly-supervised multi-label classification. In *European Conference on Computer Vision*, pages 409–425. Springer Verlag, 2018.

[211] Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. Sequential latent spaces for modeling the intention during diverse image captioning. In *IEEE International Conference on Computer Vision*, pages 4260–4269, 2019.

[212] Manzil Zaheer, Amr Ahmed, and Alexander J. Smola. Latent lstm allocation joint clustering and non-linear dynamic modeling of sequential data. In *International Conference on Machine learning*, volume 8, 2017.

[213] Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Neural Information Processing Systems*, volume 2017-December, 2017.

[214] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a 'best of many' sample objective. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[215] Artidoro Pagnoni, Kevin Liu, and Shangyan Li. Conditional variational autoencoder for neural machine translation, 12 2018.

[216] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Annual Meeting of the Association for Computational Linguistics*, 2017.

[217] Colin Graber and Alexander G. Schwing. Dynamic neural relational inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[218] Alexander A. Alemi, Ben Poole, Ian Fische, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine learning*, volume 1, 2018.

[219] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. In *Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, volume 1, 2019.

[220] Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. A surprisingly effective fix for deep latent variable modeling of text. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[221] Arya D. McCarthy, Xian Li, Jiatao Gu, and Ning Dong. Addressing posterior collapse with mutual information for improved variational neural machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[222] Ali Razavi, Oriol Vinyals, Aäron Van Den Oord, and Ben Poole. Preventing posterior collapse with δ-vaes. In *International Conference on Learning Representations*, 2019.

[223] Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. Controlvae: Controllable variational autoencoder. In *International Conference on Machine learning*, 2020.

[224] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2019.

[225] Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. Avoiding latent variable collapse with generative skip models. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[226] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

[227] Giorgos Tziafas and Hamidreza Kasaei. Few-shot visual grounding for natural human-robot interaction. In *2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 50–55. IEEE, 2021.

[228] Mohit Shridhar and David Hsu. Interactive visual grounding of referring expressions for human-robot interaction. In *Proceedings of Robotics: Science and Systems*, 2018.

[229] Pingping Huang, Jianhui Huang, Yuqing Guo, Min Qiao, and Yong Zhu. Multi-grained attention with object-level grounding for visual question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3595–3600, Florence, Italy, July 2019. Association for Computational Linguistics.

[230] Aisha Urooj, Hilde Kuehne, Kevin Duarte, Chuang Gan, Niels Lobo, and Mubarak Shah. Found a reason for me? weakly-supervised grounded visual question answering using capsules. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8465–8474, June 2021.

[231] Gunnar A. Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, João Carreira, Phil Blunsom, and Andrew Zisserman. Visual grounding in video for unsupervised word translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[232] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *IEEE International Conference on Computer Vision*, 2021.

[233] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[234] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[235] Alon Lavie and Michael J Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2):105–115, 2009.

[236] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.

[237] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[238] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[239] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.

[240] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. In *Foundations and Trends in Machine Learning*, 2019.

[241] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Neural Information Processing Systems*, 2015.

[242] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[243] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[244] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another

way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[245] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[246] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[247] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[248] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, 2017.

[249] Wei Wang, Junyu Gao, and Changsheng Xu. Weakly-supervised video object grounding via stable context learning. In *ACM International Conference on Multimedia*, New York, NY, USA, 2021. Association for Computing Machinery.

[250] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In *European Conference on Computer Vision*, 2020.

[251] Xun Yang, Xueliang liu, Meng Jian, Xinjian Gao, and Meng Wang. Weakly-supervised video object grounding by exploring spatio-temporal contexts. In *ACM International Conference on Multimedia*, New York, NY, USA, 2020. Association for Computing Machinery.

