Machine perception of interactivity in videos

Chen, S.

Publication date
2023
Document Version
Final published version

Challenging the dependence on dense annotations, this work opens new avenues in machine learning to perceive video interactivity. It paves the way for systems to learn from minimal supervision, temporal patterns, and multimodal data, transforming our understanding of interactivity.

Machine Perception of Interactivity in Videos

Shuo Chen

Machine Perception of Interactivity in Videos

Shuo Chen

Machine Perception of Interactivity in Videos

Shuo Chen

This book was typeset by the author using LaTeX 2_ε.

Machine Perception of Interactivity in Videos

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op vrijdag 14 juli 2023 te 11:00 uur

door

Shuo Chen

geboren te Anhui, China

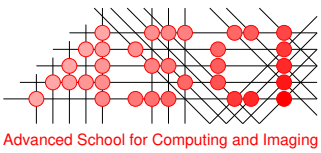
Promotiecommissie

Promotor:	prof. dr. C.G.M. Snoek	Universiteit van Amsterdam
Co-promotor:	dr. P.S.M. Mettes	Universiteit van Amsterdam
Overige leden:	prof. dr. P.T. Groth	Universiteit van Amsterdam
	prof. dr. M.R. Lindegaard	Universiteit van Amsterdam
	prof. dr. A.A. Salah	Universiteit Utrecht
	prof. dr. R. Cucchiara	Università di Modena e Reggio Emilia
	dr. I.I.A. Groen	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

UNIVERSITEIT VAN AMSTERDAM

The work was carried out in the ASCI graduate school, ASCI dissertation number 447, and at the Video & Image Sense lab of the University of Amsterdam.



CONTENTS

1

INTRODUCTION

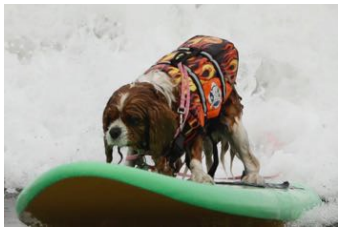
1.1 INTERACTIVITY AND COGNITIVE PSYCHOLOGY

As humans, our brains perceive objects through the visual system and interact with the environment automatically and unconsciously. Cognitive psychologists often use the example of picking up a cup of coffee, as shown in Figure 1, to illustrate the interaction of perception with objects [48]. First, the person identifies the coffee cup among the other objects on the table and approaches it while avoiding obstacles. As they reach for the cup, they consider its position on the table and adjust their fingers to grasp the handle, all while continuously perceiving the cup's location relative to their hand and fingers. Finally, they lift the cup without spilling any coffee, requiring the calibration of their actions. This simple daily activity involves a series of interactive processes between the person and the cup, including changes in their relationship from simply looking at the cup to approaching, grasping, and lifting it. These interactive processes are collectively known as interactivity. While humans perform these processes easily and unconsciously, it is challenging to make a machine understand and detect interactivity. The machine must be able to recognize objects and understand how they interact with each other, including their relationships and how these relationships may change over time.

The ability of machines to perceive interactivity plays a vital role in our daily lives. One such application is in surveillance systems, where analyzing the interactions between objects and humans can help detect violent behavior [26], identify potential threats [134],

(a) Look at cup (b) Approach cup (c) Grasp cup

Figure 1: The interactivity between a person and the cup: (a) perceiving and recognizing the cup; (b) approaching it; (c) grasping and picking it up. While it's a routine activity for humans to interact with objects like a coffee cup, it is a challenging task for machines to perceive and understand the interactivity involved. Figure from [48].



ride or stand?

get on or off?

open or close?

Figure 2: The interactivity can only be determined after providing their motion information. Without taking the motion signal into account, recognizing these activities causes ambiguity for both humans and machine understanding.

and improve overall security [22]. In addition, such systems can also be used to assist in caring for the elderly [60] and protecting wildlife [20]. Another significant application is in robotics. By understanding human-object interactivity, robots can better interact with objects and humans, and perform tasks more efficiently [9, 71]. In summary, it is crucial for machines to understand interactivity for a wide range of applications.

1.2 PERCEIVING INTERACTIVITY IN VIDEOS

How can we make a machine perceive interactivity in videos? One leading solution in the computer vision literature is learning rich context information from a large amount of data [15, 69, 155]. There are three aspects of context information that we can leverage: First, motion information is necessary for the machine to predict interactivity correctly. As shown in Figure 2, the motion signal is essential for the human and machine to understand the interactivity. Second, modeling a sequence of interactivities between moving objects is helpful, as interactivities could occur sequentially in many videos. Such modeling will be beneficial for reasoning future interactivity. Third, besides visual information, multiple modalities included in video signals, such as audio and spoken language, are beneficial when perceiving interactivity.

After we know that learning from rich data is essential, the next question is what we want the machine to learn. In other words, how to formalize the interactivity in videos? One way is by using a semantic triplet structure, which consists of subject–predicate–object [89]. The semantic triplet represents an interactivity in the video as a combination of three components: the subject performing the interactivity, the interactivity itself (predicate), and the object being interacted upon. For instance, in the triplet ⟨*person–open–trunk*⟩, the subject is *person*, the predicate is *open*, and the object is *trunk*. The semantic output can bridge the gap between the video and natural language, leading to better performance in downstream tasks, such as captioning [139], grounding [42], video retrieval [112], and visual question answering [4]. However, this triplet by itself is not enough, as the machine should also output temporal and spatial boundaries of the interactivity. Temporal boundaries are necessary to understand a sequence of interactivities, while spatial boundaries are essential to describing where multiple interactivities could occur simultaneously at a given timestamp. The machine should output continuous bounding boxes to indicate the location of corresponding objects. To better understand in-

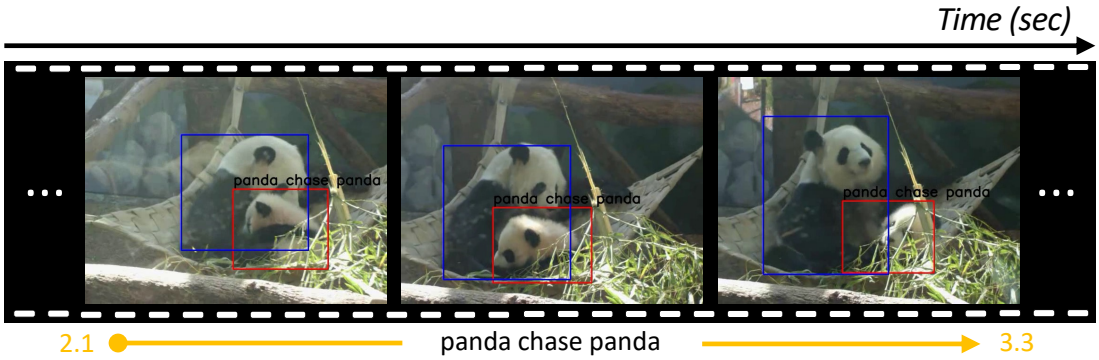


Figure 3: An example detection of the interactivity specified by the semantic triplet: ⟨*panda–chase–panda*⟩.

teractivity in videos, the machine should output the semantic triplet, temporal boundaries, and spatial bounding boxes to understand interactivity in videos. Figure 3 demonstrates an example.

In this thesis, we investigate what, when, and where specific interactivities occur in video by examining multi-modal context information. To obtain the interactivity's semantic triplet as well as its spatio-temporal boundaries, we need to design and analyze automatic methods that address several challenging problems in computer vision. These may include object detection [146], multiple object tracking [27], and scene graph generation [16]. The task is challenging as it requires a deep understanding of the underlying principles governing visual perception and reasoning. Developing solutions entails advancing the state-of-the-art in video relation detection [108], human-object relationship detection [65], and dynamic scene graph generation [28]. Perceiving interactivity is not fully explored in video, in this thesis, we focus on investigating the perception and understanding of interactivity, and designing methods to obtain an interactivity as specified by a semantic triplet and its spatio-temporal boundaries in a systematic and comprehensive way.

1.3 RESEARCH QUESTIONS

Detecting interactivity in video content is a challenging problem. In this thesis, we ask the key research question:

How to automate the perception of interactivity in video content?

Recognizing an interactivity in video requires not only the reasoning from multi-modal information, but also the determination of its spatio-temporal location. Due to the challenging nature, we divide the main research question into four subquestions. First, we want to know which pair of objects have interactivity in video content. Therefore we pose the following research question:

How to define interactivity in video content?

We address this question in Chapter 2. Instead of just considering the actions per-

formed by subjects, our approach takes into account the objects that are involved in the interactions as well. We introduce a score called interactivityness, which reflects the likelihood that a subject and an object are interacting. We propose a network that uses the subject and object trajectories to compute local interactivity likelihoods, which identify intervals of high interactivityness and generate spatio-temporal interactivity proposals. We evaluate our approach on an interactivity dataset using new evaluation metrics and show that our approach outperforms traditional temporal and spatio-temporal action proposal methods.

We are interested in finding the spatio-temporal interactivity of interest for the second research question and assigning the appropriate interactivity label, a task also known as video relation detection [108]. We pose the following research question:

How to recognize interactivity in video content?

In Chapter 3, we aim to detect and classify the relationship between object tubelets in a video as a *subject-predicate-object* triplet like ⟨*adult-chase-child*⟩. Previous works have treated object proposals or tubelets as individual entities and modeled their relations *a posteriori*. Instead, we propose classifying and detecting predicates for pairs of object tubelets *a priori*. We also introduce Social Fabric, an encoding representing a pair of object tubelets as a combination of interaction primitives. These primitives are learned from all relations, resulting in a compact representation that can localize and classify relations from the pool of co-occurring object tubelets across all timespans in a video. We design a two-stage network utilizing the encoding. In the first stage, we train Social Fabric to suggest likely interacting proposals. In the second stage, we use Social Fabric to fine-tune and simultaneously predict predicate labels for the tubelets. Our experiments show the benefits of early video relation modeling, our encoding, and the two-stage architecture, achieving a new state-of-the-art on two benchmarks. We also demonstrate how the encoding enables query-by-primitive-example to search for spatio-temporal interactivities in video content.

For the third question, we analyze the error sources of state-of-the-art methods for interactivity recognition in video content. We pose the research question:

What makes recognizing interactivity in video content hard?

To answer this question, we start by highlighting the errors made by current methods in Chapter 4. The problem of recognizing interactivity in video content is a challenging one in computer vision, requiring the localization of subjects and objects in both space and time, as well as the assignment of a predicate label when there is an interaction between the two. While recent progress has been made in this area, overall performance is still relatively low and it is not yet clear what the key factors are in solving the problem. In order to better understand the sources of errors in current approaches to video relation detection, we have developed a diagnostic tool that goes beyond the standard metric of mean Average Precision by defining different error types specific to this problem. Our tool allows us to evaluate and compare existing approaches, and to conduct false positive and false negative analyses. We have also studied the influence of various factors, such as relation length, the number of instances, and subject/object spatial size, on performance.

Finally, we have examined the effect on video relation performance when certain error types are corrected. Our results, which are based on two benchmarks, highlight the strengths and weaknesses of current approaches and suggest potential directions for future research.

For the fourth question, we turn our attention to the inherent long-tail of interactivity recognition in video content. We propose the research question:

How to recognize rare interactivities in video content?

In Chapter 5, we show that current methods for interactivity recognition in video content are limited by the available data. In popular benchmarks, the imbalance ratio between the most and least frequent predicates is extremely high, even higher than in benchmarks specifically designed to address long-tailed recognition. As a result, state-of-the-art methods often rely heavily on the most common predicate classes, ignoring those in the long tail. We analyze the limitations of these approaches and find that there is a strong correlation between predicate frequency and recall performance. To address this bias, we propose a multi-label meta-learning framework that learns a meta-weight network for each training sample based on all possible label losses. We evaluate our approach on the two benchmarks using two state-of-the-art methods per benchmark. Our experiments show that our multi-label meta-weight network improves performance for predicates in the long tail without sacrificing performance for head classes, resulting in better overall performance and improved generalizability.

To summarize, this thesis aims to systematically and comprehensively investigate the machine perception of interactivity in videos. Specifically, we start with the definition and detection of interactivity proposals. Then we learn how to recognize the interactivity, the relationship between the subject and the object, using the rich multi-modal information provided in videos. Following, we provide an analytic tool to diagnose methods for interactivity detection. Finally, we study the problem of classifying rare interactivities in videos. We conclude the thesis in Chapter 6.

List of Publications

- **Chapter 2** is based on "Interactivity Proposals for Surveillance Videos", *International Conference on Multimedia Retrieval*, 2020 [22], by Shuo Chen, Pascal Mettes, Tao Hu, and Cees G. M. Snoek.

  Contribution of authors

  Shuo Chen: all aspects,
  Pascal Mettes: guidance and technical advice,
  Tao Hu: technical advice,
  Cees G. M. Snoek: supervision and insight.

- **Chapter 3** is based on "Social Fabric: Tubelet Compositions for Video Relation Detection", *International Conference on Computer Vision*, 2021 [24], by Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek.

  Contribution of authors

  Shuo Chen: all aspects,
  Zenglin Shi: technical advice,
  Pascal Mettes: guidance and technical advice,
  Cees G. M. Snoek: supervision and insight.

- **Chapter 4** is based on "Diagnosing Errors in Video Relation Detectors", *British Machine Vision Conference*, 2021 [23], by Shuo Chen, Pascal Mettes, Cees G. M. Snoek.

  Contribution of authors

  Shuo Chen: all aspects,
  Pascal Mettes: guidance and technical advice,
  Cees G. M. Snoek: supervision and insight.

- **Chapter 5** is based on "Multi-Label Meta Weighting for Long-Tailed Dynamic Scene Graph Generation", *International Conference on Multimedia Retrieval*, 2023 [21], by Shuo Chen, Yingjun Du, Pascal Mettes, and Cees G. M. Snoek.

  Contribution of authors

  Shuo Chen: all aspects,
  Yingjun Du: technical advice,
  Pascal Mettes: guidance and technical advice,
  Cees G. M. Snoek: supervision and insight.

More works by the author are provided in the Complete List of Publications at the end of this thesis.

2

INTERACTIVITY PROPOSALS FOR SURVEILLANCE VIDEOS

2.1 INTRODUCTION

The goal of this chapter is to generate spatio-temporal proposals that capture the interaction between subjects and objects in surveillance videos. Spatio-temporal proposals in videos are generally focused on actions [47, 52, 62, 122, 143], *i.e.* centered around subjects only. The objects with which actions might interact are generally ignored or only used implicitly. In surveillance settings, interactions between subjects and objects are key, because they denote important events to analyze. Think about a person entering a car or loading gear into a trunk. Since surveillance videos may contain several events that happen simultaneously, localizing the temporal extent of an interactivity is insufficient; spatial localization is mandatory. We aim to explicitly capture subjects performing actions, and the objects with which they interact, in space and time. We focus on the proposal generation step, where a video is split into spatio-temporal segments, upon which detection algorithms can be applied.

To arrive at spatio-temporal interactivity proposals, we take inspiration from *objectness* [2] and *actionness* [25, 128]. These approaches estimate the likelihood of object presence in a spatial region or action presence in a spatio-temporal region. Based on the likelihood, object or action proposals can be generated. Subsequently, such proposals are scored by classifiers to obtain object or action detections. Here, we take this line of work further and introduce *interactivityness*. Rather than estimating the individual likelihoods of objects or subjects performing an action, we estimate when and where subjects and objects are jointly occurring and are also in interaction. Akin to objectness and actionness, we use interactivityness to obtain interactivity proposals, which we define as pairs of subject and object trajectories with the same start and end time, see Figure 4.

We make three contributions in this chapter. First, we introduce the new task of spatio-temporal interactivity proposal generation in surveillance videos. Second, we introduce an interactivity network. This network estimates the interactivityness between a subject and object using an interactivity module that models the context around subjects and objects, as well as a geometric encoding that models the spatial relations of the pair. Third, we set up an interactivity proposal evaluation, including a dataset distilled from the ActEV surveillance benchmark [6] and interactivity evaluation metrics. Experiments on this evaluation show the effectiveness of our approach, outperforming existing approaches from the temporal and spatio-temporal action proposal literature. The dataset, evaluation protocols, and code are available at `https://github.com/shanshuo/Interactivity_Proposals`.

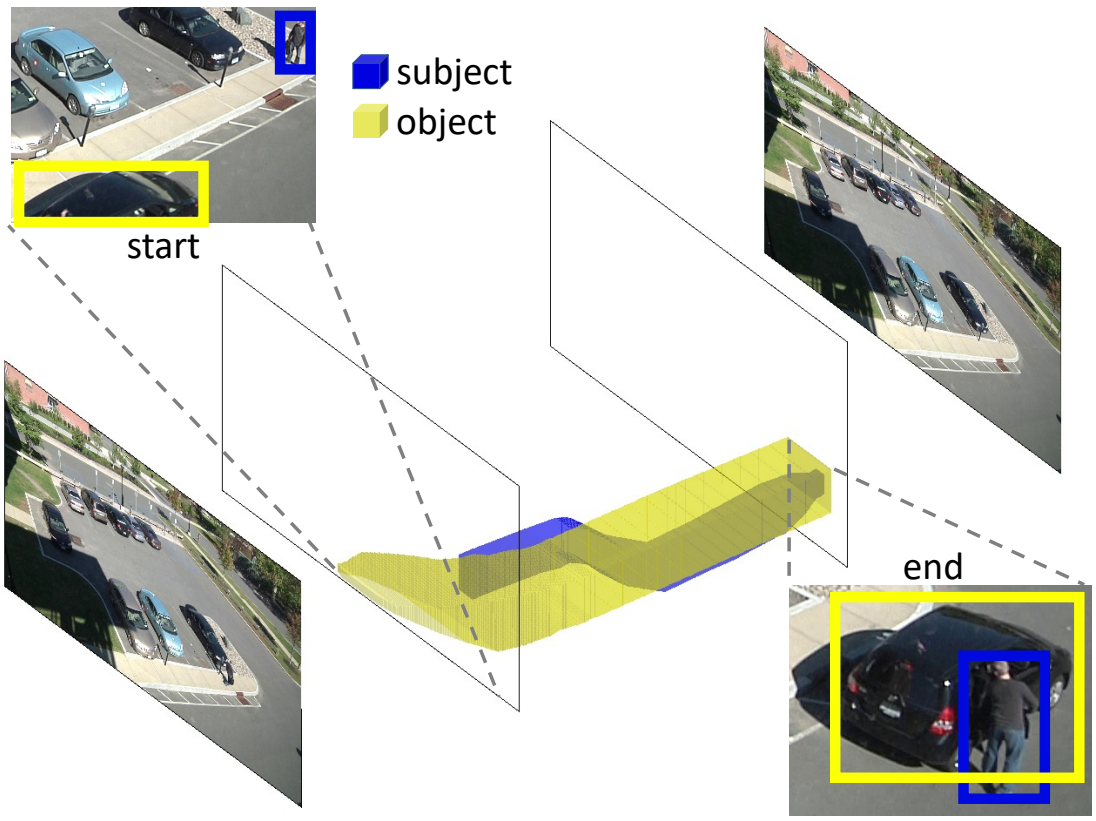


Figure 4: **Interactivity proposals** encapsulate a subject and object trajectory with the same start and end time. In this chapter we define, generate and evaluate this new type of proposals for video surveillance.

2.2 RELATED WORK

2.2.1 *Action proposals*

TEMPORAL ACTION PROPOSALS. Proposal methods for temporal action localization form an active research topic [12, 35, 40, 41, 82, 87, 99, 151, 151]. Escorcia *et al.* [35] utilize LSTMs on extracted CNN features to capture temporal information. Buch *et al.* [12] adopt the C3D network architecture as a feature extractor with a gated recurrent unit to capture long-term temporal information. Gao *et al.* [41] collect proposal candidates through a sliding window, which utilizes unit-level information for training. For each proposal, the average unit representation is adopted as proposal representation. Afterwards, temporal regression is performed on the unit-level to refine the start and end times of the proposals. Zhao *et al.* [151] generate actionness for each frame and group continuous frames with high actionness as proposals. All temporal action proposal methods use whole frames as input. In outdoor surveillance settings, many action and interactions can occur at the same time, hence using whole frames as input is not precise enough. Therefore, we target interactivity proposals in both space and time.

SPATIO-TEMPORAL ACTION PROPOSALS. Spatio-temporal action proposals target the spatio-temporal locations of subjects in videos [47, 52, 62, 95, 122, 143]. One

common manner to obtain spatio-temporal action proposals is by clustering local voxels or dense trajectories in a hierarchical manner [62, 95, 122]. Yu *et al.* [143] generate generic action proposals in unconstrained videos by linking subject detections over time. He *et al.* [52] propose a tupelet proposal network for action detection, which adopts Faster RCNN [101] to collect boxes with high action scorse. They link the highest scoring boxes to obtain tubelet proposals. Gleason *et al.* [47] generate spatio-temporal cuboid proposals by clustering detected boxes in spatio-temporal regions, followed by jittering to collect more proposals for better recall. Where current spatio-temporal proposal methods focus on actions only, we target spatio-temporal proposals of both subjects and objects. More concretely, where a spatio-temporal action proposal is described by a single tube, a spatio-temporal interactivity proposal is described by two tubes with the same start and end time. The tubes represent a subject and an object that should be in interaction.

2.2.2 *Visual human-object interaction*

A wide range of works have investigated the relationship between humans (subjects) and objects [17, 39, 46, 137, 148] in images. Gkioxari *et al.* [46] learn to predict an action-specific density over object locations using detected subjects. Chao *et al.* [17] capture interaction information in images by measuring relative location information between boxes. Xu *et al.* [137] utilize semantic regularities for human-object interaction detection in images with knowledge graphs. Gao *et al.* [39] propose an instance-centric attention module that learns to dynamically highlight regions in an image conditioned on the appearance of each instance. Prest *et al.* [97] previously studied human-object interaction in actor-centric videos, such as *Drinking* and *Smoking*. In this setting, the person boxes generally cover the object boxes. In the surveillance domain, we aim for proposals of interactivities with unique boxes for persons and objects by focusing on the surveillance domain. Wang *et al.* [127] also investigate interactions in videos, but do so for agent-object animations, while we focus on interactivity detections by proposals.