[252] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[253] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[254] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[255] Joshua Gleason, Rajeev Ranjan, Steven Schwarcz, Carlos D. Castillo, Jun Cheng Chen, and Rama Chellappa. A proposal-based solution to spatio-temporal action detection in untrimmed videos. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[256] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan Dave, Yogesh Singh

Rawat, and Mubarak Shah. Gabriella: An online system for real-time activity detection in untrimmed security videos. In *IEEE International Conference on Pattern Recognition*, 2020.

[257] Yasaman S. Sefidgar, Arash Vahdat, Stephen Se, and Greg Mori. Discriminative key-component models for interaction detection and recognition. *Computer Vision and Image Understanding*, 135, 2015.

[258] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[259] Namrata Vaswani, Amit Roy Chowdhury, and Rama Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2003.

[260] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[261] Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware activity modeling using hierarchical conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 2015.

[262] Mohamed R. Amer and Sinisa Todorovic. Sum-product networks for modeling activities with stochastic structure. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[263] M. Jain, J. C. van Gemert, H. Jegou, P. Bouthemy, and C. G. M. Snoek. Tubelets: Unsupervised action proposals from spatiotemporal supervoxels. *International Journal of Computer Vision*, 2017.

[264] Joshua Gleason, Carlos D. Castillo, and Rama Chellappa. Real-time detection of activities in untrimmed videos. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, March 2020.

[265] Wenhe Liu, Guoliang Kang, Po Yao Huang, Xiaojun Chang, Lijun Yu, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, and Alexander G. Hauptmann. Argus: Efficient activity detection system for extended video analysis. In *IEEE Winter Conference on Applications of Computer Vision Workshops*, 2020.

[266] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, 1994.

[267] Darnell J. Moore, Irfan A. Essa, and Monson H. Hayes. Exploiting human actions and object context for recognition tasks. In *IEEE International Conference on Computer Vision*, volume 1, 1999.

[268] Nandita M. Nayak, Yingying Zhu, and Amit K. Roy Chowdhury. Hierarchical graphical models for simultaneous tracking and recognition in wide-area scenes. *IEEE Transactions on Image Processing*, 24, 2015.

[269] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft.

Simple online and realtime tracking. In *IEEE International Conference on Image Processing*, pages 3464–3468, 2016.

[270] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal Chem*, 36(8):1627–1639, 1964.

[271] Actev 2021 sdl evaluation plan. https://actev.nist.gov/pub/Phase3_ActEV_2021_SDL_EvaluationPlan_20210803.pdf.

[272] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. CMU informedia at TRECVID 2020: Activity detection with dense spatio-temporal proposals. In *TREC Video Retrieval Evaluation, TRECVID, 2020*, 2020.

[273] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *CoRR*, 2019.

[274] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *IEEE International Conference on Computer Vision*, pages 13708–13718, October 2021.

[275] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10776–10785, 2021.

[276] Cristina González, Nicolás Ayobi, Isabela Hernández, José Hernández, Jordi Pont-Tuset, and Pablo Arbeláez. Panoptic narrative grounding. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1364–1373, 2021.

[277] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021.

[278] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[279] Ishan Dave, Zacchaeus Scheffer, Akash Kumar, Sarah Shiraz, Yogesh Singh Rawat, and Mubarak Shah. Gabriellav2: Towards better generalization in surveillance videos for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 122–132, 2022.

[280] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490, 2019.

[281] Ankit Singh, Omprakash Chakraborty, Ashutosh Varshney, Rameswar Panda, Rogerio Feris, Kate Saenko, and Abir Das. Semi-supervised action recognition with temporal contrastive learning. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10389–10399, 2021.

Vita

Effrosyni Mavroudi received her Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 2015. Next, she joined the Biomedical Engineering PhD program at Johns Hopkins University in 2015. She is a former intern with Google (Summer 2020) and Facebook Reality Labs (Summer 2021). Her research interests lie in the broad areas of computer vision and machine learning, with a focus on deep learning for video understanding. She is excited to be working at Meta AI (FAIR) as a research scientist after completing her PhD.