2.2.3 *Video Surveillance*

Recognition in video surveillance is a long-standing challenge [19, 70, 78, 93, 117, 125, 132, 152]. Surveillance settings are often indoor with an explicit focus on subjects, as exemplified by the recent benchmark of Zhao *et al.* [152]. The works of Maguell *et al.* [105, 106] relates to our work as they focus on tracking loitering activities across multiple surveillance cameras. Our work focuses on capturing interactivity on single surveillance camera, without considering the explicit interactivity class.

The works of Walker *et al.* [125] and Misra *et al.* [93] also relate to our work in that both tackle object localization in space and time. In this work, we focus on outdoor surveillance videos with the ActEv benchmark [6] and we focus on jointly capturing the spatio-temporal localization of subjects and objects in interaction. For spatio-temporal action detection, several datasets have been introduced, such as AvA [50], UCF-Sports [102], and J-HMDB51 [64].

Current datasets are commonly focused on human-centric actions in non-surveillance domains. Only the annotations of subjects is provided, while the spatio-temporal annota-

tions of objects are absent. Hence, we will not consider these datasets for our experiments. Instead, we will set up an interactivity proposal evaluation, including a dataset distilled from the ActEV surveillance video benchmark [6] and interactivity evaluation metrics.

2.3 METHOD

In order to obtain interactivity proposals from an input video, our approach consists of three components: 1). obtaining interactivity candidates, 2). computing interactivityness, and 3). generating interactivity proposals. The overview of our method is sketched in Figure 5. We will describe each component in detail next.

2.3.1 *Obtaining interactivity candidates*

We first generate an over-complete set of interactivity candidates, where each candidate denotes a pair of subject and object trajectories that potentially interact. Due to the possibly overwhelming number of subjects and objects in a surveillance video, evaluating all possible subject and object pairs is infeasible. Physically, a subject can only interact with an object when they are close enough at some point in time. Hence, in most cases, the interactivity only happens when the subject and the object are in close contact with each other.

Suppose we have obtained N subject trajectories and M object trajectories in a video. Each trajectory has consecutive bounding boxes, *e.g.*, the subject trajectory $t_s = \{b_s^1, b_s^2, ..., b_s^n\}$ has n boxes and the object trajectory $t_o = \{b_o^1, b_o^2, ..., b_o^m\}$ has m boxes. A box $b \in \mathbb{R}^4$ is denoted by the leftmost, topmost, rightmost, and bottommost coordinates. For each frame f in the video, we calculate the Intersection over Union (IoU) between subject box $b_s^f \in t_s$ and object box $b_o^f \in t_o$. If they overlap with each other, *i.e.*, their IoU score is larger than zero at any point in time, we deem the pair as a potential interactivity. In addition, we compute a union box that tightly unifies the subject and object boxes as follows:

$$\begin{aligned} b_u^f = \Big(& \min(b_s^f[0], b_o^f[0]),\ \ \min(b_s^f[1], b_o^f[1]), \\ & \max(b_s^f[2], b_o^f[2]),\ \ \max(b_s^f[3], b_o^f[3]) \Big). \end{aligned} \tag{2.1}$$

We add the union boxes to the subject-object pairs and obtain k interactivity candidates, each consisting of a triplet of spatio-temporal trajectories, *e.g.* for temporal length k candidate c is denoted as $c = \{(b_u^1, b_s^1, b_o^1), (b_u^2, b_s^2, b_o^2), ..., (b_u^k, b_s^k, b_o^k)\}$.

This procedure is performed for test videos to obtain an initial pool of candidates. During training, we use ground truth trajectories of subjects and objects that are known to interact. The interactivity label itself is ignored, only the trajectories are used.

2.3.2 *Interactivity network*

Given a subject-object pair from our candidate pool, we need to detect whether this pair has any interactivity. If so, we also want to know when it starts and ends. Here we train a binary classifier to estimate the interactivity likelihoods, called interactivityness, for each

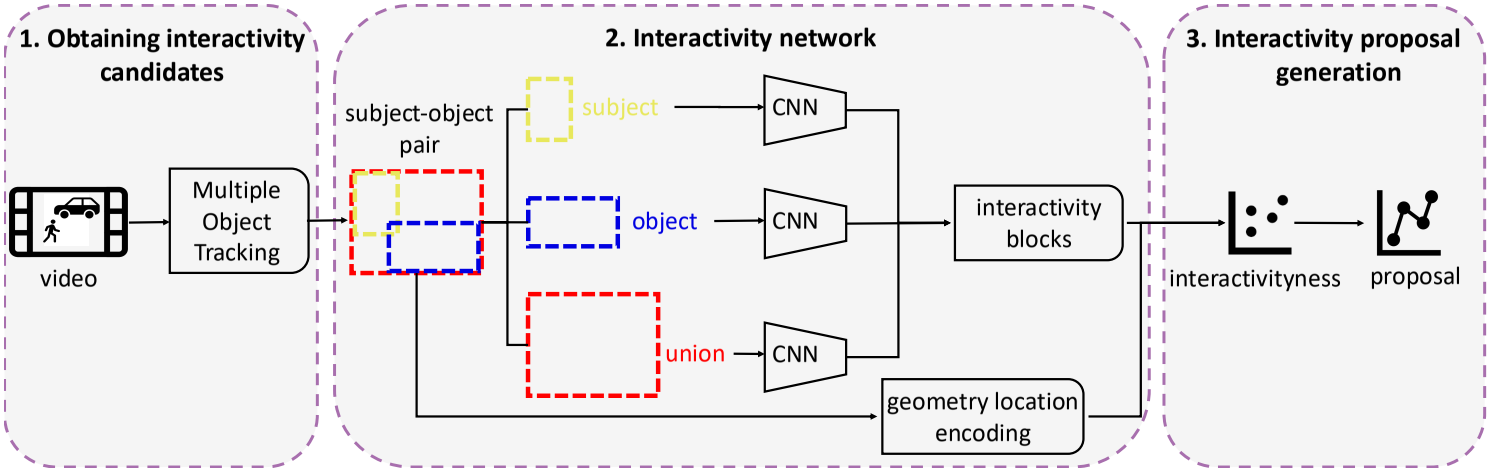


Figure 5: **Method overview.** During testing, we first obtain interactivity candidates by detecting and tracking subjects and objects in a surveillance video. For each frame of each subject-object pair, we input a subject-, object- and union-box to our interactivity network and obtain their interactivityness. Finally, we group continuous regions with high interactivityness to generate spatio-temporal interactivity proposals.

triplet of boxes in each frame of the pair. The frame-level interactivityness scores will be used to generate our final spatio-temporal interactivity proposals.

The main idea of our method is to capture interaction information to aid recognition. We achieve the goal in two ways: (1) We propose the interactivity block, an attention mechanism to compute interactions between the subject, object and union box features. The union box provides spatial contextual information, which is beneficial to recognize interactivity. (2) We encode the geometric relation between the subject and object. The relative positions of subjects and objects change over time and therefore provide useful information.

INTERACTIVITY BLOCK. In surveillance videos, the subjects and objects are usually small due to the high camera position. So the context information around subject and object is important to capture. At the same time, the network should focus on the subject and object during feature extraction. Therefore, the interactivity block should use union features to support subject and object features. Inspired by the non-local operation in action recognition [131], we design an interactivity block to capture small region features (namely subjects and objects) and context region feature (their union). We use two interactivity blocks: one to capture the attention between the subject features and the union features, and one for the attention between the object features and the union features. From the above we know a subject-object pair is composed of continuous triplet boxes $c = \{(b_u^1, b_s^1, b_o^1), (b_u^2, b_s^2, b_o^2), ..., (b_u^k, b_s^k, b_o^k)\}$. For each frame, the three boxes are first fed to a backbone convolutional neural network to extract features. For frame f, we obtain three box features: union box features F_u^f, subject box features F_s^f and object box features F_o^f. The three features then form the input to the interactivity block. Let $F_c^f = (F_s^f, F_o^f, F_u^f)$ denote the combined feature set, then the two individual blocks are given as:

$$\mathrm{IB}_s(F_c^f) = c_1(sm(c_2(F_s^f)^T \times c_3(F_u^f)) \times c_4(F_u^f)) + F_s^f, \tag{2.2}$$

$$\mathrm{IB}_o(F_c^f) = c_1(sm(c_2(F_s^f)^T \times c_3(F_u^f)) \times c_4(F_u^f)) + F_o^f. \tag{2.3}$$

Here c_1, c_2, c_3, c_4 are convolutional layers with kernel size 1×1 and *sm* denotes the softmax function. The output dimensions of $c1, c2, c3, c4$ are 512. We also incorporate Dropout [113], Rescaling, Layer Normalization [7] and matrix transposition operations. The two interactivity blocks' convolutional layers share weights during training. The two blocks are combined as follows:

$$\mathrm{IB}(p) = \mathrm{IB}_s(p) + \mathrm{IB}_o(p). \tag{2.4}$$

The details of the interactivity blocks are illustrated in Figure 6. Interactivity block operations do not change the dimensionality of input feature. The dimensionality of input features F_s, F_o, F_u are all $\mathbb{R}^{C\times H\times W}$, the output feature $\mathrm{IB}(p)$ remains the same.

With the interactivity block, we force the network to focus on both the subject and the object. At the same time, useful contextual information is provided. The output of the function $\mathrm{IB}(p)$ is fed to an average pooling layer with kernel size 2, resulting in $F_p^f \in \mathbb{R}^C$.

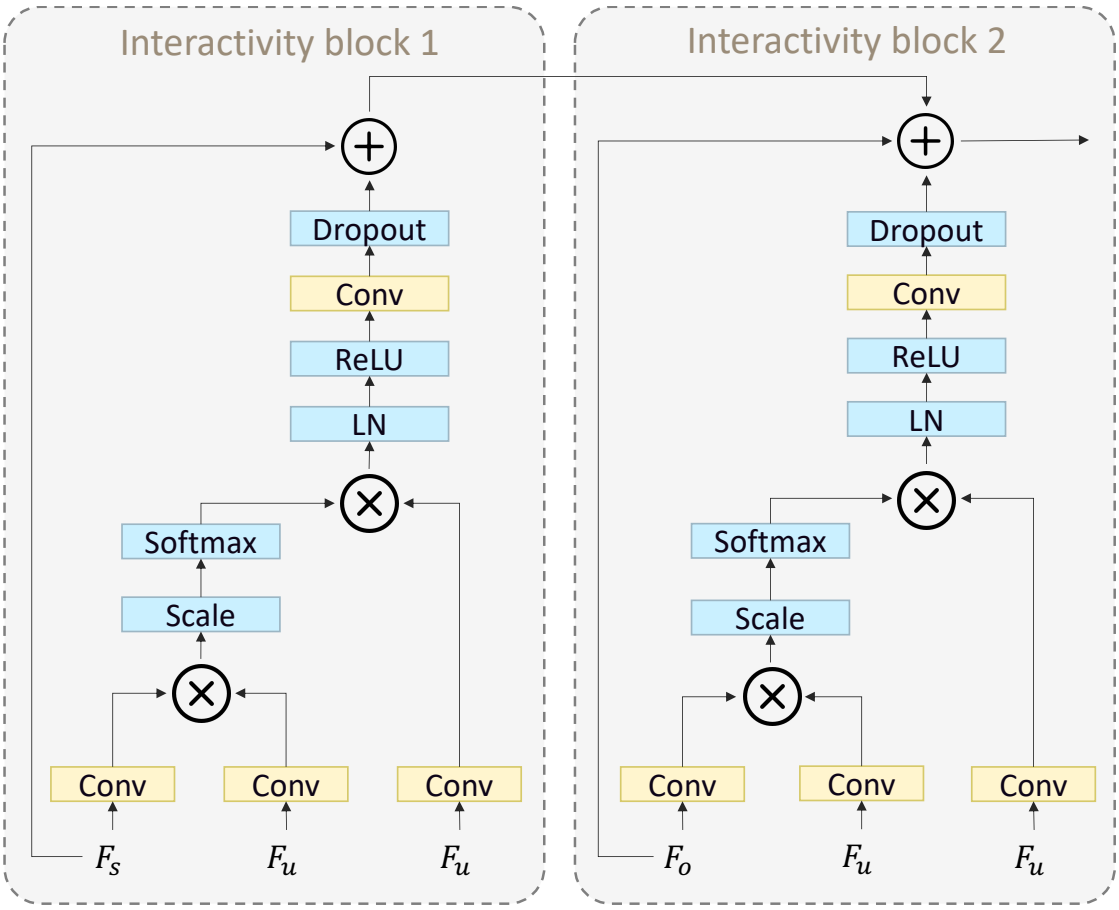


Figure 6: **Interactivity block details**. The two interactivity blocks share convolution layer weights with each other. The input are subject box feature f_s, object box feature f_o and union box feature f_u. Here $\oplus$ denotes element-wise sum and $\otimes$ denotes matrix product. LN is short for Layer Normalization.

GEOMETRIC LOCATION ENCODING. The aim of geometric location encoding is to capture the relative distance between the subject and object. Inspired by object detection in [58], we encode the relative geometric location in a subject-object pair using Eq. 2.5. For ease of notation, we now write each box using the topleft coordinate and width and height, *i.e.* the subject box in f is denoted as (x_s, y_s, w_s, h_s) and the object box as (x_o, y_o, w_o, h_o), we compute the following geometry location features $F_g^f \in \mathbb{R}^8$:

$$\begin{aligned} F_g^f = \Big[& \log\left(\frac{|x_s - x_o|}{w_s}\right), \log\left(\frac{|y_s - y_o|}{h_s}\right), \log\left(\frac{w_s}{w_o}\right), \\ & \log\left(\frac{h_s}{h_o}\right), \log\left(\frac{|x_o - x_s|}{w_o}\right), \log\left(\frac{|y_o - y_s|}{h_o}\right), \\ & \log\left(\frac{w_o}{w_s}\right), \log\left(\frac{h_o}{h_s}\right)\Big]. \end{aligned} \tag{2.5}$$

We then concatenate F_p^f and F_g^f and score the feature:

$$s = \sigma\left(\left[F_p^f; F_g^f\right]\right), \tag{2.6}$$

where σ denotes the sigmoid classification and $[;]$ denotes the concatenate operation along channel dimension to get a representation of dimensionality $C + 8$.

INTEARCTIVITYNESS. The aim of the classification head is to output an interactivityness, a score that indicates the possibility of interaction happening in this triplet of boxes. During training, we first rely on a temporal sliding window along subject-object pairs to generate spatio-temporal interactivity proposal candidates. Then we calculate the temporal Intersection over Union (tIoU) between proposal candidates and ground truths. We collect two types of proposal samples: (1) positive proposals, *i.e.* those overlap with the closest ground truth with at least 0.5 tIoU; (2) negative proposals, *i.e.* those that do not overlap with any ground truth. Due to the sparsity of ground truth proposals, the number of negative proposals is much higher than the number of positive proposals. We adopt the weighted cross-entropy loss function to deal with this class imbalance:

$$\mathcal{L} = -\omega_y\Big(y\log(s) + (1-y)\log(1-s)\Big), \tag{2.7}$$

where s denotes the interactivityness output from Eq. 2.6, y the ground truth label, and ω_y the class-dependent weight used for balancing the positive and negative samples.

2.3.3 *Interactivity proposal generation*

For a subject-object pair, our network provides an interactivity score per frame. To generate spatio-temporal interactivity proposals, we rely on the 1D-watershed algorithm [103]. The main idea is to find continuous temporal segments with high interactivityness to generate proposals. The watershed algorithm was originally used as a segmentation method and later for temporal action proposal generation [151]. We first feed the boxes from the automatically computed candidate pairs to obtain frame-level interactivityness. Then, we regard the interactivityness score as a 1D terrain with heights and basins. This method floods water on this terrain with different *"levels"* (γ), resulting in a series of *"basins"* filled with water, named by $G(\gamma)$. Each obtained basin corresponds to a segment with high interactivityness. Starting from the initial basins, we merge consecutive basins until their length is above a temporal threshold τ. We uniformly sample τ and γ with step 0.05. By using multiple values for the two thresholds, multiple sets of regions are generated. We average the interactivityness for each region as the proposal score. We repeat this procedure for all selected pairs of subjects and objects. Finally, we apply non-maximum suppression on all generated proposals to remove redundant proposals. The final output is a set of spatio-temporal interactivity proposals for a video.

2.4 EXPERIMENTAL SETUP

2.4.1 *KIEV dataset*

To accommodate the new task of spatio-temporal interactivity proposals, we have distilled a subset from the NIST TRECVID ActEV (Activities in Extended Video) dataset, a collection of surveillance videos with spatio-temporal annotations for objects and subject [6]. ActEV is an extension of the VIRAT dataset [94]. Since not all actions in ActEV are

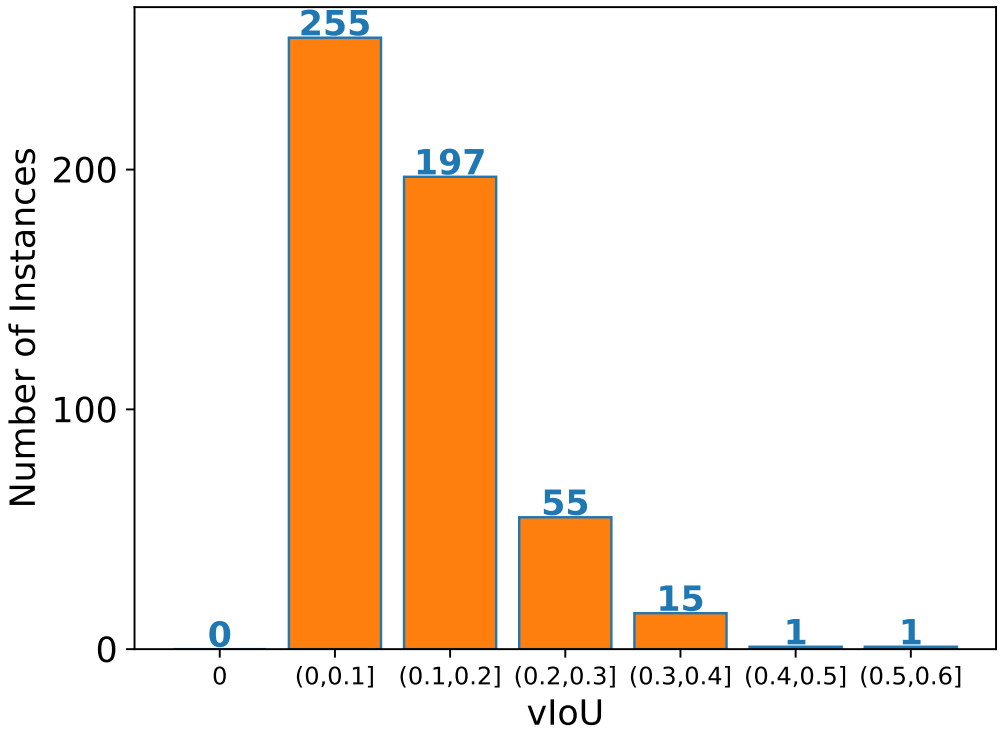


Figure 7: Histogram of vIoU between subject trajectory and object trajectory in interactivity proposal instances of KIEV. For all interactivity instances, the subjects overlap with the objects. Most overlap with vIoU from 0 to 0.2. This enforces our choice of generating interactivity candidates based on overlap.

interactions, we leverage a subset of ActEV that explicitly focuses on interactivities and call this the KIEV (Key Interactivities in Extended Video) dataset. KIEV includes high-resolution surveillance videos that are 1080p or 720p. In KIEV, the subject is a person and the object could be a person, vehicle or door. We select nine key interactivities from ActEV, namely *Closing, Closing Trunk, Entering, Existing, Loading, Opening, Opening Trunk, Unloading* and *Person Person Interaction*. Note that we do not use the interactivity labels in our approach, we are class-agnostic and are merely interested in recognizing their spatio-temporal locations. The training set has a duration of 2 hours and 17 minutes, divided over 51 long videos. The average size of bounding boxes in the training set is 264×142, only 2.6% of the pixels in any given image. The validation set has a duration of 1 hour and 47 minutes, divided over 47 long videos.

2.4.2 *Implementation details*

OBJECT DETECTION AND TRACKING. We use Faster R-CNN [101] with a ResNet-101 [54] backbone with dilated convolutions and feature pyramids [83] for person and vehicle detection. We use the model provided by [19]. The model is trained on the ActEV training set [6]. We apply this model on the unseen KIEV validation frames to obtain vehicle and person boxes. We rely on the Deep SORT tracking algorithm [133],

to generate person and vehicle trajectories. During the tracking procedure, we use the boxes and Region of Interest [53] features from the detection model to link detected subjects and objects into trajectories.

SUBJECT-OBJECT PAIRING. When pairing subjects and objects, we temporally extend each pair with three seconds in both directions. The temporal context is beneficial for recognizing interactivities. We also remove pairs whose duration is shorter than one second.

INTERACTIVITY NETWORK. We use the BN-Inception model provided by [151] as the feature extraction backbone. The model is pre-trained on ImageNet [30]. The interactivity network is inserted before the global average pooling layer. We use the features after the global_pool layer, whose dimensionality is 1024×7×7. After spatially pooling the feature from the interactivity network, we concatenate them with the geometric features and obtain a 1032-dimensional representation. The backbone, interactivity network, and interactivityness classifier are jointly optimized on the KIEV training set. All boxes are resized to 224×224 to meet the input dimension of BN-Inception. We train our model for 100 epochs using Adam with learning rate 1e-5, exponential decay rate 0.9, decay rate 0.999, and weight decay 5e-4. We follow [151] to set other parameters.

PROPOSAL GENERATION. A 1D Gaussian filter with kernel size 3 is applied to smooth the interactivityness sequence. We then apply non-maximum suppression with temporal overlap threshold 0.7 to filter out overlapping proposals.

2.4.3 *Evaluation metrics*

We consider three evaluation metrics, which measure the temporal, spatial, and spatio-temporal proposal quality.

AVERAGE TEMPORAL RECALL. The first metric, Average Temporal Recall (ATR), measures the temporal alignment between proposals and ground truth interactivities. This metric is commonly used for temporal action proposals, *e.g.* [40, 41, 151]. A proposal is a true positive if its temporal intersection over union (tIoU) with a ground truth is greater than or equal to a given threshold. ATR is the mean of all recall values using tIoU between 0.5 to 0.9 (inclusive) with a step size of 0.05. AN is defined as the total number of proposals divided by the number of videos in the validation set. We report ATR_{25}, ATR_{50}, as well as the AUC (Area Under Curve) to see how well the proposal method works across all thresholds for number of proposals per video.

AVERAGE SPATIAL RECALL. The second metric, Average Spatial Recall (ASR), is adapted from the AVA dataset [50]. We compare predicted boxes in each frame with ground truth boxes. If their overlaps are above a threshold of 0.5, we regard the predicted box as a true positive. We evaluate frame by frame to get the final recall.

interactivity block	geometric encoding	**Average Temporal Recall**		
		ATR_{25}	ATR_{50}	AUC
		6.9	14.2	6.9
✓		10.9	15.7	10.1
	✓	10.6	15.5	9.6
✓	✓	**12.4**	**19.0**	**11.3**

Table 1: **Ablating the interactivity network** based on temporal average recall (%). Both the interactivity block and the geometric encoding aid the proposal quality. Their combination works best. The results prove the efficiency of our method.

SPATIO-TEMPORAL RECALL. The third metric, Spatio-Temporal Recall, evaluates the spatio-temporal quality of an interactivity, inspired by [107]. To match a predicted interactivity proposal (t_s^p, t_o^p) to a ground truth interactivity (t_s^g, t_o^g), we require that the bounding-box trajectories overlap s.t. $\text{vIoU}(t_s^p, t_s^g) \geq 0.5$ and $\text{vIoU}(t_o^p, t_o^g) \geq 0.5$ and the proposal is not closer to another unmatched ground truth interactivity. The term vIoU refers to the voluminal Intersection over Union and is calculated as vIoU = (tube of overlap) / (tube of union). We report the spatio-temporal recall for the top 25 proposals (STR_{25}) and top 50 proposals (STR_{50}).

2.5 RESULTS

We consider three experiments: (i) we ablate the effectiveness of our interactivity networks, (ii) we assess the effect of automatic trackers over ground truth spatial locations, and (iii) we compare to other proposal methods.

2.5.1 *Ablating the interactivity network*

In the first experiment, we evaluate the two core components of our interactivity network: the interactivity block and the geometric encoding. The baseline method does not contain these two components. For the baseline we sum the subject feature, object feature and union feature obtained from CNN backbone together. Then we input the summed feature into classifier. We use the Average Temporal Recall as the evaluation metric. We rely on ground truth person and vehicle tubes as the subject and object trajectories to eliminate the influence of the tracker.

INTERACTIVITY BLOCK. Table 1 shows the effect of the interactivity block on the quality of the temporal interactivity proposal. We report the ATR_{25}, ATR_{50}, and AUC. The interactivity block improves ATR_{25} by 4 percent points, ATR_{50} by 1.5 and AUC by 3.2. This result indicates the interactivity block is an important element of the approach; capturing context around subjects and objects matters.

GEOMETRY ENCODING. In Table 1, we also show the effect of the geometric encoding, as well as its combination with the interactivity block. After adding geometry

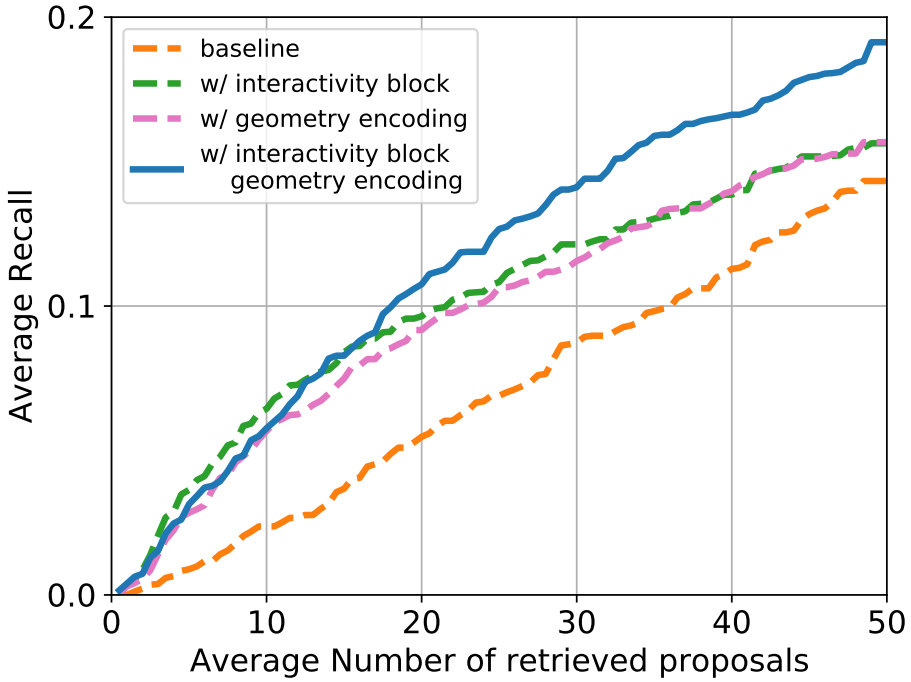


Figure 8: **Ablating the interactivity network** by increasing retrieved proposals. When using both the interactivity block and the geometry encoding we obtain best average recall.

encoding the AR_{25} is improved by 3.7, AR_{50} by 1.3, and AUC by 2.7. Combining the interactivity block with the geometric encoding is most beneficial and results in improvements on all three metrics. Evidently, encoding the geometric relations between subjects and objects aids the quality of interactivity proposals. Figure 8 shows the Temporal Average Recall as a function of the average number of retrieved proposals per video. The interactivity block and the geometric encoding improve the proposal quality scores. For their combination, the largest improvements are obtained when more proposals are generated. We conclude that the interactivity block and geometric encoding are important components of our method and we will report further experiments with their combination.

2.5.2 *Effect of automatic tracks*

Next, we evaluate the effect of using automatic tracks for subjects and objects on the interactivity proposal quality. We report both the temporal proposal quality (ATR) and spatio-temporal quality (STR) and show results in Table 2.

When evaluating the temporal dimension only, we find that automatic tracks are competitive with ground truth subject and object tubes. Indicating our method is temporally robust to noise in the spatial locations of subjects and objects. Table 2 also shows the spatio-temporal proposal quality is directly impacted by the switch from ground truth to automatic tracks. This is not surprising, since the spatio-temporal evaluation metric is very strict in its spatial evaluation; both the subject and object boxes need sufficient

Tracker	Temporal			Spatio-Temporal	
	ATR_{25}	ATR_{50}	AUC	STR_{25}	STR_{50}
ground truth	12.4	19.0	11.3	20.0	23.3
automatic	11.6	17.6	10.8	6.3	7.8

Table 2: **Effect of automatic tracks** on temporal and spatio-temporal proposal quality. For temporal recall, switching from ground truth to automatic trajectories has minimal effect on performance. For spatio-temporal recall, the scores naturally have a larger drop. Automatic tracks are robust enough for temporal proposal quality, but not for spatio-temporal quality.

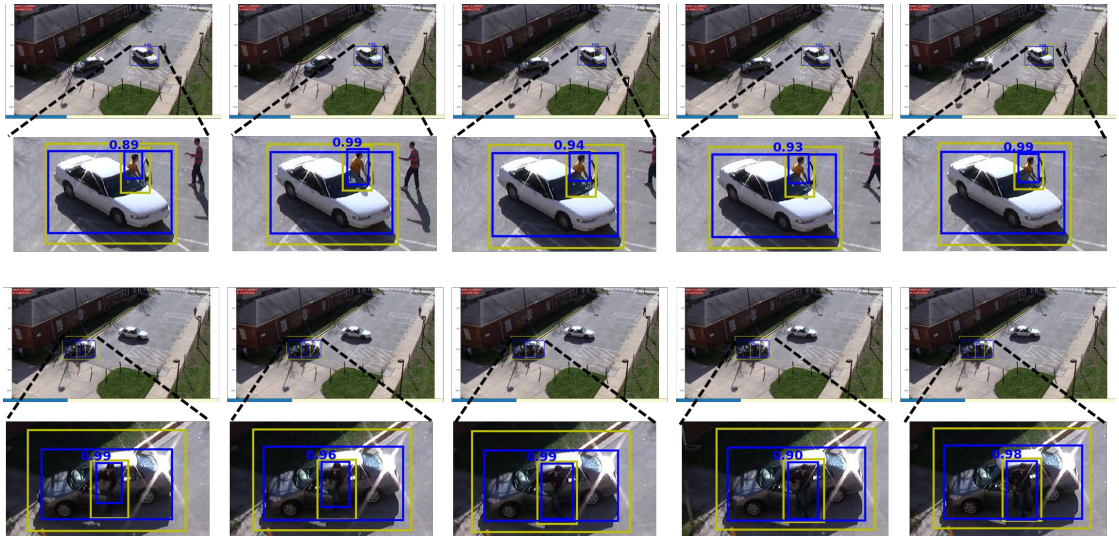
overlap. In Figure 9, we show a number of example proposals when using automatic trackers for the subject and object trajectories. The qualitative results indicate the difficult nature of the problem of finding spatio-temporal interactivities. Due to occlusions and tiny object sizes, there are some missed detection of interactivity in this dataset, as visualized in Figure 9c. Improved detection will positively affect interactivity proposal generation.

2.5.3 *Comparison to prior work*

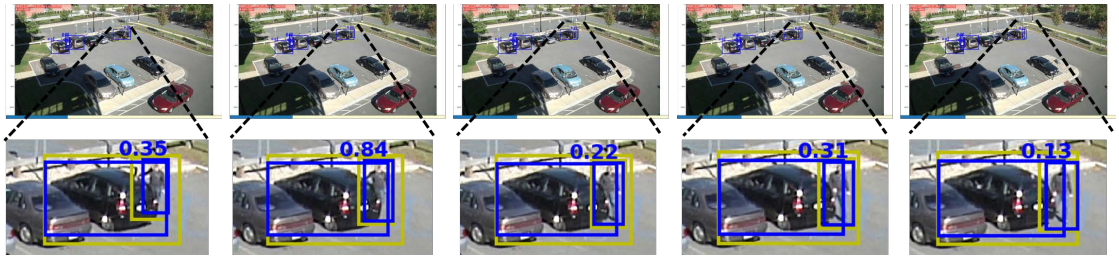
In the third experiment, we compare our approach to several baselines from both the temporal and spatio-temporal action proposal literature, to show that proposing spatio-temporal interactivity locations can not be achieved by existing action proposal methods.

BASELINES. We compare to two temporal proposal baselines and one spatio-temporal baseline. The first temporal proposal baseline is TAG from Zhao *et al.* [151], which proposes temporal regions based on actionness grouping. The second temporal proposal baseline is TURN-TAP from Gao *et al.* [41], which is based on sliding windows. The spatio-temporal baseline is by Gleason *et al.* [47], who introduce a spatio-temporal proposal cuboid approach for actions. For a fair comparison, the input object boxes are the same as our approach.

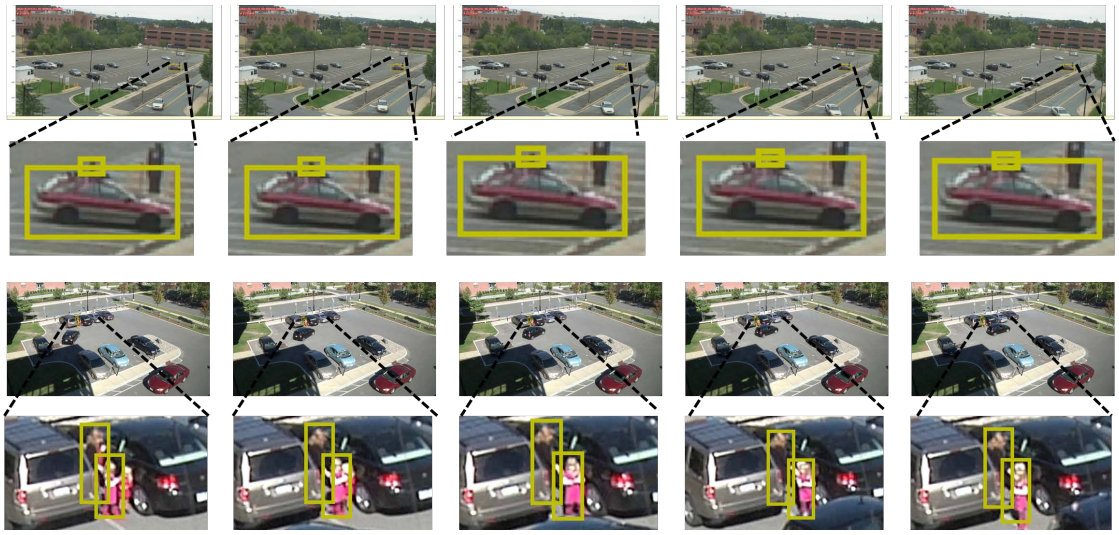
TEMPORAL COMPARISON. Since temporal action proposal methods only provide the start and end times, we first compare our proposals to all baselines using the temporal quality metrics. The results are shown in Table 3 and Figure 10. Our approach performs better than all baselines. In comparison to the best scoring baseline of Gao *et al.* [41], our method improves the ATR_{25} by 3.5, the AR_{50} by 5.2, and the AUC by 3.4. The approaches of Zhao *et al.* [151] and Geo *et al.* [41] fail to generate efficient proposals in this setting because they take the whole frame as input. Since interactivities are only a small part of the video spatially, their representations hardly capture the precise interactions, as expected. These temporal action localization methods fail to solve the interactivity proposal problem. They are capable of localizing temporal boundaries but ignore spatial boundaries. Our approach operates locally in space, which allows for a better estimation of interactivities in time. The approach of Gleason *et al.* [47]



(a) Successful cases.



(b) Imperfect case.



(c) Failed cases.

Figure 9: **Qualitative results.** (a). The top two examples show successful cases, where the proposal highly overlaps in space and time with the ground truth. From top to bottom the interactivities are Entering, Exiting, Closing, Entering and Person Person Interaction. Note that we do not output labels. Here the labels are only for clarifying. The bottom two examples show failure cases, (b). occlusion and (c). small object sizes either result in a low interactivityness or even missed subject and object trajectories. These failure cases highlight the difficult nature of finding interactivities in outdoor settings.

Method	ATR_{25}	ATR_{50}	AUC
Zhao *et al.* [151]	0.0	0.0	0.0
Gleason *et al.* [47]	1.4	1.6	1.2
Gao *et al.* [41]	8.1	12.4	7.4
Ours	**11.6**	**17.6**	**10.8**

Table 3: Temporal comparison of our interactivtyness proposals versus regular action proposals. Our method outperforms alternatives.

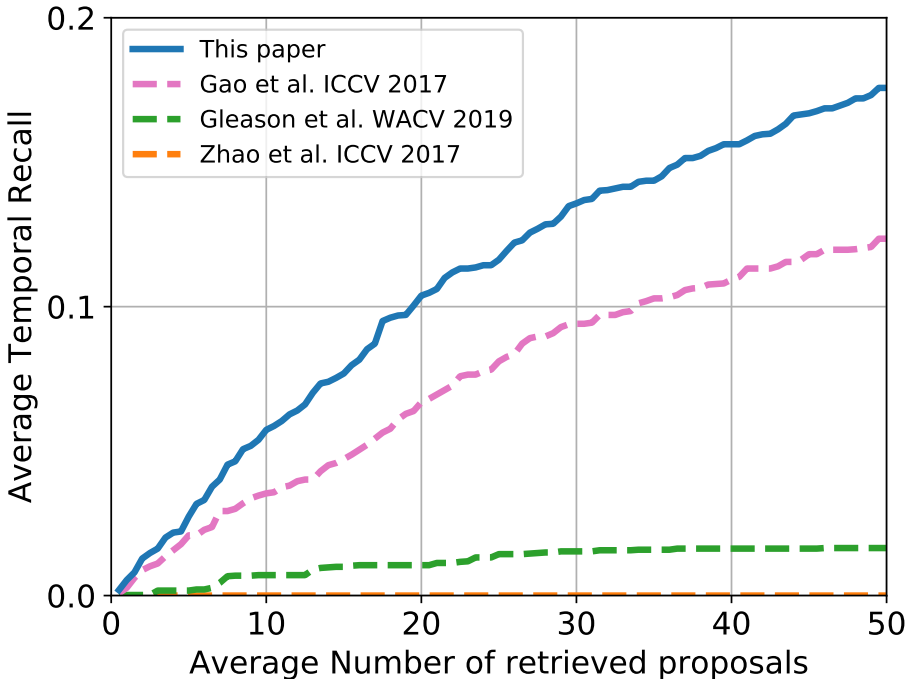


Figure 10: **Temporal comparison** of interactivtyness proposals versus regular action proposals under varying number of retrieved proposals. Modeling interactivityness rather than activity is beneficial.

does operate locally in space, but does not explicitly capture contextual and geometric relations between subjects and objects, which results in lower recall scores.

SPATIO-TEMPORAL COMPARISON. In Table 4, we also compare our approach to Gleason *et al.* [47] with respect to the spatio-temporal proposal quality. The results show that spatially, the baseline obtains an ASR of 8.4, while we reach a score of 61.5, a considerable gain. Furthermore, the spatio-temporal recall at both 25 and 50 proposals per video is 0 for the baseline, compared to 4.8 and 6.3 for our approach. The reason for this gap in performance is because the baseline generates cuboid-style proposals, leading to coarse spatial localization of subjects and objects. The cuboid-style proposals have low IoUs compared to trajectory-style ground truths. In our evaluation, we care about a precise dynamic alignment in space and time for subjects and objects. Our approach yields more accurate spatio-temporal interactivity proposals, be it the overall

Method	ASR	STR_{25}	STR_{50}
Gleason *et al.* [47]	8.4	0.0	0.0
Ours	**61.5**	**4.8**	**6.3**

Table 4: **Spatio-temporal comparison** of our interactivityness proposals versus a regular action proposal in terms of Recall (%). Explicitly modeling interactivity results in better spatio-temporal localization.

spatio-temporal recall is modest. Compared to Gleason *et al.* [47] we conclude that our approach is better equipped to find interactivities more precisely in space and time.

2.6 CONCLUSION

This chapter introduces interactivity proposals for video surveillance. Rather than focusing on the actions of the subject only, our proposals capture the interplay between subjects and objects in space and time. To that end, we propose a network to compute interactivityness between subjects and objects from which we generate class-agnostic proposals. We evaluate the proposals on an interactivity dataset with new overlap metrics, where experiments show the improvement of our approach over traditional temporal and spatio-temporal action proposal methods. Overall, the results are far from perfect, indicating the challenging nature of the problem. To encourage further progress on recognizing interactivity proposals we make the dataset split, evaluation metrics, and code publicly available.

3

SOCIAL FABRIC: TUBELET COMPOSITIONS FOR VIDEO RELATION DETECTION

3.1 INTRODUCTION

To understand what is happening where in videos, it is necessary to detect and recognize relationships between individual instances. Effectively capturing these relationships could improve captioning [139], video retrieval [112], visual question answering [4], and many other visual-language tasks. In this chapter, we strive to classify and detect the relationship between object tubelets appearing throughout a video as a ⟨*subject-predicate-object*⟩ triplet, like ⟨*dog-chase-child*⟩ or ⟨*horse-stand_behind-person*⟩.

Shang *et al.* [107, 108] pioneered this challenging problem by their definition of video datasets with dense bounding box annotations, temporal bounds, and relationship-triplet labels. Following their guidance, a leading approach to date is to generate proposals for individual objects on short video snippets, encode the proposals, predict a relation and associate the relations over the entire video, *e.g.* [98, 114, 136]. To better detect long-term interactions, Liu *et al.* [85] forego the need for snippets by first localizing individual object tubelets throughout the entire video, filtering out unlikely pairs, and predicting predicates for the remaining ones. Different from all these existing works on video relation prediction, which treat object proposals or tubelets as single entities and model their relations *a posteriori*, we propose to classify and detect predicates for pairs of object tublets *a priori*.

Considering objects as tubelet pairs from the start requires an encoding that enables us to localize and classify interactions from the pool of all co-occurring object tubelets across all timespans in a video. This is reminiscent of many classical problems in computer vision that need to aggregate spatial, *e.g.* [5, 63, 111, 123], temporal, *e.g.* [80,129,144] or spatio-temporal, *e.g.* [44,45,91] primitives into a common representation. We take inspiration from ActionVLAD by Girdhar *et al.* [45], which encodes actions as a composition of local action primitives to capture the entire spatio-temporal extent of actions. In this chapter, we also learn to encode local spatio-temporal video features in a compositional manner. Different from ActionVLAD, which operates on an entire video, our Social Fabric encoding operates on tubelet pairs, *i.e.* on inputs from multiple object tubelets and multiple modalities, with a set of interaction primitives that is dynamically learned during video relation training. Social Fabric captures information across the entire scope of tubelet pairs, which is especially beneficial when interactions last long. See Figure 4 for an illustrative example. Code is available at `https://github.com/shanshuo/social-fabric`.

We make three contributions. First, we propose to classify and detect video relations for pairs of object tubelets from the start. Second, we introduce Social Fabric, a compositional encoding suited for multi-tubelet and multi-modal inputs. The interaction primitives that form the encoding are learned and updated dynamically, akin to the NetVLAD layer from Arandjelović *et al.* [5] for visual place recognition. Third, to leverage the Social Fabric, we propose a two-stage network for video relation classification and detection. In the first stage, we localize interactions by training Social Fabric to propose tubelet pairs that are likely interacting. In the second stage we use the Social Fabric to simultaneously fine-tune and learn to predict predicate labels for the tubelets. Experiments on the benchmarks for video relation detection of Shang *et al.* [107, 108] show the benefits of our approach, especially when interactions are long and complex. Social Fabric outperforms alternative video encodings and our two-stage architecture sets a new state-of-the-art for both video relation classification and detection. Besides classification and detection, we show that our encoding enables searching for relations in videos by providing primitive-examples as queries.

3.2 RELATED WORK

Image relation detection. Visual relation recognition has a long-standing tradition for static images [18, 49, 51, 59, 61, 77, 79, 90, 126, 142]. Besides recognizing visual relationships between objects, Chao *et al.* [17] introduce the problem of detecting human-object interactions in static images and contribute a corresponding dataset. It inspired many to contribute to human-object-interaction detection, *e.g.* [29, 77, 126, 130, 137]. Li *et al.* [77], for example, learn the knowledge between human and object categories from the provided datasets and use this knowledge as a prior while performing detection. Wan *et al.* [126] introduce a pose-aware network that employs a multi-level feature strategy. Where image-based relation detection requires two boxes (subject and object) and a predicate, we aim to perform video-based relation detection, which requires us to also localize and track subjects and objects over time.

Snippet relation detection. Many before us have investigated relation detection in videos [14, 32, 73, 85, 98, 107, 108, 114–116, 121, 136, 153]. Relations in videos provide additional temporal information, important for interactions such as pushing or pulling a closed door. Shang *et al.* [108] pioneered this problem and introduced the ImageNet-VidVRD dataset, the first video relation detection benchmark in which all video relation triplets, along with their object and subject trajectories, are labelled. Building on the foundational work of Shang *et al.* [108], Tsai *et al.* [121] propose a gated spatio-temporal energy graph using conditional random fields to model video relations. In a similar spirit, Qian *et al.* [98] built a spatio-temporal graph between adjacent video snippets and used multiple layers of graph convolutional networks to pass messages between nodes. Shang *et al.* [107] later introduced VidOR, the largest video relation detection benchmark to date. On this dataset, Sun *et al.* [115] utilize language context features along with spatio-temporal features for predicate prediction.

All the aforementioned methods adopt a three-stage framework. A video is first to split into short snippets and subject/object tubelets are generated per snippet. Then, short-term relations are predicted for each tubelet. The subject/object proposals are

obtained in the short snippets using an image object detector and tracker [98, 108, 121]. In the second stage, spatio-temporal features of each pair of object tubelets are extracted and used to predict short-term relation candidates. Xie *et al.* [136] combine a wide variety of multi-modal features for each pair to predict the relations with impressive relation classification accuracy. In the third stage, the short-term relation proposals are merged by a greedy relational association algorithm. Su *et al.* [114] maintain multiple relation hypotheses during the association process to accommodate for inaccurate or missing proposals in the earlier steps. Instead of treating the relations independently at the various analysis stages, we consider the object tubelets as interacting pairs from the start.

Proposal relation detection. Liu *et al.* [85] are the first to avoid the need to split videos into snippets. In the first stage, they generate object tubelets for the whole video. The second stage refines the tubelet-features and finds relevant object pairs using a graph convolutional network. The third stage focuses on predicting the predicates between related pairs. In this manner, interactions can be detected without a need for snippet splitting. Like Liu *et al.*, we also avoid the need for snippets. Different from them, we view subjects and objects as interactions from the start. As a result, we only need two stages, one for interaction proposal generation from the tubelet pairs and one for predicting the appropriate predicate. At the core of both our stages is the Social Fabric, which allows us to encode a set of interaction primitives, like the ones in Figure 4, from which we classify and detect different video relations.

3.3 SOCIAL FABRIC ENCODING

The goal of video relation detection is to localize interactions between two entities in space and time. Formally, a spatio-temporal interaction $\mathcal{I}$ is defined as a triplet $\mathcal{I} = \{O_1, P, O_2\}$, with subject tubelet $O_1 \in \mathbb{R}^{4\times(T_2-T_1)}$, object tubelet $O_2 \in \mathbb{R}^{4\times(T_2-T_1)}$ and their relation predicate category P. Here, T_1 and T_2 denote the start and end frame of the interaction and each frame contains box coordinates. To address both video relation classification and detection, we propose a two-stage approach that encodes subjects and objects as pairs from the start. Central to both stages is our Social Fabric encoding for representing compositions of tubelet pairs. Below, we outline how to learn the encoding, how to use it to represent tubelet pairs, and how the encoding relates to existing video encodings.

Learning the encoding. The idea behind the encoding is that a pair of tubelets, which form a video relation triplet, are composed of multiple interaction primitives. These primitives can represent different relations by varying their combinations. For example, let {"approach","run","watch","touch"} denote a set of primitives, then a hugging relation can be represented by {"watch","approach","touch"}, while a chasing relation can be represented by {"run","approach"}. In the object detection and action recognition literature, compositional learning and encoding is well established, with advantages such as sharing components amongst categories *e.g.* [38], efficient and compact encoding *e.g.* [145], and high discriminative ability *e.g.* [68, 72]. By introducing a compositional encoding for video relation detection we share the same benefits and show some examples of the primitives we learned in Figure 11.

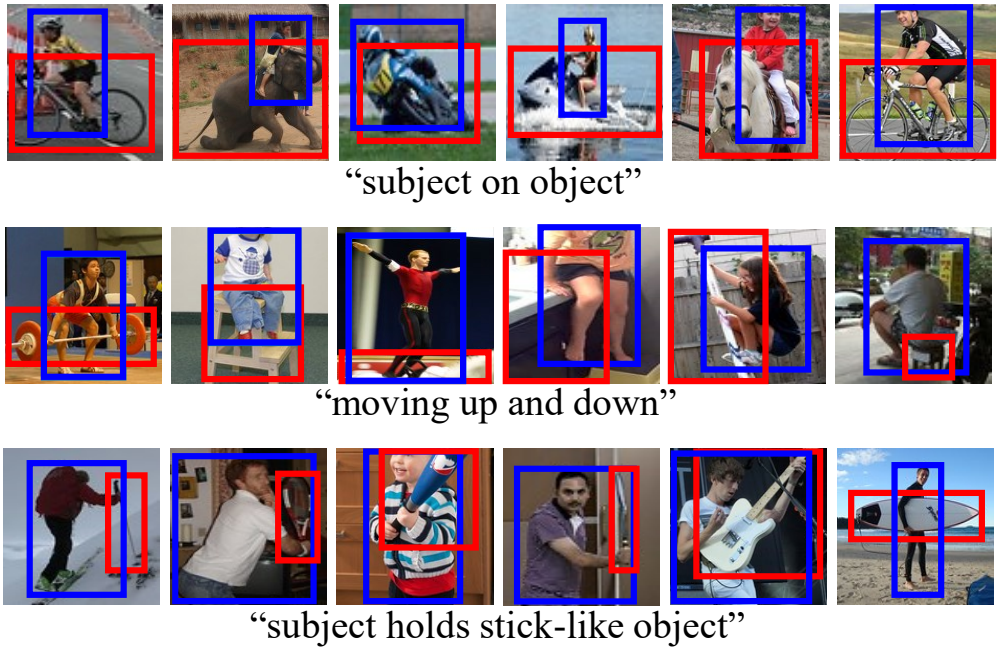


Figure 11: **Interaction primitives** that our Social Fabric encoding learns when trained for multi-modal features. Each row shows several frames from videos that get assigned to one specific primitive. Blue boxes indicate the subject while red boxes denote the object. Here we show some easy-to-interpret primitives.

For each task, we are given a training set of tubelet pairs, denoted as $\mathcal{R}$, where the input representation of each tubelet pair is denoted as $S_i \subset \mathcal{R} \in \mathbb{R}^{N \times F}$, with N the number of frames of the tubelets and F the feature dimensionality for each frame, denoting the combined subject and object representations. On top of the features, we apply layer normalization [7], followed by a linear layer to obtain embedded representation $R_i \subset \mathcal{R} \in \mathbb{R}^{N \times D}$. In this D-dimensional embedding space, we learn a set $C \in \mathbb{R}^{K \times D}$ consisting of K primitives. The idea behind our encoding is to describe a tubelet pair entirely as a weighted combination of these primitives. So tubelet pair i is encoded with our approach as a concatenation of weighted primitive locations:

$$E_i = [E_{i,1}, \cdots, E_{i,K}], \; E_{i,k} = \sum_{j=1}^{N} z_{ijk} C_k, \tag{3.1}$$

where the weight is inversely proportional to the distance between a local relational feature vector and the primitive:

$$z_{ijk} = \frac{\exp\left[-\beta \left\| R_{ij} - C_k \right\|^2\right]}{\sum_{l=1}^{K} \exp\left[-\beta \left\| R_{ij} - C_l \right\|^2\right]}, \tag{3.2}$$

where $\beta > 0$ denotes a temperature parameter to tune how soft or hard the assignments should be, fixed to $1/\sqrt{D}$ throughout this work. Intuitively, our encoding describes how much a relation is in line with each primitive in C. Each portion $E_{i,k}$ of the encoding forms a line between the primitive C_k and the origin; the stronger the agreement, the

closer $E_{i,k}$ is to the primitive and the more its values contribute to the next layer. The diagram of the Social Fabric Encoding is shown in Figure 12.

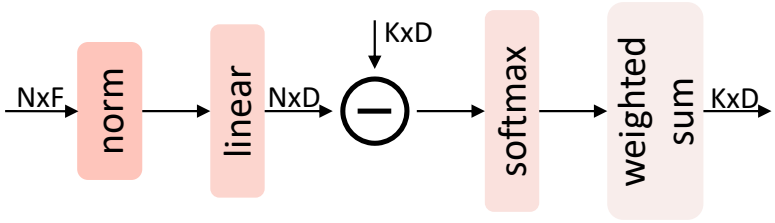


Figure 12: **Social Fabric Encoding**.

On top of the representation E_i, we learn a fully-connected layer classification head, which can be used to determine whether a tubelet pair makes for a good proposal or to predict its predicate using a shallow network head. The layers of the network and the set C are jointly learned during the optimization.

Relation to alternative encodings. A common encoding in video-based representations is average pooling [144]. In our encoding, average pooling is a special case where the codebook contains a single primitive. Average pooling implicitly assumes that the features of the input representation follow a single mode. Video relations, however, consist of multiple interaction primitives that evolve over time. Moreover, these primitives are shared between different relations, which we capture. Encodings such as transformers follow the self-attention architecture, where each feature is a weighted sum of other features [124]. Compared to transformers, our approach provides a fixed-sized representation, important because tubelet pairs are of varying lengths. Other encodings like NetVLAD [5] and ActionVLAD [45] operate on whole images and videos, while residuals between local features and clusters are used to obtain a representation. In contrast, our encoding operates on pairs of spatio-temporal tubelets, accepts multi-modal features, and we directly use the primitives to encode inputs. Lastly, we are the first to rely on a compositional encoding for the task of video relation detection.

3.4 TWO-STAGE VIDEO RELATION NETWORK

We utilize the Social Fabric Encoding to both classify and detect video relations using two stages, rather than the three stages common in the literature. In the first stage, we sift through all combinations of co-occurring tubelets across all timesteps to obtain a set of interaction proposals that likely cover all ground truth video relations. In the second stage, we classify each proposal with a predicate label. An overview of our approach is visualized in Figure 13. Next, we detail both stages and show how to obtain the final classification and spatio-temporal detection results.

Stage 1: Interaction proposals. We initialize the video relation optimization by performing object detection in each frame, followed by linking over time-based on [133]. For a video V, this results in M object tubelets. We consider all unique combinations of tubelets for proposal generation and train a binary classifier to determine interactivityness at the frame-level using a local window around the box pairs in a frame [22]. For the two objects (O_1, O_2) in a tubelet pair and frame f, we consider a neighbourhood of $m/2-1$ frames in both temporal directions of the tubelets. We compute and stack the multi-modal

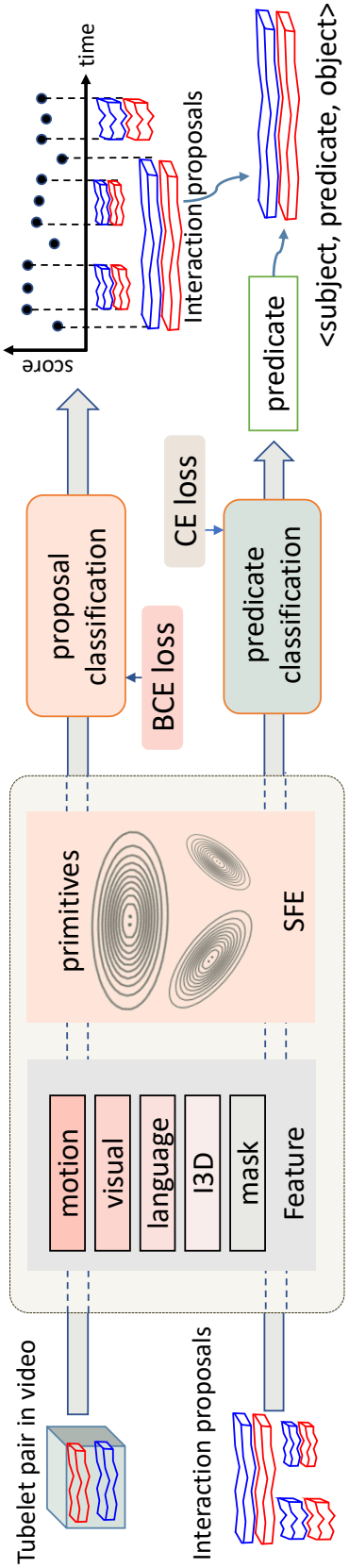


Figure 13: **Two-stage video relation network**. We first obtain interaction proposals and then predicate predictions. Social Fabric Encoding (SFE) is essential to both stages to represent an object tubelet with a composition of interaction primitives. BCE loss and CE loss represent binary cross-entropy loss and cross-entropy loss separately.

features for the windowed tubelet pair, resulting in $R^1_f(O_1, O_2) \in \mathbb{R}^{m \times D}$ for frame f. We feed this as input to Social Fabric, resulting in $E^1(O_1, O_2) \in \mathbb{R}^{K \times D}$. During training, the encoding is used to train a binary classifier to separate potential interactions from non-interactions with a binary cross-entropy loss $\mathcal{L} = \big(y \log(s) + (1 - y) \log(1 - s)\big)$, where s denotes the interactivityness. Simultaneously, the primitives in the Social Fabric are learned. For each frame in a tubelet pair, this results in a score indicating its interactivityness. Over the array of scores over all timesteps of the tubelet pair, we employ a 1D watershed algorithm [22, 103] to generate spatio-temporal interaction proposals. We repeat this procedure for all co-occurring tubelets and combine the outputs per pair into a final set of interaction proposals for a video.

Stage 2: Predicate prediction. Once a video is decomposed into a set of interaction proposals, each consisting of two tubelets with a similar start and end time, we seek to score all proposals for their predicate. For interaction proposal (O_1, O_2), we sample n frames uniformly. For each sampled frame, we extract a single uni-modal or several multi-modal features. Then we stack the features over all frames and obtain $R^2(O_1, O_2) \in \mathbb{R}^{N \times D}$ for this tubelet. This is fed into Social Fabric and the output representation is in $E^2(O_1, O_2) \in \mathbb{R}^{K \times D}$. In stage 2 we fine-tune the Social Fabric trained in stage 1 to accelerate the convergence. After encoding each proposal, we feed the representation into a final linear layer to obtain predicate scores. The predicate prediction is optimized with softmax cross-entropy. After obtaining predicate predictions, we multiply the predicate score and corresponding subject and object scores as the relation triplet prediction score. The subject and object scores are obtained from the tubelet pairs in stage 1. Relation triplets are the predicted results for relation classification. The relation triplet associated with subject and object tubelets acts as the predicted results for relation detection.

Search-by-primitive-example. The Social Fabric encoding is optimized for video relation classification and detection, but is not limited to these tasks. Here, we show how we can also search for spatio-temporal video relations in a collection of videos by querying primitive examples. As input, a user can provide one or more frames with a subject and object performing a basic interaction. We compute the non-temporal features for each input and use it to find the nearest learned primitive. To find the interaction proposal across all videos that best describes the primitive examples, we use the weights from Equation 3.2 to score the relevance of each primitive for an entire proposal. In turn, we simply sum the scores for the few primitives determined by the user and output the interaction proposal with the highest score. As a result, we can search on-the-fly for video relations that are composed of example primitives provided by a user, without the need for search optimization.

3.5 EXPERIMENTAL SETUP

3.5.1 *Datasets*

To evaluate the proposed methods, we perform experiments on ImageNet-VidVRD [108] and Video Object Relation (VidOR) [107].

ImageNet-VidVRD. [108] consists of 1,000 videos, created from the ILSVRC2016-VID dataset [104]. There are 35 object categories and 132 predicate categories. The

Feature type					Relation tagging			Relation detection		
motion	visual	language	I3D	mask	P@1	P@5	P@10	mAP	R@50	R@100
✓					50.97	39.57	31.58	6.14	6.74	8.70
✓	✓				56.89	44.76	34.07	8.93	7.38	9.22
✓	✓	✓			59.24	47.24	35.99	9.54	8.49	10.17
✓	✓	✓	✓		61.52	50.05	38.48	10.04	8.94	10.69
✓	✓	✓	✓	✓	**68.86**	**55.16**	**43.40**	**11.21**	**9.99**	**11.94**

Table 5: **Benefit of multi-modal features** on VidOR. More is better. The increasing gaps indicate Social Fabric effectively captures multi-modal features for relation classification and detection.

videos are densely annotated with relation triplets in the form of ⟨*subject-predicate-object*⟩ as well as the corresponding subjects and objects trajectories. Following [108, 121], we use 800 videos for training and the remaining 200 for testing.

VidOR. [107] contains 10,000 user-generated videos selected from YFCC-100M [118], for a total of about 84 hours. There are 80 object categories and 50 predicate categories. Besides providing annotated relation triplets, the dataset also provides the bounding boxes of objects. The dataset is split into a training set with 7,000 videos, validation set with 835 videos, and a testing set with 2,165 videos. Since the ground truth of the test set is not available, we use the training set for training and the validation set for testing, following [85, 98, 114, 136].

3.5.2 *Implementation and evaluation details*

Tubelet pairing. We first detect all the objects per video frame by Faster R-CNN [101] with a ResNet-101 [54] backbone. The detector is trained on MS-COCO [84]. The detected bounding boxes are linked with the Deep SORT tracker [133] to obtain individual object tubelets. Finally, each tubelet is paired with any other tubelet to generate the tubelet pairs. We use the object trajectories of ImageNet-VidVRD and VidOR adopted in [98, 108, 114, 115] for fair comparison.

Feature extraction. In the video relation literature, features from multiple modalities are commonly used, *e.g.* Sun *et al.* [115] use motion features and language features. Liu *et al.* [85] use motion features, visual features and I3D features. Xie *et al.* [136] use motion features, visual features, language features and location mask features. We consider all features and arrive at motion features, visual features, language features, I3D features, and location mask features. We follow [115] to calculate the spatial location feature as motion features. The visual features are extracted using the detection backbone in Faster R-CNN and followed by an RoI pooling layer. For the language features we use a word2vec module, pre-trained on GoogleNews [92], to encode the subject and object classes into language features with dimension of 600. We use the I3D module from [15] to extract I3D features with fixed dimension of 832. We follow the method of [136] to generate a mask based on the bounding boxes of the subject and object in the tubelet pair.

Two-stage network optimization. The size of the linear layer for embedding representation is D=512. In the first stage, we consider m=30 neighbourhood frames on both

Clusters	1	8	32	64	128
mAP	10.05	10.69	10.91	**11.21**	11.01

Table 6: **Influence of encoding size** on VidOR for relation detection. Using multiple primitives results in a more accurate predicate prediction, where we achieve best performance for 64 primitives.

temporal directions. The interaction proposal generation network is trained for 20 epochs using an SGD optimizer with a mini-batch of 128. We use a fixed learning rate and set its value to 0.01. In the second stage, we sample n=25 frames for each interaction proposal. The predicate prediction network is trained for 10 epochs using an SGD optimizer with a mini-batch of 128. We use a fixed learning rate and set its value to 0.01.

Evaluation metrics. Following [108], we adopt Precision@1, Precision@5 and Precision@10 to measure the ability of classifying visual relations. We will refer to the classification task as relation tagging in the experiments for consistency with current literature. For video relation detection we report mAP (mean Average Precision), Recall@50 and Recall@100.

3.6 RESULTS

Benefit of multi-modal features. We first evaluate the benefit of the use of multi-modal features on VidOR in Table 5. With only motion features, our method achieves a P@1 of 50.97 for relation tagging and an mAP of 6.14 for relation detection. With all features included, the performance is clearly improved with a P@1 of 68.86 for relation tagging and an mAP of 11.21 for relation detection. The results show that our encoding benefits from incorporating information from many modalities. In the following ablations, we use all features.

Influence of encoding size. Next, we evaluate the influence of the number of interaction primitives in the Social Fabric Encoding. Intuitively, the more primitives, the finer commonalities between interactions are modelled. In Table 6, we find that multiple primitive components indeed improves over a single component (which resembles conventional average pooling). When increasing the number of primitives, we further improve the performance. The Social Fabric Encoding performs best at K=64, where it provides a balance between coverage of the space and sharing amongst relations. We use this encoding size for further experiments.

Importance of two stages. Next, we show the importance of the interaction proposal stage and the predicate predication stage on VidOR in Table 7. The baseline (first row) splits the video into short snippets. Relationships are separately detected in each snippet and merged afterwards, akin to [98, 114, 136]. It average pools the features before predicate prediction. With the interaction proposal stage added (second row), we have spatio-temporal proposals covering long-range interactions. It provides the necessary context to recognize long duration interactions. Accordingly, both recall and precision improve. The Recall@50 is improved by 1.09 and P@1 is improved by 3.47 compared to the baseline. Upon adding the second stage (Third row), the P@1 increases by 4.67

Stage 1	Stage 2	Relation tagging			Relation detection		
		P@1	P@5	P@10	mAP	R@50	R@100
		60.72	46.40	36.62	9.61	8.73	10.81
✓		64.19	49.60	39.22	10.16	9.62	11.63
✓	✓	**68.86**	**55.16**	**43.40**	**11.21**	**9.99**	**11.94**

Table 7: **Importance of two stages** on VidOR. Incorporating Social Fabric into the two stages of our pipeline (third row) is preferred over baselines based on average pooling of features with video snippet proposals (first row) and using Social Fabric only for the proposals (second row).

Encoding	Relation tagging	Relation detection
	P@1	mAP
average pooling	62.73	10.05
transformer	63.86	10.07
NetVLAD	65.34	10.15
NetRVLAD	66.80	10.55
Social Fabric	**68.86**	**11.21**

Table 8: **Comparison with alternative encodings** on VidOR. Social Fabric performs well.

compared to when we only use interaction encoding in proposal generation. We conclude that both stages matter in combination with our encoding.

Comparison with alternative encodings. We compare to the following encodings on VidOR: average pooling, transformer encoding, NetVLAD [45], NetRVLAD [91]. Average pooling corresponds to our encoding with a single mixture component. Transformers were proposed in [124] for textual sequence-to-sequence tasks and recently adopted in video tasks [13, 43, 44]. Here, we investigate their potential for interaction detection. We feed the frame-level representations to the transformer encoder. The output representation is average pooled and then fed into the predicate classifier. NetVLAD was first introduced for place recognition and later adopted for video action classification in [45]. We train a classifier over the NetVLAD layer initialized by k-means on all features to initialize the cluster centroids (and keep it fixed). As our method, we use 64 cluster centroids. NetRVLAD [91] is a simplification of the original NetVLAD architecture that averages the actual descriptors instead of the residuals.

We report the P@1 and mAP on VidOR dataset in Table 8. All encodings take the same multi-modal representations as input. The transformer and average pooling baselines obtain similar performance. NetVLAD improves over average pooling and transformers, highlighting the effectiveness of codebook-based encodings. NetRVLAD further improves over NetVLAD, potentially because aggregating the actual feature instead of residuals may benefit the performance [34]. Our encoding uses a similar strategy with a dynamic learning scheme and outperforms all baselines, with an mAP of 11.21% compared to 10.55% for NetRVLAD as the best performing alternative.

	ImageNet-VidVRD						VidOR				
	Relation tagging			Relation detection			Relation tagging		Relation detection		
	P@1	P@5	P@10	mAP	R@50	R@100	P@1	P@5	mAP	R@50	R@100
Shang *et al.* [108]	43.00	28.90	20.80	8.58	5.54	6.37	-	-	-	-	-
Tsai *et al.* [121]	51.50	39.50	28.23	9.52	7.05	8.67	-	-	-	-	-
Qian *et al.* [98]	57.50	41.00	28.50	16.26	8.07	9.33	-	-	-	-	-
Sun *et al.* [115]	-	-	-	-	-	-	51.20	40.73	6.56	6.89	8.83
Su *et al.* [114]	57.50	41.40	29.45	19.03	9.53	10.38	50.72	41.56	6.59	6.35	8.05
Liu *et al.* [85]	60.00	43.10	32.24	18.38	11.21	13.69	48.92	36.78	6.85	8.21	9.90
Xie *et al.* [136]	-	-	-	-	-	-	67.43	-	9.93	9.12	-
Ours, features as Su *et al.* [114]	57.50	43.40	31.90	19.23	12.74	16.19	54.57	43.58	8.93	9.15	11.13
Ours, features as Liu *et al.* [85]	61.00	47.50	36.60	19.77	12.91	16.32	55.40	45.74	9.13	9.36	11.30
Ours, features as Xie *et al.* [136]	-	-	-	-	-	-	68.62	53.34	11.05	9.91	11.89
Ours, our features	**62.50**	**49.20**	**38.45**	**20.08**	**13.73**	**16.88**	**68.86**	**55.16**	**11.21**	**9.99**	**11.94**

Table 9: **Comparison with state-of-the-art** for relation tagging and detection on ImageNet-VidVRD and VidOR. We outperform the recent snippet relation detection methods of both Su *et al.* and Xie *et al.* for almost all metrics when using their features. We also outperform the proposal relation detection method of Liu *et al.* when using their features. When we rely on our full set of features results improve further and set a new state-of-the-art on both tasks for both benchmarks.

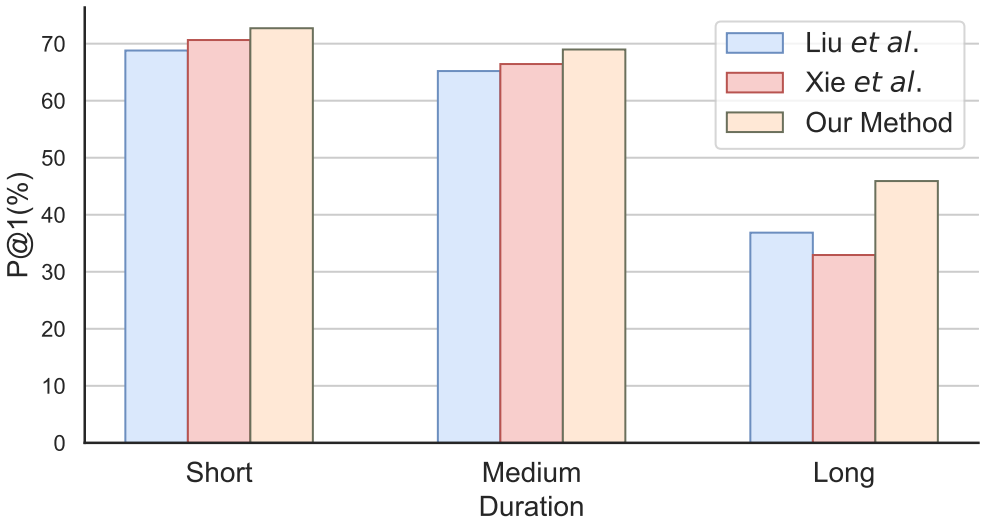


Figure 14: **Comparison along relation duration** on VidOR. We observe our method's performance improves over alternatives as the duration of the video relation increases.

Comparison with state-of-the-art. We compare with the state-of-the-art in video relation classification and detection in Table 9 for both ImageNet-VidVRD and VidOR. Liu *et al.* [85] report good results for relation classification and detection on both sets. When we compare with them using the same input features, *i.e.* visual, I3D and motion feature, we improve over their work on all metrics. Most notably, the mAP for relation detection improves from 18.38 to 19.77 on ImageNet-VidVRD and from 6.85 to 9.13 on VidOR. We also compare favorably against the recent snippet-based video relation detection of Su *et al.* [114] using their features. We are on par for the relation classification P@1 on ImageNet-VidVRD, but outperform them on all other metrics and datasets, demonstrating the benefit of detecting predicates for social tubelets from the start. Xie *et al.* [136] improved the state-of-the-art considerably by combining a motion feature, visual feature, language feature and location mask feature for each trajectory pair before predicting their relation. Our method profits from such a rich set of multi-modal features also. When we use the same features as Xie *et al.* our results get better as well, obtaining 68.62 P@1 and 11.05 mAP for relation classification and detection respectively. Our features adds I3D feature to the feature set used by Xie *et al.* [136]. Using our features we obtain state-of-the-art performance with 11.21 mAP and 68.86 P@1. We also consider the computational aspects of our method. We test using a GTX 1080 Ti GPU. With the same features as Liu *et al.* [85], the average time to process one ImageNet-VidVRD validation video is 58.2s for Liu *et al.* [85], and 48.3s for our method.

Comparison along relation duration. To verify the effectiveness of our approach on long-range relations. we break down the performance into three bins according to the duration of the relation instances: "short", "medium" and "long". We compare our method with Liu *et al.* [85] and Xie *et al.* [136] on the VidOR validation set. Results are shown in Figure 14. The three methods use the same features as Xie *et al.* [136] for a fair comparison. The results of Xie *et al.* [136] are provided by the authors. The results of Liu *et al.* [85] are obtained by running the provided code. As expected, Liu *et*

al. [85] surpasses Xie *et al.* [136] for long-duration relations as they are designed to be effective beyond short-snippets. Our method is beyond both Liu *et al.* and Xie *et al.* for all durations. Compared to Xie *et al.* [136] who do not consider long-range relations, our method's performance gain increases as the relation length increases. We conclude our approach is beneficial for encoding multi-modal features for relation detection especially at long-range. Besides, we have split the predicates in VidOR into two super categories: action-based and spatial-based relations, following [37]. We obtain a mAP of 7.33% for action-based relations and a mAP of 12.89% for spatial-based relations, while the state-of-the-art by Xie *et al.* [51] obtains a mAP of 6.25% for action-based relations and a mAP of 11.23% for spatial-based relations. We show some success and failure cases in Figure 15.

Video relation query-by-primitive-examples. In Figure 16 we show three search cases, where for each case three primitive examples are given as input. We use the VidOR validation set for the search. The results show that we can find relevant video relations in space and time across many videos, simply by providing a few primitive examples, further highlighting the importance of compositions for video relations.

3.7 CONCLUSION

We propose an approach to video relation classification and detection that operates on pairs of object tubelets from the start. By doing so we no longer have to scatter the video into snippets or individual object tubelets and gather them at the end. To represent all pairs of object tubelets appearing in a video, we propose Social Fabric: an encoding built on a composition of data-driven interaction primitives, akin to the classical codebook approach. We use the encoding in a two-stage network, that first suggests proposals that are likely interacting and then fine-tunes and predicts it most likely predicate label. Experiments demonstrate the benefit of early video relation modeling, our encoding, as well as the two-stage architecture, leading to new state-of-the-art on two video relation benchmarks. We also show how the encoding enables spatio-temporal video search by query-by-primitive-examples.

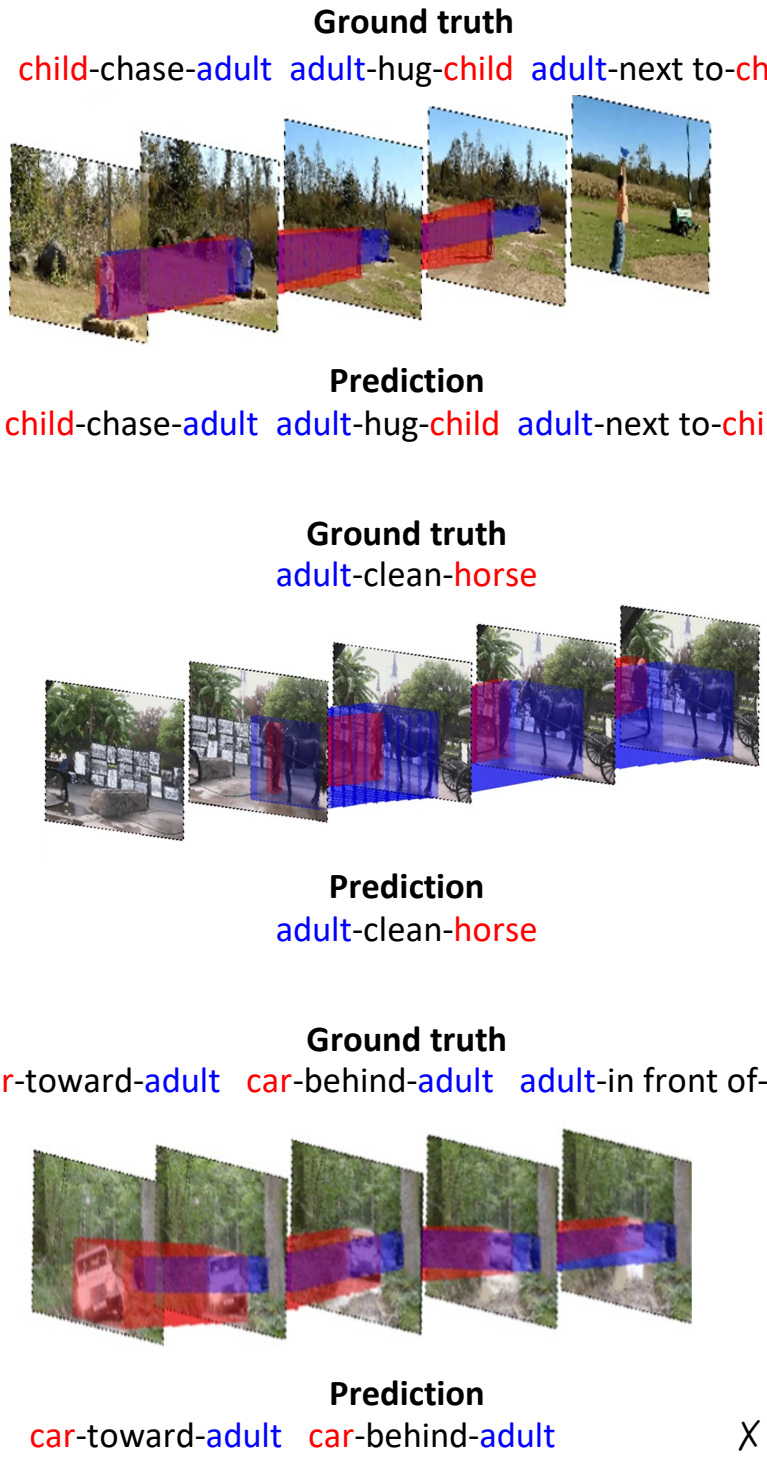


Figure 15: **Success and failure cases** on VidOR. For the left example, we detect all the ground truth relation instances and successfully predict the long-range relation `chase`. The middle case needs temporal context information to detect an adult cleaning a horse. Our method's detection proves its ability to detect long-range relations. In the right example, our approach detects `behind` and `toward` relations. But since the object detector wrongly recognizes `car` as `truck`, the final triplet predictions are wrong even though the relation predicates are correct. Incorrect object categories also lead to imprecise semantic features, which may contribute to the missing of a relation prediction. We provide more qualitative results and example videos with success and failure in the supplemental material.

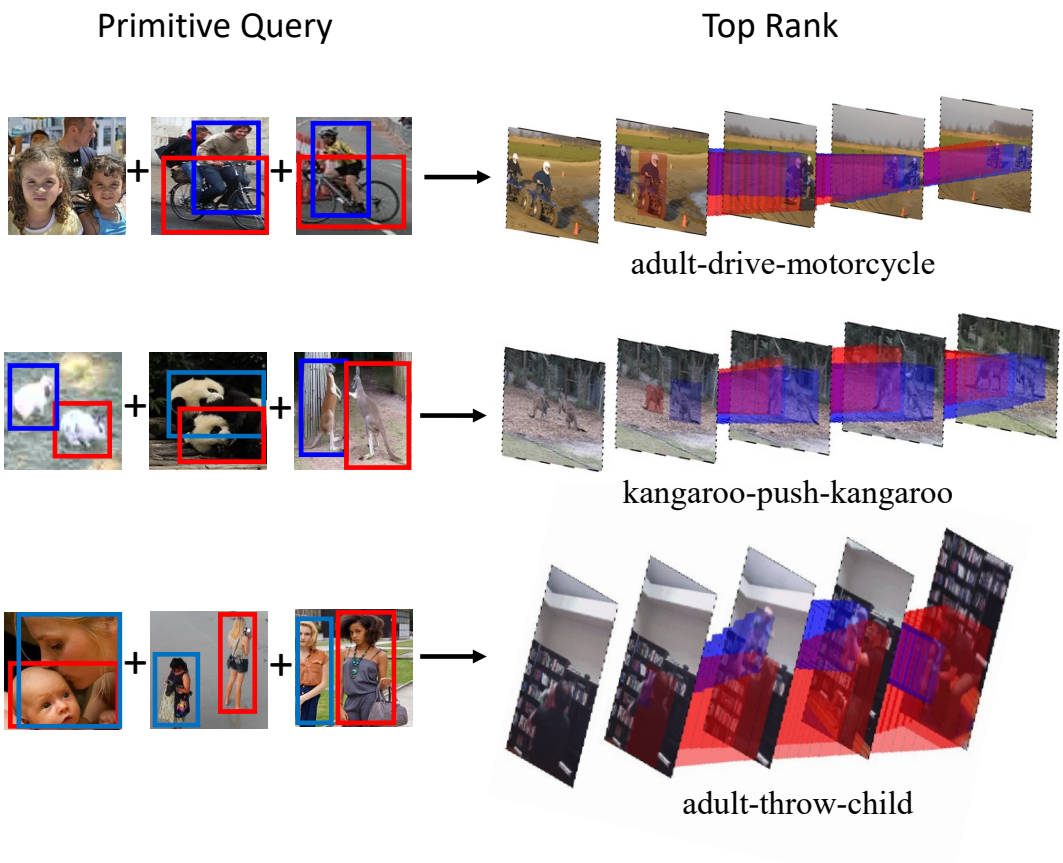


Figure 16: **Query-by-primitive-examples.** We use three examples of primitives as queries. Among the VidOR validation set, the relation whose primitive weights are closest to the three examples is selected. *e.g.*, in the third row, three examples represent primitives of "subject touches object", "subject and object moving away" and "subject and object are person". And the top-ranked relation we return is *adult, throw, child*.

4

DIAGNOSING ERRORS IN VIDEO RELATION DETECTORS

4.1 INTRODUCTION

This chapter performs an in-depth investigation into the video relation detection task. Video relation detection, introduced by Shang *et al.* [108], requires spatio-temporal localization of object and subject pairs in videos, along with a predicate label that describes their interaction. To tackle this challenging problem, Shang *et al.* [108] first proposed a three-stage approach: split a video into snippets, predict the predicate, and associate the snippets over time. Such a three-stage tactic has since become popular for video relation detection [32, 98, 114, 121, 136]. Among them, Tsai *et al.* [121], Qian *et al.* [98] and Xie *et al.* [136] focus on improving predicate prediction. Tsai *et al.* and Qian *et al.* construct graphs to pass messages between object nodes, while Xie *et al.* utilizes multi-modal features. Alternatively, both Di *et al.* [32] and Su *et al.* [114] shift their attention to a better association process.

Not all works follow a canonical three-stage approach. Cao *et al.* [14], for example, propose a 3D proposal network to learn relational features in an end-to-end manner. Sun *et al.* [115] and Liu *et al.* [85] rely on a sliding window to generate proposals and recognize predicates within proposals. Chen *et al.* [24] learn interaction primitives to generate interaction proposals [22] and recognize predicates. While video relation results keep progressing, there is still a lot of room for improvement. For example, Xie *et al.* [136], the winner of the Video Relation Detection task from the Video Relation Understanding Challenge 2020, combines a wide variety of multi-modal features for each subject-object tubelet pair to predict the relations with an improved detection performance. Nonetheless, their final mAP (mean Average Precision) is only 9.66% on the VidOR validation set [107]. In short, the task is far from solved. Moreover, it is unclear which factors are most critical for better results. We seek to fill this void.

We take inspiration from error diagnosis in the spatial domain for object detection [11, 55] and in the temporal domain for action detection [3, 96]. These works have previously performed a deep dive into the main sources of errors for their respective tasks, including false positive analysis, false negative analysis, and mAP sensitivity tests for object attributes or action characteristics. The analyses have helped to explain limitations in the field and to provide guidance for the next steps [1, 3, 8, 11, 37, 55–57, 149, 154]. In a similar spirit, we shine a light on the spatio-temporal domain for video relation detection, where the spatial challenges of object detection and the temporal challenges of action detection need to be simultaneously addressed. Code is available at `https://github.com/shanshuo/DiagnoseVRD`.

We provide an error diagnosis for video relation detection, which starts with an outline of current benchmarks, evaluation protocols, the algorithms under consideration, and a categorisation of different possible error types. Under this setup, we make the following analytical contributions:

- false positive analysis outlining which types of errors are most common, along with potential cures for each error type, evaluated on two state-of-the-art approaches;
- false negative analysis along with a categorization of the kind of relation characteristics that are most difficult to detect;
- analysis of the different video relation characteristics and their influence on the performance, including relation length, number of subject/object/predicate instances, and spatio-temporal subject and object size;
- oracle analysis to identify which aspects lead to the biggest improvements.

4.2 ERROR DIAGNOSIS SETUP

As a starting point of the error diagnosis, we first outline the core characteristics and biases of the current video relation detection datasets, the definitions of different error types, and the methods from the literature under investigation.

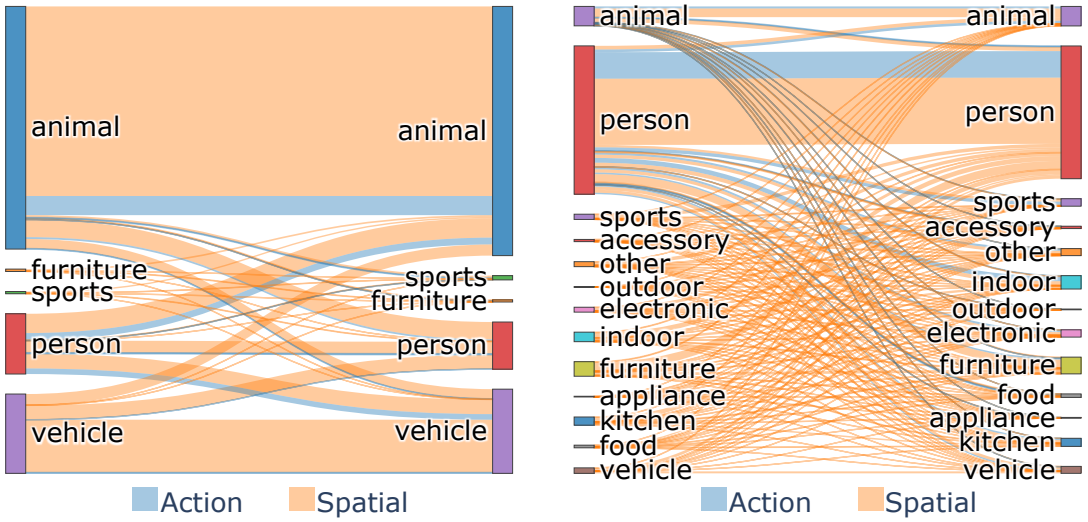
4.2.1 *Dataset characterization*

We perform our analysis on the two existing datasets in video relation detection, namely ImageNet-VidVRD [108] and VidOR [107].

ImageNet-VidVRD [108] consists of 1,000 videos, created from the ILSVRC2016-VID dataset [104]. There are 35 object categories and 132 predicate categories. The videos are densely annotated with relation triplets in the form of ⟨*subject-predicate-object*⟩ as well as the corresponding subjects and objects trajectories. Following [108, 121], we use 800 videos for training and the remaining 200 for testing. We analyze the method performance on the 200 test videos.

VidOR [107] contains 10,000 user-generated videos selected from YFCC-100M [118], for a total of about 84 hours. There are 80 object categories and 50 predicate categories. Besides providing annotated relation triplets, the dataset also provides the bounding boxes of objects. The dataset is split into a training set with 7,000 videos, a validation set with 835 videos, and a testing set with 2,165 videos. Since the ground truth of the test set is not available, we use the training set for training and the validation set for testing, following [85, 98, 114, 136]. We report the analysis of method performance on the VidOR validation set.

Prevalent relations. To gain insight into the large number of possible combinations of subjects, objects, and interactions in ImageNet-VidVRD and VidOR, we first categorize all into super categories and investigate patterns among the super categories. For VidOR, the object categories are based on MS-COCO [84] and we, therefore, use its 12 object super categories, along with an *other* category for exceptions. For the predicates, we employ the hierarchy in VidOR that makes a split into *action-based* and *spatial* predicates. In the supplementary materials, we show the prevalent objects and predicates of ImageNet-VidVRD and VidOR. Animals and persons are the dominant subjects and



(a) ImageNet-VidVRD. (b) VidOR.

Figure 17: **Subject, object, and predicate diagrams** on ImageNet-VidVRD and VidOR. On both datasets, knowledge about animals, persons, vehicles, and spatial relations will go a long way for video relation detection due to a large bias towards these overarching category types.

objects, while spatial predicates form the dominant interactions between them. This is not surprising, as spatial relations are common and omnipresent.

Predicate biases. For a given dataset, the number of relations consists of all combinations of subjects, objects, and predicates. Most combinations are however not likely to occur, resulting in a bias towards common and generic ⟨*subject-predicate-object*⟩ triplets. We find that subject and object labels are highly predictive of predicate labels. Figure 17 shows which subjects and objects are likely to be in interaction and indicates which type of predicate commonly occurs between super categories of subjects and objects. To quantify the bias towards predicate categories for subject-object pairs, we predict the predicate using a naïve Bayes classifier built upon training set statistics between subjects and objects. On ImageNet-VidVRD, the predicate accuracy on the validation set is 14.02% compared to 0.8% for random guessing. On VidOR, the accuracy is 36.11% compared to 2.0% for random guessing. Evidently, there is not only a strong bias towards common predicates but also from subjects and objects to predicates. Empirically, we will investigate whether current video relation detection approaches also mirror this bias.

4.2.2 *Evaluation protocol and error types*

In the literature, the mean Average Precision (mAP) is widely used for video relation detection evaluation [85, 98, 108, 109, 114, 115, 120, 136]. Different from conventional Average Precision evaluation for detection [36], the averaging per category is performed over videos, not categories. Let G be the set of ground truth instances for a video such that an instance $g^{(k)} = (\langle s, p, o \rangle^g, (T_s^g, T_o^g))$ consists of a relation triplet label $\langle s, p, o \rangle^g$ with subject and object bounding-box trajectories (T_s^g, T_o^g). Let P be the set of predictions such that a prediction $p^{(i)} = (p_s^{(i)}, \langle s, p, o \rangle^p, (T_s^g, T_o^g)$ consists of a relation triplet score

Error type	Definition
Classification error	Overlap between discovered and ground truth relation is above 0.5, the relation triplet labels are not identical.
Localization error	Overlap between discovered and ground truth relation is between 0.1 and 0.5, the relation triplets labels are identical.
Confusion error	Overlap between discovered and ground truth relation is between 0.1 and 0.5, the relation triplets are not identical.
Double detection	Overlap between discovered and ground truth relation is above 0.5, the relation triplet are identical, but the ground truth instance has already been detected.
Background error	Overlap between discovered and *any* ground truth relation is lower than 0.1.
Missed ground truth	An undetected ground truth instance not covered by other errors.

Table 10: **Categorization of six different types** covering all errors that a video relation detector can make. The error types are used for our in-depth false positive analysis.

$p_s^{(i)}$, a triplet label $\langle s, p, o\rangle^p$, and predicted subject and object trajectories. To match a predicted relation instance $(\langle s, p, o\rangle^p, (T_s^p, T_o^p))$ to a ground truth $(\langle s, p, o\rangle^g, (T_s^g, T_o^g))$, we require:

i their relation triplets to be exactly the same, i.e. $\langle s, p, o\rangle^p = \langle s, p, o\rangle^g$;
ii their bounding-box trajectories overlap s.t. $\text{vIoU}(T_s^p, T_s^g) \geq 0.5$ and $\text{vIoU}(T_o^p, T_o^g) \geq 0.5$, where vIoU refers to the voluminal Intersection over Union;
iii the minimum overlap of the subject trajectory pair and the object trajectory pair $\text{ov}_{pg} = \min(\text{vIoU}(T_s^p, T_s^g), \text{vIoU}(T_o^p, T_o^g))$ is the maximum among those paired with the other unmatched ground truths G, *i.e.* $\text{ov}_{pg} \geq \text{ov}_{pg'}(g' \in G)$.

While calculating the score, we only consider the top 200 predictions for each video. After we get AP for each video, we finally calculate the mean AP (mAP) over all testing/validation videos. The above criteria make it hard for the ground truth to match the prediction. In this work, we are not only interested in the matches, but also in analyzing the mismatches. In Table 10, we have outlined six possible error types, five False Positives, and one False Negative. We visualize and show qualitative examples of true positives as well as different error types in Figure 18. We will use these error types to investigate common pitfalls in current video relation detection approaches.

4.2.3 *Algorithms under investigation*

We exemplify the use of our diagnostic tool by studying two state-of-the-art approaches which have conducted experiments on ImageNet-VidVRD and VidOR. Both methods tackled the problem in a three-stage manner, similar to [108]. However, there are design differences in each stage which are relevant to highlight.

Liu *et al.* [85] avoid the need to split videos into snippets. In the first stage, they generate object tubelets for the whole video. The second stage refines the tubelet-features and finds relevant object pairs using a graph convolutional network. The third stage focuses on predicting the predicates between related pairs. In this manner, interactions can be detected without a need for snippet splitting.

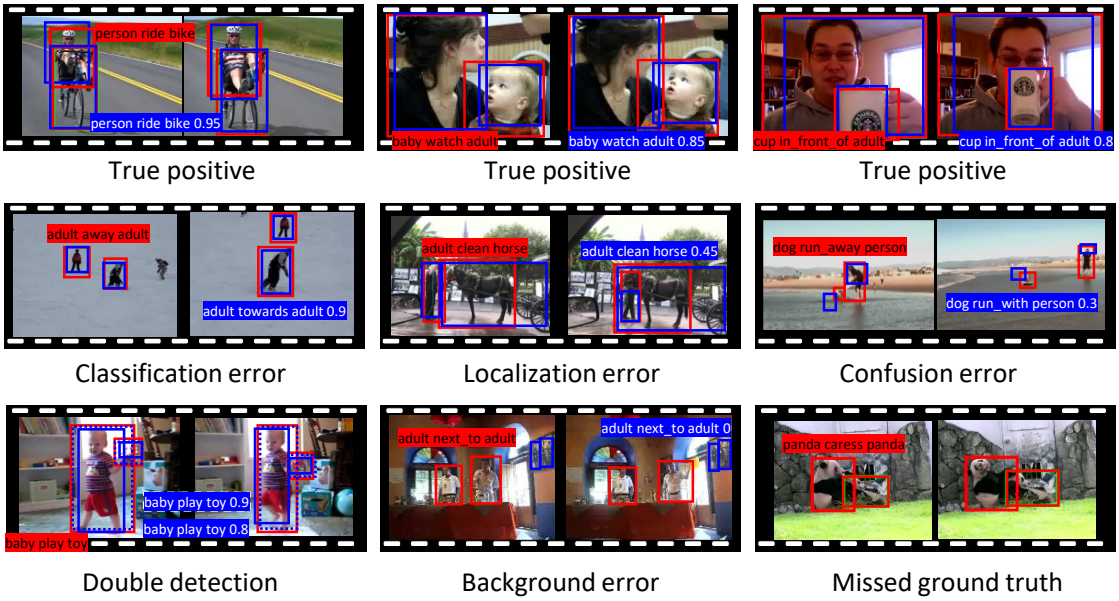


Figure 18: **Video relation detection examples** of true positives and the six error types from Table 10. Red boxes indicate ground truth and blue boxes specify predictions. The number in the blue box is the vIoU between the detection and the ground truth. The dashed boxes in double detection represent the best-mapped prediction to this ground truth. To match a prediction to ground truth is difficult and many factors could influence the final performance.

Su *et al.* [114] is based on the three-stage architecture proposed in Shang *et al.* [108]. A video is first to split into short snippets and subject/object tubelets are generated per snippet. Then, short-term relations are predicted for each tubelet. In the second stage, spatio-temporal features of each pair of object tubelets are extracted and used to predict short-term relation candidates. In the third stage, they maintain multiple relation hypotheses during the association process to accommodate for inaccurate or missing proposals in the earlier steps.

4.3 FINDINGS

In this section, we demonstrate the generality and usefulness of our analysis toolbox by exploring what restricts the performance of video relation detection approaches. We first conduct a false positive analysis, composed of the first five error types defined in Table 10 (classification, localization, confusion, double detection, background). Then, we analyze the false negatives, *i.e.* missed ground truth (Miss), along with different relation characteristics that correlate with the false negatives. Finally, we contribute the mAP gain of each error type.

4.3.1 *False positive analysis*

The first experiment investigates which error types are prevalent in current approaches. To answer this question, we break down the false positives and present the distribution

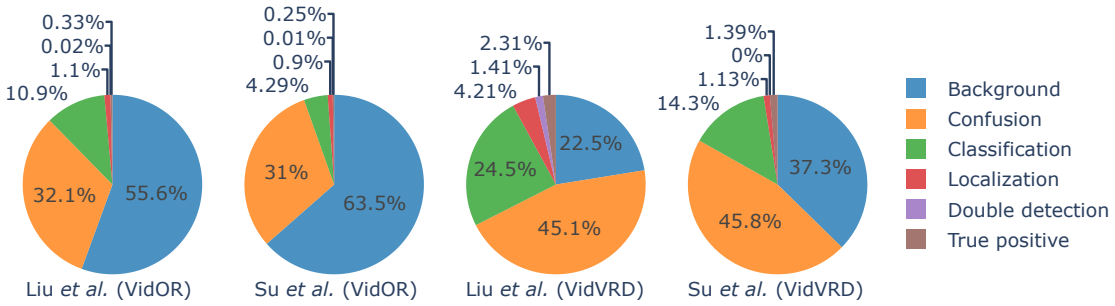
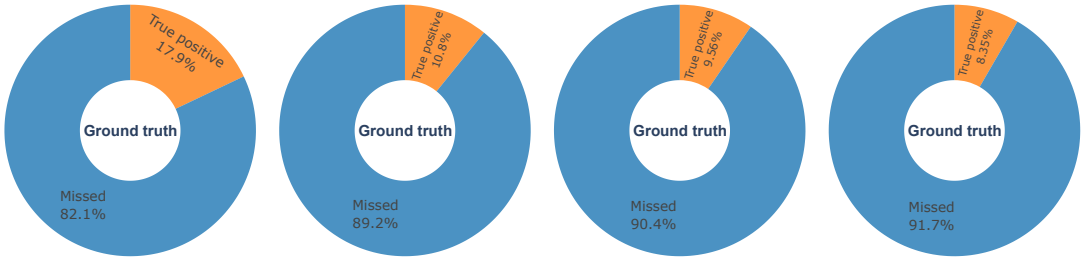


Figure 19: **The false positive error breakdown** in Liu *et al.* [85] and Su *et al.* [114] on the VidOR and ImageNet-VidVRD datasets. The classification error, which is also one cause of confusion error, as well as background error, should be solved first in future research.

of errors for Liu *et al.* [85] and Su *et al.* [114] on ImageNet-VidVRD and VidOR in Figure 19. To our surprise, we find that in all four cases, the localization error takes only a small part of all false positives in the spatio-temporal detection task. Since in diagnostic papers on well-established detection tasks such as object detection [11, 55] and temporal action detection [3], localization error is important and takes a much larger ratio. Due to the large amount of possible triplet combinations, it is more common to have both low overlapping volumes as well as wrong triplet labels, categorized as confusion errors. Next, we see that there is almost no double detection error. When predicting predicates, Liu *et al.* and Su *et al.* keep the top 20 prediction results for each subject-object pair. Thus, the diversity in the predicted detection results makes it difficult to map the multiple detections to the same ground truth.

Comparison across methods. From Figure 19 we can observe that the background error ratio is much lower in Liu *et al.* compared to Su *et al.*. Liu *et al.* generate less detections where no interesting relations are involved. We attribute this to their proposal generation and filtering stages. Su *et al.*'s split and merge pipeline might be unable to remove bad proposals efficiently. Another observation is that Liu *et al.*'s classification error is much higher than the one of Su *et al.* on ImageNet-VidVRD. Su *et al.*'s multiple hypothesis association enables to connect neighbour segments with low predicate prediction scores. When ranking detection results, the scoring reflects the reliability of forming the corresponding hypothesis video relation, enabling a more robust ranking for those with a lower predicate prediction score. This is beneficial especially for ImageNet-VidVRD with more predicate categories but less training data, resulting in undistinguished classification scores for predicates. Su *et al.* have fewer true positives than Liu *et al.*, but higher mAP. This also shows that Su *et al.*'s scoring algorithm outperforms Liu *et al.*. In VidOR, with more training data and fewer predicate categories, Su *et al.* have a lower classification error ratio than Liu *et al.*, but the gap is not as large as on ImageNet-VidVRD. We conclude that Liu *et al.* and Su *et al.* have their own advantages for dealing with different error types. Both have in common that the background error and classification error should have higher priority than the other error types to gain the most in performance.



(a) Liu *et al.* (VidVRD) (b) Su *et al.* (VidVRD) (c) Liu *et al.* (VidOR) (d) Su *et al.* (VidOR)

Figure 20: **The missed ground truth error** (false positive) ratio on ground truth in Liu *et al.* [85] and Su *et al.* [114] on ImageNet-VidVRD and VidOR datasets. Both have many ground truths undetected.

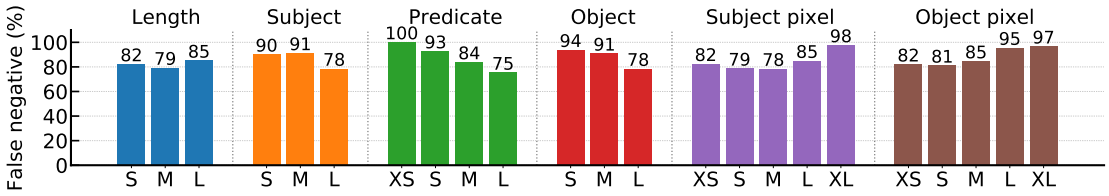
4.3.2 *False negative analysis*

So far, we have only considered the types of false positive errors introduced by the detection algorithms. However, false negative errors (missed ground truth) also influence the mAP.

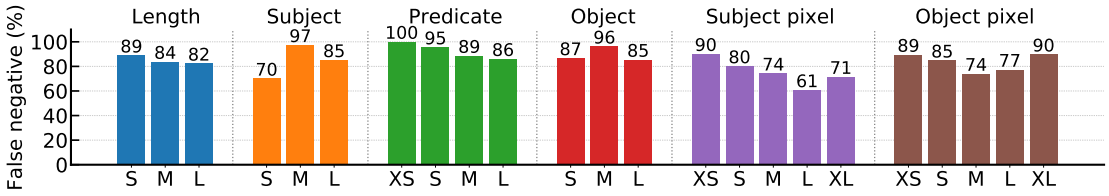
In Figure 20 we present the missed ground truth ratios for Liu *et al.* and Su *et al.* on ImageNet-VidVRD and VidOR. For both ImageNet-VidVRD and VidOR, roughly 90% of the ground truth relation instances remain undetected. VidOR has a higher missed ground truth ratio, highlighting the more complex nature of the dataset. On ImageNet-VidVRD, Liu *et al.* detect more instances than Su *et al.* but attribute them with lower scores, leading to a lower mAP value. This tells us that proposal-based methods can cover more relations, while Su *et al.*'s scoring method helps to better rank detected predictions. It is insightful to study what makes these missed ground truth instances difficult to detect. Towards this end, we group the instances according to six relation characteristics defined below:

- **Length:** we measure relation length by the duration in seconds and create three different length groups: Short (S: (0, 10]), Medium (M: (10, 20]), and Long (L: ¿ 20). Overall, most of the instances are short, both in ImageNet-VidVRD (94.11%) and VidOR (80.06%). The number of medium and long relations is roughly similar.
- **Number of predicate instances**: we count the total number of predicate instances over all videos and create four categories: XS: (0, 10]; S: (10, 100]; M: (100, 1000]; L: (1000, 10000]; XL: (10000, 100000]; XXL: ¿100000.
- **Number of subject instances**: idem but for subjects.
- **Number of object instances**: idem but for objects.
- **Subject pixel scale**: we take the average of the bounding boxes for the subject trajectories and group the mean bounding box. We define subjects with pixel areas between 0 and 162 as extra small (XS), 162 to 322 as small (S), 322 to 962 as medium (M), 962 to 2882 as (L), and 2882 and above as extra large (XL).
- **Object pixel scale**: idem but for objects.

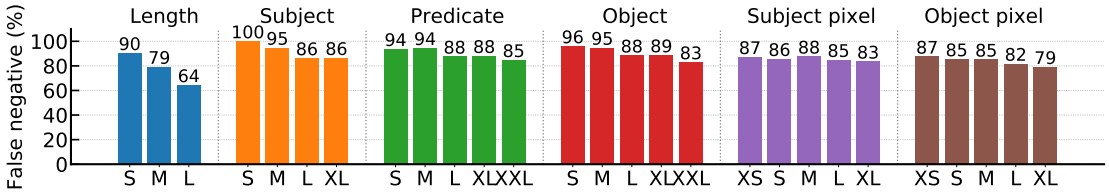
Figure 21 shows the overview of the effect for all relation characteristics for both Liu *et al.* and Su *et al.* on ImageNet-VidVRD and VidOR. We first observe a long-tail issue for the predicates. On ImageNet-VidVRD, both methods completely fail on relation



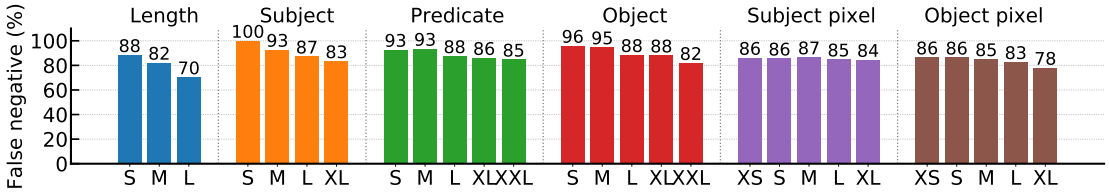
(a) Liu *et al.* (ImageNet-VidVRD).



(b) Su *et al.* (ImageNet-VidVRD).



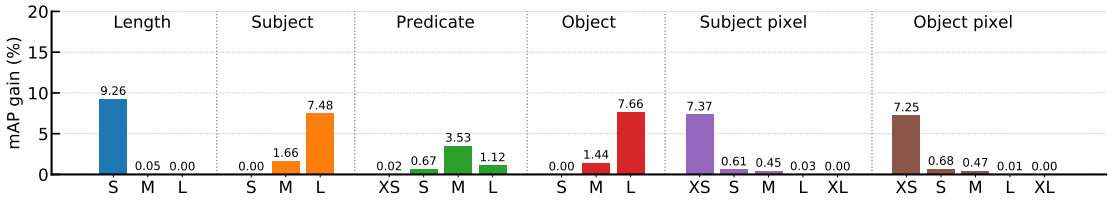
(c) Liu *et al.* (VidOR).



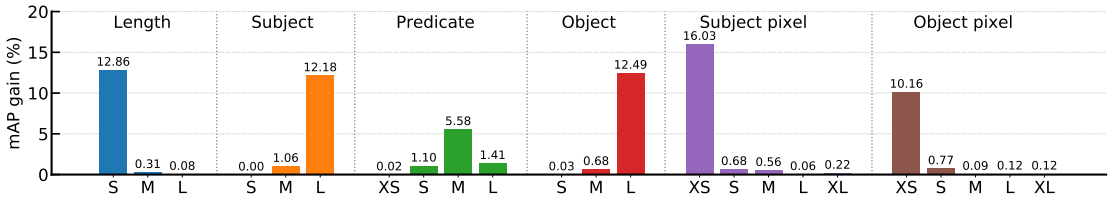
(d) Su *et al.* (VidOR).

Figure 21: **Relation characteristics** of Liu *et al.* and Su *et al.* on ImageNet-VidVRD and VidOR. Relations with fewer subject/predicate/object instances and smaller subject/object pixel areas are more difficult to detect.

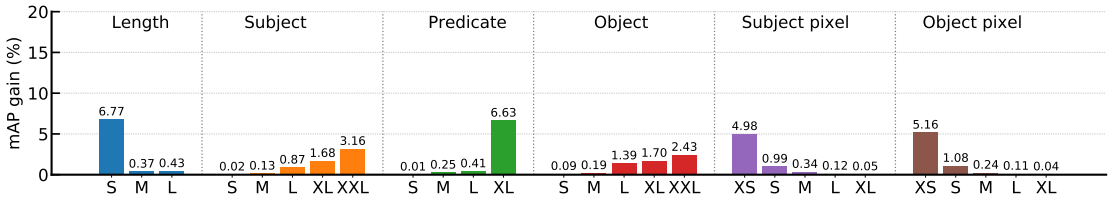
instances for which the predicate category has fewer than 10 samples. This means that datasets with more training samples are essential to this task, or methods should better exploit the few available samples. Another observation is that Su *et al.* have fewer false negatives on long-range relations on ImageNet-VidVRD, even though Liu *et al.* focus on long-range representations in their approach. This may be due to the construction of the ImageNet-VidVRD dataset, which was built through asking annotators to label segment-level visual relation instances in decomposed videos. This annotation procedure results in an abundance of relations that can be recognized without the need for long-range information. VidOR is annotated differently. Given a pair of object tubelets, the annotators are asked to find and temporally localize relations, resulting in more long-lasting relations. The patterns regarding the number of subject and object instances are intuitive in VidOR; the more instances to train on the better. Moreover, subjects and objects with larger size are easier to detect than smaller size. This pattern does, however, not hold for ImageNet-VidVRD, which could be due to the overall dataset size. Since



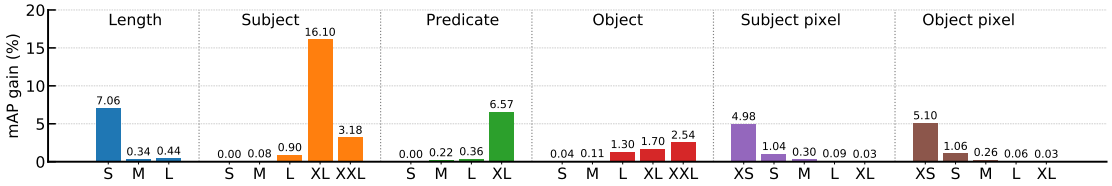
(a) Liu *et al.* (ImageNet-VidVRD).



(b) Su *et al.* (ImageNet-VidVRD).



(c) Liu *et al.* (VidOR).



(d) Su *et al.* (VidOR).

Figure 22: **The mAP gain on relation characteristics** of Liu *et al.* and Su *et al.* on ImageNet-VidVRD and VidOR. Focusing on detecting relation instances with a short temporal timespan, a large number of instances, and small pixel areas for the subject and object will improve the mAP by the largest margin.

the numbers of 'XL' subpxl and 'XL' objpxl in ImageNet-VidVRD are much lower than in VidOR.

To deepen the analysis of each characteristic's effect, we calculate the mAP gain after dropping the missed ground truths under this characteristic. From Figure 22, we observe that not all characteristics contribute equally to gains in mAP. It reveals that to improve the final metric the most, methods should focus on detecting relation instances with a short temporal timespan, a large number of instances, and small pixel areas for the subject and object.

4.3.3 *mAP sensitivity*

Where we have so far looked into which errors are most prevalent, we also want to examine to what extent each error type in Table 10 is holding back progress. We do so by quantifying the impact on the mAP for each error type by means of an oracle fix. We show how the mAP changes when each error type would be fixed. Rather than only

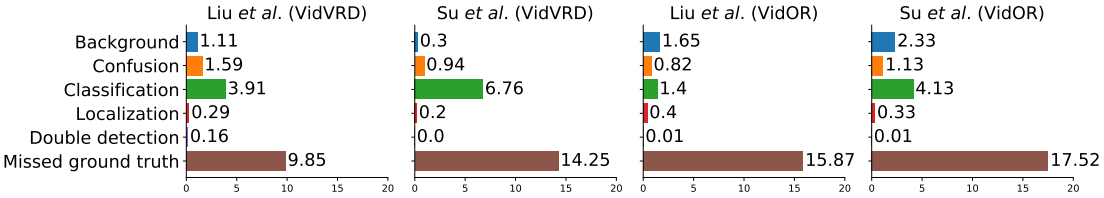


Figure 23: **The mAP gain** in Liu *et al.* [85] and Su *et al.* [114] on ImageNet-VidVRD and VidOR. Fixing missed ground truth error will maximize the performance improvement.

removing the predictions causing this error [3], we define the following cures for each of the main error types:

- **Classification cure**: Correct the class of the detection (thereby making it a true positive). If this results in a duplicate detection, remove the lower scoring detection.
- **Localization cure**: Set the localization of the detection equal to the ground truth localization (thereby making it a true positive). If this results in a duplicate detection, remove the lower scoring detection.
- **Confusion cure**: Since we cannot be sure of which ground truth the detector was attempting to match to, we remove the false positive detection.
- **Double detection cure**: Remove the duplicate detection with lower score.
- **Background cure**: Remove the background detection.
- **Missed ground truth cure**: Reduce the number of ground truth instances in the mAP calculation by the number of missed ground truth.

Figure 23 shows the error types impact on the mAP. Note that the sum of each error type's mAP gain is not 100%. The reason is due to the property of mAP. If we fix the error types progressively, the final mAP will be 100%. But the later fixed error types will gain more weights than earlier fixed error types. For a meaningful comparison, we fix them separately. In Figure 23, fixing missed ground truth errors will improve the mAP by a large margin, Su *et al.* with 14.25% on ImageNet-VidVRD and 17.52% on VidOR. However, in practice, we cannot simply drop these missed ground truths. The solution is to include more ground truths in the selected top 200 detections of a video. And many detections that could be matched to missed ground truths are not selected due to their low scores. We believe one direction is improving the performance of the predicate prediction module, to give the background proposals low scores and proposals of correct predicate categories high scores. This will also fix the classification errors and background errors to boost the final mAP further.

4.4 CONCLUSION

This work performs a series of analyses to understand the challenging problem of video relation detection better. Using two canonical approaches, we first perform false positive analyses and define the different types of errors. Two error types are prevalent across approaches and datasets: confusion with non-matching ground truth relations and detecting relations that are part of the background. We then perform false negative analyses, which show that most ground truth instances are missed entirely. Focusing on detecting relation instances with a short temporal length, a large number of instances,

and small pixel areas for the subject and object will improve the mAP the most. Lastly, to create a future outlook, we investigate several cures for common errors and find that the ability to discard background relations provides the shortest path to improve video relation detection performance. Our toolbox is generic and can be employed on top of any video relation detection approach. We make the toolbox and evaluation scripts publicly available to help researchers dissect their video relation detection approaches. Currently, our tool only considers the single variant's effect on the final metric, we will investigate a multivariate statistical analysis in the future.

4.5 APPENDIX

Super Category	Category	Classes	Instances
	Subjects & Objects		
Animal	turtle, antelope, lion, cattle, red_panda, horse, monkey, fox, elephant, bird, sheep, giant_panda, squirrel, bear, tiger, snake, rabbit, whale, dog, domestic_cat, lizard, hamster, zebraf	23	39097 (63.57%)
Furniture	sofa	1	356 (0.58%)
Person	person	1	8536 (13.88%)
Sports	ball, frisbee, skateboard	3	519 (0.84%)
Vehicle	bicycle, motorcycle, airplane, watercraft, bus, train, car	7	12996 (21.13%)
	Predicates		
Action	bite, chase, drive	14	2956 (9.61%)
Spatial	walk_above, stand_behind, next_to, fly_toward, sit_right, jump_with, walk_behind, creep_above, stand_front, run_front, run_left, jump_next_to, right, creep_right, walk_left, fly_left, swim_beneath, swim_behind, creep_left, creep_away, creep_next_to, lie_left, creep_behind, walk_right, stand_inside, stand_left, jump_above, move_past, run_past, walk_toward, left, creep_toward, jump_toward, walk_next_to, sit_inside, stand_right, run_next_to, lie_behind, fly_right, lie_beneath, sit_left, past, run_away, stop_above, move_with, move_right, lie_above, stop_with, jump_left, stop_right, front, jump_beneath, walk_past, sit_behind, move_above, lie_next_to, walk_beneath, walk_with, move_beneath, run_above, run_with, toward, run_beneath, stop_behind, jump_behind, move_left, walk_front, move_toward, move_behind, above, move_away, swim_left, stand_with, stop_left, stand_beneath, beneath, stand_next_to, swim_front, creep_beneath, lie_front, move_front, fly_above, sit_beneath, sit_front, jump_away, stop_beneath, sit_above, run_behind, fly_front, creep_front, faster, stop_next_to, away, lie_with, run_toward, lie_right, lie_inside, stop_front, run_right, taller, stand_above, swim_with, jump_past, fly_away, creep_past, walk_away, behind, move_next_to, jump_front, swim_next_to, jump_right, swim_right, sit_next_to, fly_with, larger, fly_behind, fly_next_to, fly_past	118	27796 (90.39%)

Table 11: **Subject/object and relation categories in Imagenet-VidVRD dataset**, organized by super categories. Note the bias towards animals and spatial relations.

Super Category	Category	Classes	Instances
	Subjects & Objects		
Accessory	handbag, backpack, suitcase	3	5948 (1.00%)
Animal	leopard, snake, chicken, hamster/rat, stingray, antelope, turtle, panda, tiger, sheep/goat, crocodile, pig, fish, cat, dog, lion, bird, elephant, duck, camel, kangaroo, crab, cattle/cow, penguin, horse, squirrel, bear, rabbit	28	51009 (8.58%)
Appliance	refrigerator, sink, oven, microwave, electric_fan	5	2034 (0.34%)
Electronic	camera, cellphone, screen/monitor, laptop	4	16005 (2.69%)
Food	fruits, bread, cake, vegetables, dish	5	8094 (1.36%)
Furniture	table, toilet, stool, chair, sofa	5	41089 (6.91%)
Indoor	toy	1	30034 (5.05%)
Kitchen	bottle, cup	2	21330 (3.59%)
Other	piano, baby_walker, baby_seat, faucet, guitar	5	15789 (2.65%)
Outdoor	bench, stop_sign, traffic_light	3	1744 (0.29%)
Person	adult, baby, child	3	368549 (61.97%)
Sports	bat, snowboard, surfboard, skateboard, racket, ball/sports_ball, ski, frisbee	8	16697 (2.81%)
Vehicle	scooter, watercraft, bus/truck, bicycle, aircraft, car, motorcycle, train	8	16382 (2.75%)
	Predicates		
Action	kiss, bite, push, point_to, wave_hand_to, drive, carry, open, watch, throw, clean, feed, wave, shake_hand_with, play(instrument), get_off, hug, touch, hold, pat, press, chase, close, release, grab, lift, smell, hold_hand_of, knock, lick, cut, kick, pull, get_on, lean_on, hit, speak_to, ride, shout_at, squeeze, caress, use	42	69066 (23.23%)
Spatial	above, next_to, beneath, in_front_of, away, behind, inside, towards	8	228286 (76.77%)

Table 12: **Subject/object and relation categories in VidOR**, organized by their super categories. Note the bias towards persons and spatial relations.

5

MULTI-LABEL META WEIGHTING FOR LONG-TAILED DYNAMIC SCENE GRAPH GENERATION

5.1 INTRODUCTION

Scene graph generation in videos focuses on detecting and recognizing relationships between pairs of subjects and objects. The corresponding dynamic scene graph is a directed graph whose nodes are objects with their relationships as edges in a video. Extracting such graphs from videos is a highly challenging research problem [67] with broad applicability in multimedia and computer vision. Effectively capturing such structural-semantic information boosts downstream tasks such as captioning [139], video retrieval [112], visual question answering [4] and many other visual-language tasks.

Current methods place a heavy emphasis on recognizing subject-to-object relationship categories. A leading approach to date is to extract multi-modal features for relation instances. Then the multi-modal features are either pooled [98, 114, 136] or learned a feature representation [24] to be fed into the predicate classifier network. Despite a strong focus on relation recognition, existing methods ignore the extremely long-tailed distribution of predicate classes. Figure 24 shows the recall per predicate class from STTran [28] with its corresponding occurrences on the Action Genome dataset. Such a trend is even worse on the VidOR dataset. Figure 25 shows the occurrence distribution vs. Recall@50 from Social Fabric [24] for the video relation detection task on the VidOR dataset, where a few head predicates dominate all other classes.

This phenomenon is currently not actively investigated since the evaluation metrics do not penalize lower scores for predicates in the long tail. In light of these observations, this chapter argues for the importance of methods for scene graph generation in videos that work for common and rare predicates.

We introduce a meta-learning framework to address the long-tailed dynamic scene graph generation problem. We take inspiration from the concept of meta weighting [110] and propose a Multi-Label Meta Weight Network (ML-MWN) to learn meta weights across both examples and classes explicitly. These meta weights are, in turn, used to steer the downstream loss to optimize the parameters of the predicate classifier. We adopt a meta-learning framework to optimize the ML-MWN parameters, where we calculate each instance's per-class loss in a training batch and obtain a loss matrix. The loss matrix is fed into our ML-MWN and outputs a weight matrix, where each row is the weight vector for an instance's loss vector. We sample a meta-validation batch and use unbiased meta-loss to guide the training of ML-MWN. We adopt the inverse frequency binary cross-entropy loss as the meta-loss. Finally, we plug our framework on top of existing methods to steer the predicate classification.

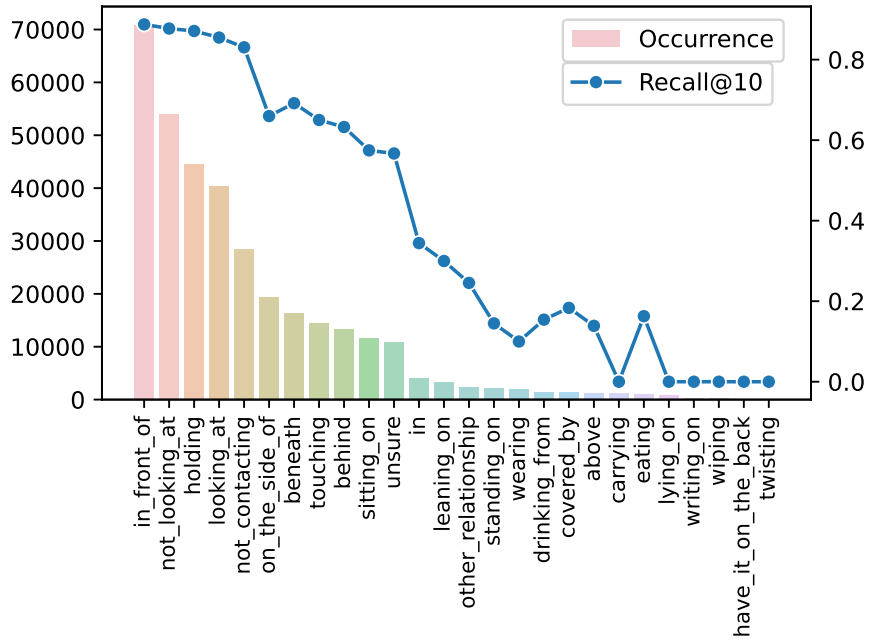


Figure 24: Long-tailed predicate occurrences vs. class-wise recall from STTran [28] on the Action Genome dataset [66]. The class-wise performance drops drastically, highlighting the importance of long-tailed dynamic scene graph generation.

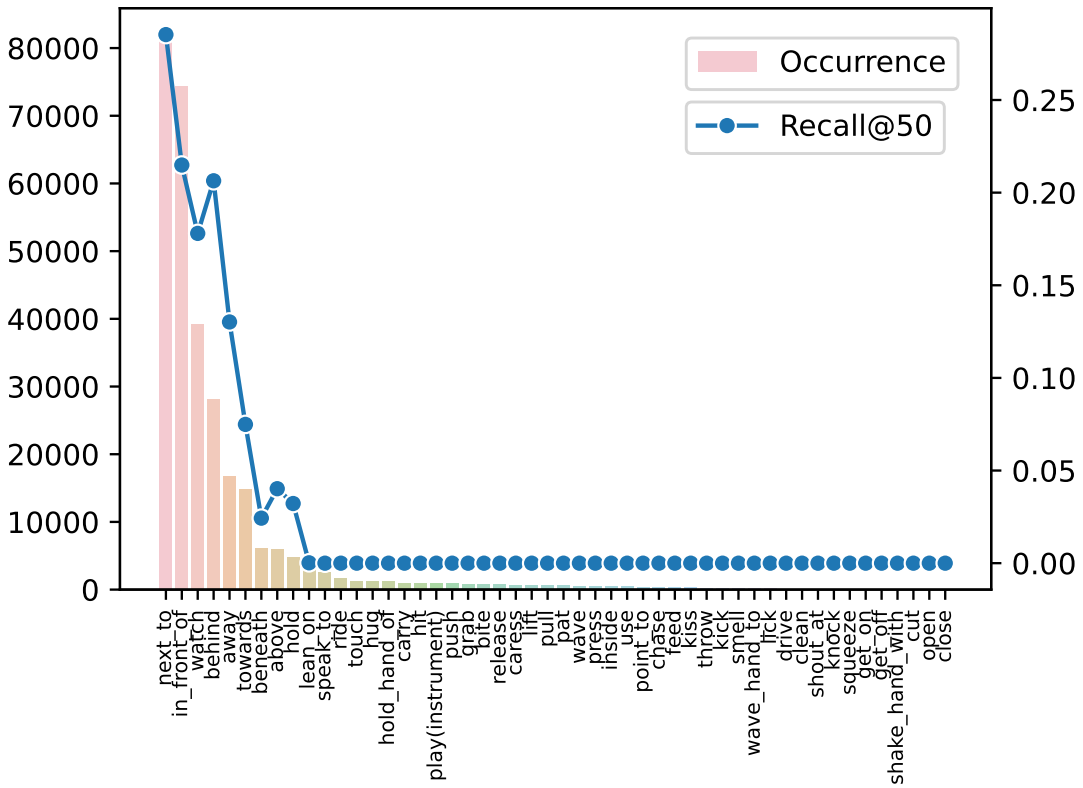


Figure 25: On the VidOR dataset [107], the long-tailed issue is even worse. We can observe that Social Fabric [24] ignores most predicates with limited samples.

To evaluate our meta-learning framework, we take two recent state-of-the-art methods [24, 28], one for the scene graph generation task on the Action Genome dataset and one for video relation detection on the VidOR dataset. We show empirically that our approach boosts the predicate predictions of the recent methods across various evaluation metrics. We furthermore show that our framework boosts the performance of long-tailed predicates without hampering the performance of more common classes. Our approach is generic and works on top of any scene graph generation method, allowing for broad applicability. Code is available at `https://github.com/shanshuo/ML-MWN`.

In summary, our contributions are three-fold:

1. We investigate the long-tail issue in dynamic scene graph generation and analyze the limitations of existing methods.

2. We introduce a multi-label meta-learning framework to deal with the biased predicate class distribution.

3. We propose a Multi-Label Meta Weight Network (ML-MWN) to learn a weighting function explicitly, which achieves a generalization ability performance on two benchmarks when plugged into two existing approaches,

5.2 RELATED WORKS

DYNAMIC SCENE GRAPH GENERATION. Scene graph generation was first pioneered in [67] for image retrieval, after which the task quickly gained further traction, see *e.g.* [86, 138, 141, 147]. Recently, a number of papers have identified the long-tailed distribution in image scene graphs and focus on generating unbiased scene graphs [31, 33, 74–76, 140]. We seek to bring the same problem to light in the video domain. Ji *et al.* [66] firstly extended scene graph generation to videos and introduced the Action Genome dataset. A wide range of works have since proposed solutions to the problem [14, 24, 42, 73, 85, 114–116, 136, 153]. Recently, Li *et al.* [76] proposed an anticipatory pre-training paradigm based on Transformer to model the temporal correlation of visual relationships. In a similar spirit, the VidOR dataset collected by Shang *et al.* [107] is another popular benchmark. Leading approaches are to generate proposals [22] for individual objects on short video snippets, encode the proposals, predict a relation and associate the relations over the entire video, *e.g.* [98, 114, 136]. Liu *et al.* [85] generate the proposals using the sliding window way. More recently, Gao *et al.* [42] proposed a classification-then-grounding framework, which can avoid that the quality of proposals highly influences the performance. Chen *et al.* [23] performs a series of analyses to video relation detection. In this paper, we use STTran [28] and Social Fabric [24] to extract the relation feature and insert our multi-label meta-weight network on top. Cong *et al.* [28] propose a spatial-temporal Transformer to capture the spatial context and temporal dependencies for a dynamic scene graph. Moreover, Chen *et al.* [24] proposes an encoding that represents a pair of object tubelets as a composition of interaction primitives. Both approaches provide competitive results and form a fruitful testbed for our meta learning framework.

MULTI-LABEL LONG-TAILED CLASSIFICATION. Multi-label long-tailed recognition is a challenging problem that deals with sampling differences and biased label

co-occurrences [150]. A few works have studied this topic, with most solutions based on new loss formulations. Specifically, Wu *et al.* [135] propose a distribution-based loss for multi-label long-tailed image recognition. More recently, Tian *et al.* [119] proposed a hard-class mining loss for semantic segmentation task by weighting the loss for each class dynamically based on instantaneous recall performance. Inspired by these loss-based works, we utilize inverse frequency cross-entropy loss during our meta learning process.

META LEARNING FOR SAMPLE WEIGHTING. Ren *et al.* [100] pioneered to adopt a meta learning framework to re-weight samples for imbalanced datasets. Based on [100], Shu *et al.* [110] utilize an MLP to explicitly learn the weighting function through an auxiliary MLP. Recently, Bohdal *et al.* [10] present EvoGrad to compute gradients more efficiently by preventing computing second-order derivatives in [110]. However, these methods are targeted for multi-class single-label classification. So we present the multi-label meta weight net for predicate classification, with an MLP that output weight for each class loss.

5.3 MULTI-LABEL META WEIGHT NETWORK

Dynamic scene graph generation [28] takes a video as the input and generates directed graphs whose objects of interest are represented as nodes, and their relationships are represented as edges. Each relationship edge, along with its connected two object nodes, form a ⟨subject, predicate, object⟩ semantic triplet. These directed graphs are structural representations of the video's semantic information. Highly related to dynamic scene graph generation, video relation detection [108] also outputs ⟨subject, predicate⟩ object triplets, aiming to classify and detect the relationship between object tubelets occurring within a video. Due to the high similarity between the two tasks, we consider them both in the experiments. For brevity, in this paper, we use the term dynamic scene graph generation to denote both tasks.

Action Genome [66] and VidOR [107] are two popular benchmark datasets for dynamic scene graph generation. However, both suffer from a long-tailed distribution on predicate occurrences, as shown in Figure 24. However, the evaluation metrics forgo the class-wise difference and count for all classes during the inference. As a result, the trained predicate classifier has a strong bias toward head classes such as in_front_of and next_to. While these predicate classes are often spatial-oriented and object-agnostic. The tail classes, *e.g.* carrying, twisting, and driving, are of more interest to us. Besides the long-tailed distribution, the predicate classification suffers an additional issue. Since multiple relationships could occur between a pair of subject and object simultaneously, predicate classification is a multi-label classification problem. The co-occurrence of labels leads to the situation that head-class predicate labels frequently appear together with tail-class predicate labels, which further emphasizes the imbalance problem.

This paper proposes a meta-learning framework that focuses on the long-tailed multi-label predicate classification task. We propose a Multi-Label Meta Weight Net (ML-MWN) to learn a weight vector for each training instance's multi-label loss. The gradient of the sum of weighted loss is then calculated to optimize the classifier network's parameters in backward propagation. Our model-agnostic approach can be plugged

into existing dynamic scene graph generation methods. In particular, the framework includes two stages: (1) Relation feature extraction, where we use existing dynamic scene graph generation methods to obtain the feature representation of the relation instances and (2) multi-label meta-weighting learning. We adopt a meta-learning framework to re-weight each instance's multi-label loss, and we propose to learn an explicit weighting function mapping from training loss to weight vector. We learn a weight vector for each training instance to re-weight its multi-label loss, *i.e.* multi-label binary cross-entropy loss. Specifically, we learn the weight vector through an MLP. The input of the MLP is the multi-label training loss, and the weight vector forms the output. We sample a meta-validation set used to guide the training of MLP. Ideally, the meta-validation set should be clean and without the long-tailed issue, as in [110]. However, we cannot sample such a clean meta-validation set due to the label-occurrence issue. To deal with the issue, we adopt the inverse frequency binary cross-entropy loss on meta-validation set. Below, we describe the ML-MWN and the meta-learning framework in detail.

5.3.1 *Learning weights for multi-label losses*

Let x_i denotes the feature representation of i-th relation instance from the training set $\mathcal{D}$ and $y_i \in \mathbb{R}^C$ is the corresponding multi-label one-hot vector, where $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$. f_θ represents the multi-label predicate classifier network with θ the parameters. In the presence of long-tailed multi-label training instances, we impose weights $w_{i,c}$ on the i-th instance's c-th class loss $l_{i,c}$ to enhance the robustness of training. Unlike existing methods that pre-specify the weights based on the class size [81, 119], we learn an explicit weighting function directly from data. Specifically, we propose the ML-MWN (Multi-Label Meta Weight Net) denoted by g_ϕ, with ϕ as its parameters, to obtain the weighting vector for each relation instance's multi-label loss. We take the loss from f_θ as the input of g_ϕ. A small meta-validation set $\widehat{\mathcal{D}} = \{x_j, y_j\}_{j=1}^M$, where M is the number of meta-validation instances and $M \ll N$, is sampled to guide the training of ML-MWN. The meta-validation set does not overlap with the training set. The weighted losses are then calculated to guarantee that the learned multi-label predicate classifier is unbiased to dominant classes. During training, the optimal classifier parameter θ^* can be extracted by minimizing the training loss:

$$L^{train}(\theta) = \frac{1}{n}\frac{1}{C}\sum_{i=1}^{n}\sum_{c=1}^{C} w_{i,c} \cdot l_{i,c}, \tag{5.1}$$

where n is the number of training instances in a batch, and C is the number of classes. During inference, we only use the optimal classifier network f_{θ^*} for evaluation.

5.3.2 *The meta-learning process*

We adopt a meta-learning framework is adopted to update the classifier and ML-MWN. The meta-validation set represents the unbiased relation instances following the balanced predicate class distribution. Due to the above-mentioned label-occurrence issue of multi-label classification [150], we use an inverse frequency BCE loss on the meta-validation

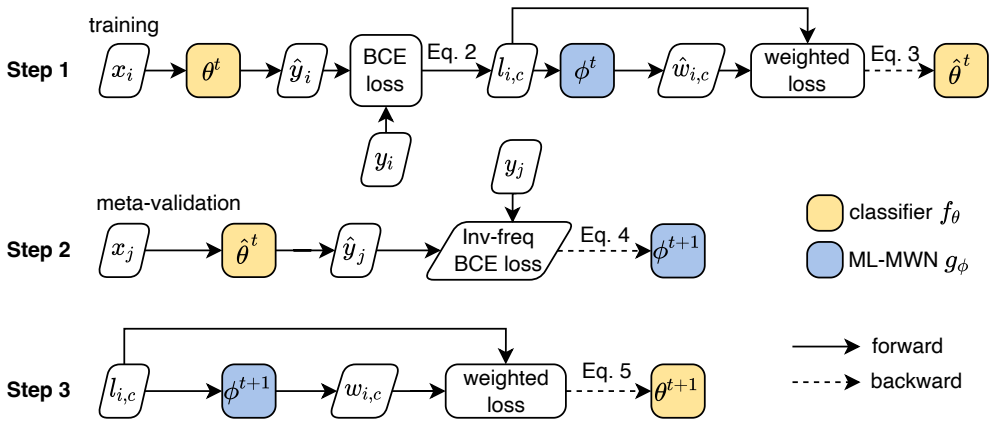


Figure 26: The overview of our proposed meta-learning process. Here we ignore the relation feature extraction part for simplification. x_i and x_j represent the feature of the training instance and meta-validation instance. During a training batch, there are three steps: 1. calculate the weighted loss and obtain a pseudo classifier; 2. evaluate the pseudo classifier on the meta-validation set and update the ML-MWN; 3. calculate the new weighted loss with updated ML-MWN then update the classifier. BCE (Binary Cross-Entropy) loss is adopted for multi-label classification. Inv-freq BCE loss represents inverse frequency BCE loss, which we use to mimic an unbiased meta-validation set. During inference, we only use the predicate multi-label classifier network for evaluation.

set to mimic the balanced label distribution. As shown in Figure 26, the process contains three main steps to optimize θ and ψ within a batch.

Suppose we are at t-th iteration during training. As a first step, in a batch of n training instances, we have their corresponding feature representations and multi-labels $\{x_i, y_i\}, 1 \leq i \leq n$. We feed x_i into and obtain $\hat{y}_i = f_{\theta^t}(x_i) \in \mathbb{R}^C$. The unweighted BCE training loss is calculated as

$$l_{i,c}(\theta^t) = -y_{i,c} \cdot \log\left(\hat{y}_{i,c}(\theta^t)\right) + (1 - y_{i,c}) \cdot \log\left(1 - \hat{y}_{i,c}(\theta^t)\right). \tag{5.2}$$

Then $l_{i,c}$ is fed into the ML-MWN to obtain the weight $\hat{w}_{i,c} = g_{\phi^t}\left(l_{i,c}(\theta^t)\right)$. After, we calculate the weighted loss as $\hat{w}_{i,c} \cdot l_{i,c}$ and update the θ^t. We have:

$$\hat{\theta}^t = \theta^t - \alpha \frac{1}{n} \frac{1}{C} \sum_{i=1}^{n} \sum_{c=1}^{C} g'_{\phi^t}(l_{i,c}(\theta^t)) \nabla_{\theta^t} l_{i,c}(\theta^t) \Big|_{\theta^t}, \tag{5.3}$$

where α is the step size. Here we call updated $\hat{\theta}^t$ the pseudo classifier parameters since the $\hat{\theta}^t$ is not used for the next batch.

The second step is to update the ML-MWN parameters based on the meta-validation loss. We feed the meta-validation relation instance to the pseudo classifier and obtain $\hat{y}_j = f_{\hat{\theta}^t}(x_j) \in \mathbb{R}^C$. Let M_c denote the total number of relation instances belonging to predicate class $c \in \{1, \ldots, C\}$. We use the inverse frequency BCE loss The frequency of a predicate class is calculated as $freq(c) = M_c / M$. By using inverse frequency

Algorithm 1 The ML-MWN learning algorithm

Require: Training data set $\mathcal{D}$, meta-validation set $\widehat{\mathcal{D}}$, max epochs N_{Epoch}
Ensure: Predicate multi-label classifier network parameter θ^*

```
for t = 1 to N_Epoch do
    for each mini batch {x_i, y_i}, 1 ≤ i ≤ n do
        Calculate the prediction ŷ_i.
        Calculate the unweighted loss using Eq. 5.2.
        Formulate the pseudo predicate classifier θ̂^t by Eq. 5.3.
        Get meta-validation instances {x_j, y_j} ∈ D̂.
        Update φ^{t+1} by Eq. 5.4.
        Update θ^{t+1} by Eq. 5.5.
    end for
end for
```

weighting, the meta-validation loss is re-balanced to mimic a balanced predicate label distribution. We then update the ML-MWN parameters ϕ on the meta-validation data:

$$\begin{aligned}\phi^{t+1} &= \phi^t - \beta \frac{1}{M}\sum_{j=1}^{M}\sum_{c=1}^{C}\frac{1}{freq(c)}\nabla_{\phi^t} l_{j,c}(\hat{\theta}^t)\Big|_{\phi^t} \\ &= \phi^t - \beta \sum_{c=1}^{C}\frac{M}{M_c}\nabla_{\phi^t} l_{j,c}(\hat{\theta}^t)\Big|_{\phi^t},\end{aligned} \tag{5.4}$$

where β is the step size.

Lastly, the updated ϕ^{t+1} is employed to output the new weights $w_{i,c}$. The new weighted losses are used to improve the parameters θ of the classifier network:

$$\theta^{t+1} = \theta^t - \alpha\frac{1}{n}\frac{1}{C}\sum_{i=1}^{n}\sum_{c=1}^{C} g'_{\phi^{t+1}}(l_{i,c}(\theta^t))\nabla_{\theta^t} l_{i,c}(\theta^t)\Big|_{\theta^t}. \tag{5.5}$$

The ultimate goal is to guide the classifier network to have a balanced performance on the unbiased meta-validation set. The sequences of steps are shown in Algorithm 1. By alternating standard and meta-learning, we can learn unbiased dynamic scene graphs by increasing the focus specifically on those examples and predicate classes that do not often occur in a dataset.

5.4 EXPERIMENTS

5.4.1 *Datasets*

Action Genome

AG [66] is a dataset which provides frame-level scene graph labels. It contains 234,253 annotated frames with 476,229 bounding boxes of 35 object classes (without person) and 1,715,568 instances of 25 relationship classes. For 25 relationships, there are three

different types: (1) attention relationships indicating if a person is looking at an object or not, (2) spatial relationships describing where objects are relative to one another, and (3) contact relationships denoting the different ways the person is contacting an object. In AG, there are 135,484 subject-object pairs. Each pair is labeled with multiple spatial relationships (*e.g.* ⟨phone-in front of-person⟩ and ⟨phone-on the side of-person⟩) or contact relationships (*e.g.* ⟨person-eating-food⟩ and ⟨person-holding-food⟩). There are three strategies to generate a scene graph with the inferred relation distribution [28]: (a) *with constraint* allows each subject-object pair to have one predicate at most. (b) *semi constraint* allows a subject-object pair has multiple predicates. The predicate is regarded as positive only if the corresponding confidence is higher than the threshold (0.9 in the experiments). (c) *no constraint* allows a subject-object pair to have multiple relationships guesses without constraint.

EVALUATION METRICS. We have three tasks for evaluation following [28]: (1) predicate classification (PREDCLS): with the subject and object's ground truth labels and bounding boxes, only predict predicate labels of the subject-object pair. (2) scene graph classification (SGCLS): with the subject and object's ground truth bounding boxes given, predict the subject, object's label and their corresponding predicate. (3) scene graph detection (SGDET): detect the subject and object's bounding boxes and predict the subject, object, and predicate's labels. The object detection is regarded as positive if the IoU between the predicted and ground-truth box is at least 0.5. Since traditional metrics Recall@K (R@K) are not able to reflect the impact of long-tailed data, we use the mean Recall@K (mR@K), which evaluates the R@K (k = [10, 20, 50]) of each relationship class separately and averages them.

IMPLEMENTATION DETAILS. In line with [28], we adopt the Faster-RCNN [101] based on the ResNet101 [54] as the object detection backbone. The Faster-RCNN model is trained on AG and provided by Cong *et al.* [28]. We use an AdamW [88] optimizer with an initial learning rate $1e^{-4}$ and batch size 1 to train our relation feature model STTran part. We train ML-MWN using SGD with a momentum of 0.9, weight decay of 0.01, and an initial learning rate of 0.01. We train 10 epochs. Other hyperparameter settings are identical to Cong *et al.* [28]. If not specified, the ML-MWN is an MLP of 1-100-1.

VidOR

VidOR [107] contains 10,000 user-generated videos selected from YFCC-100M [118], for a total of about 84 hours. There are 80 object categories and 50 predicate categories. Besides providing annotated relation triplets, the dataset also provides the bounding boxes of objects. The dataset is split into a training set with 7,000 videos, a validation set with 835 videos, and a testing set with 2,165 videos. Since the ground truth of the test set is not available, we use the training set for training and the validation set for testing, following [85, 98, 114, 136]. We report the analysis of method performance on the VidOR validation set.

	PredCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
STTran [28]	37.96	39.65	39.66	27.61	28.14	28.14	17.89	21.76	22.89
STTran + MW-Net [110]	40.29	42.21	42.24	30.21	30.90	30.90	20.06	23.66	24.99
STTran + ML-MWN	**43.23**	**44.43**	**44.64**	**32.13**	**32.70**	**32.72**	**23.46**	**27.13**	**28.52**

Table 13: Evaluating the effect of meta learning on Action Genome in the *with constraint* setting. Enriching the recent STTran approach with meta learning improves recall across all metrics, with the best results achieved using the proposed multi-label meta weighting.

	PredCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
STTran [28]	49.94	59.07	59.77	40.17	44.27	44.51	21.63	31.36	40.96
STTran + MW-Net [110]	52.61	62.32	63.1	43.12	47.11	47.77	24.19	34.83	43.85
STTran + ML-MWN	**55.95**	**65.79**	**68.01**	**46.20**	**50.60**	**50.83**	**26.21**	**40.12**	**49.96**

Table 14: Evaluating the effect of meta learning on Action Genome in the *semi constraint* setting. Similar to the *with constraint* setting, the proposed multi-label meta weighting obtains the best results across all metrics.

EVALUATION METRICS. We use the relation detection task for evaluation. The output requires a ⟨subject, predicate, object⟩ triplet prediction, and the subject and object boxes. We adopt mR@K (K = [50, 100]) as the evaluation metric. Here we ignore the mAP used in Chen *et al.* [24] since we care more about covering the ground truth relationship belonging to tail classes during the predictions. **Calculating mR@K.** For annotated video I_v, in its G_v ground truth relationship triplets, there are $G_{v,c}$ ground truth triplets with relationship class c. The number of relationship classes is C, where $T^K_{v,c}$ triplets are predicted successfully by the model. In V videos of validation/test dataset, for relationship c, there are V_c videos which contain at least one ground truth triplet with this relationship. The R@K of relationship c can be calculated:

$$R@K_c = \frac{1}{V_c} \sum_{v=1, G_{v,c} \neq 0}^{V_c} \frac{T^K_{v,c}}{G_{v,c}}. \tag{5.6}$$

Then we can calculate

$$mR@K = \frac{1}{C} \sum_{c=1}^{C} R@K_c. \tag{5.7}$$

IMPLEMENTATION DETAILS. We adopt the same training strategy of Chen *et al.* [24] for the relation feature extraction model. First, we detect all the objects per

	PredCLS			SGCLS			SGDET		
	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50	mR@10	mR@20	mR@50
STTran [28]	52.61	68.30	82.90	42.52	51.14	64.77	21.64	30.64	35.53
STTran + MW-Net [110]	55.12	70.42	85.45	45.55	54.46	67.24	24.24	34.25	37.98
STTran + ML-MWN	**57.13**	**74.22**	**89.24**	**48.48**	**57.65**	**70.15**	**27.59**	**36.67**	**40.57**

Table 15: Evaluating the effect of meta learning on Action Genome in the *no constraint* setting. Also in this challenging setting, our approach works best over all metrics.

video frame by Faster R-CNN [101] with a ResNet-101 [54] backbone. The detector is trained on MS-COCO [84]. The detected bounding boxes are linked with the Deep SORT tracker [133] to obtain individual object tubelets. Then each tubelet is paired with any other tubelet to generate the tubelet pairs. We extract spatial location features [115], language features, I3D features, and location mask features for each pair. Then the multi-modal features are used as the representation of the elation instance. For the classifier and ML-MWN, we use an SGD optimizer with an initial learning rate of 0.01. We train 10 epochs.

5.4.2 *Multi-label meta weighting on top of the state-of-the-art*

VIDEO SCENE GRAPH GENERATION. First, we investigate the effect of incorporating our meta learning approach on top of existing state-of-the-art methods for scene graph generation in videos and video relation detection. We build upon the recent STTran approach of Cong *et al.* [28] for video scene graph generation. We compare STTran as is and a baseline that uses conventional meta-learning without considering the multi-label nature of scene graphs, namely MW-Net [110]. Table 13 shows the results for the setting *with constraints*. Across the PredCLS, SGCLS, and SGDET tasks, incorporating our meta-learning approach improves the results. For PredCLS, our proposed STTran + ML-MWN improved mR@10 by **5.27**, compared to the STTran baseline. On mean recall @ 50, we improve the scores by **4.98**, from 39.66 to 44.64. On SGDET, the mean recall @ 50 goes up from 22.89 to 28.52. The MW-Net baseline already improves the STTran results, highlighting the overall potential of meta-learning to deal with the long-tailed nature of scene graphs. However, the proposed multi-label meta learning performs best across all tasks and recall thresholds. This is a direct consequence of increasing the weight of classes in the long tail when optimizing the classifier network.

The results also hold for the *semi constraint* and *no constraint* settings, as shown in Table 14 and Table 15. In Table 14, the mean recall is higher than on the *with constraint* setting since more predicted results are involved. For the SGCLS task, our framework achieves 50.60% on mR@20, 6.33% better than STTran and 3.49% better than STTran + MW-Net. Our framework is also best for all metrics in the *no constraint* setting. In particular, for SGDET, our method reaches 27.59% at mR@10, 5.95% better than STTran, and 3.35% higher than STTran + MV-Net. We conclude that our meta learning framework is effective for video scene graph generation and can be adopted by any existing work. In Table 15, the mean recall is the highest among the three settings. Unlimited predictions will boost the recall performance. Under such a setting, STTran + ML-MWN still achieves the best on all metrics among all tasks. The results prove our method's generality on different tasks with different settings.

VIDEO RELATION DETECTION. For video relation detection, we start from the recent Social Fabric approach by Chen *et al.* [24]. Table 16 shows the effect of incorporating the proposed meta learning framework for relation detection. The Social Fabric baseline is the current state-of-the-art in this setting yet struggles to get good results for relation detection with mean recall as metrics. This highlights the difficulty of the problem. This holds similarly for the baseline by Sun etal [115]. When adding

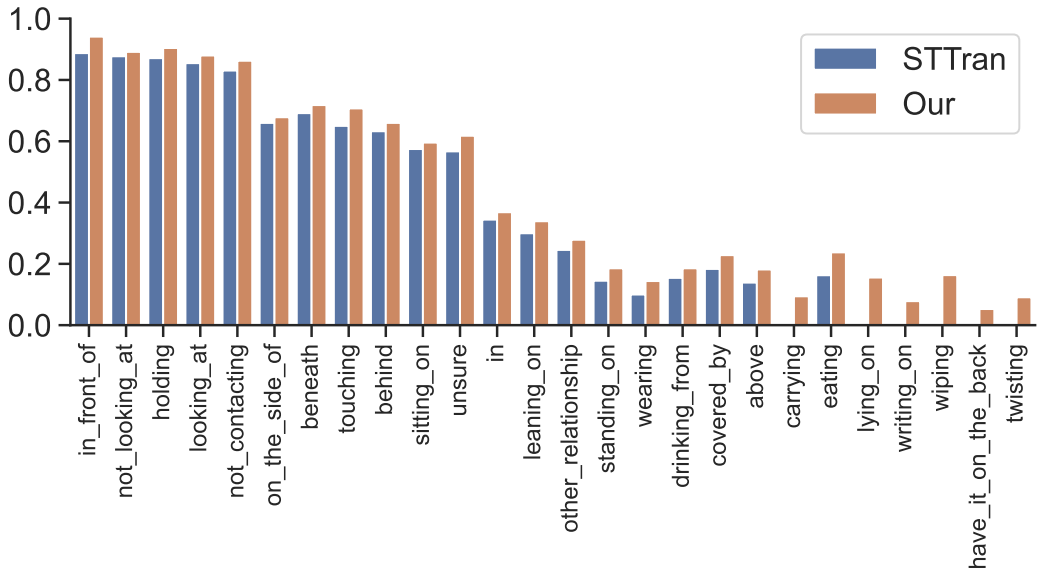


Figure 27: Class-wise R@10 comparison of PredCLS on AG. Our framework outperforms STTran on all predicate classes.

Method	Relation detection	
	mR@50	mR@100
Sun *et al.* [115]	1.48	2.78
Social Fabric (SF) [24]	2.37	3.79
SF + MW-Net [110]	4.45	5.35
SF + ML-MWN	**6.35**	**7.54**

Table 16: Comparison on the VidOR dataset. Our meta learning framework provides clear improvements for relation detection on top of Social Fabric [24].

MW-Net [110], the results already clearly improve and improve further with multi-label meta weighting. For mR@50, adding our meta learning on top of Social Fabric improves the results from 2.37 to 6.35. We conclude that multi-label meta learning is key in video relation detection to achieve meaningful relation detection recalls over all classes.

5.4.3 *Analyses, ablations, and qualitative examples*

PREDICATE-LEVEL ANALYSIS. We show the class-wise R@10 of predicate classification task on Action Genome in Figure 27. We observe from Figure 27 that our method outperforms STTran [28] in all predicate categories. For those tail classes with limited training samples, the improvement is much bigger than the head classes. The out-performance proves that the meta-validation set successfully guides the classifier to balance the tail classes, without scarifying the head predicate classes' performance.

ABLATING THE MLP ARCHITECTURE. We ablate the MLP architecture on Action Genome for the PredCLS task. Table 17 depicts the results for six structures with

Architecture	PredCLS		
	mR@10	mR@50	mR@100
C-50-*C*	41.34	42.24	42.56
C-100-*C*	**43.23**	**44.43**	**44.64**
C-200-*C*	42.16	42.85	42.97
C-100-100-*C*	43.01	43.96	44.03
C-10-10-*C*	42.18	42.74	42.96
C-10-10-10-*C*	42.85	44.01	44.28

Table 17: Performance on AG with constraint for different MLP architecture. The 1-100-1 architecture is the best.

varying depths and widths. We find that maximum width and depth are not required, with the best results for the 1-100-1 variant, which we use as default in all experiments.

QUALITATIVE EXAMPLES. We provide the qualitative results in Figure 28 and Figure 29. In Figure 28, we compare our method with STTran [28] on the Action Genome dataset. Our method has better recognition of tail predicates in the Action Genome. In the top row, STTran incorrectly classifies the tail class *beneath* as head class *in front of*, and *sit on* as *touch*. In the bottom row, STTran misses *drink from* amongst others, while our method classifies them all correctly. In Figure 29, we compare our method with Social Fabric [24] on the VidOR dataset. Social Fabric misses the tail class *lean_on* on all frames while our method successfully predicts.

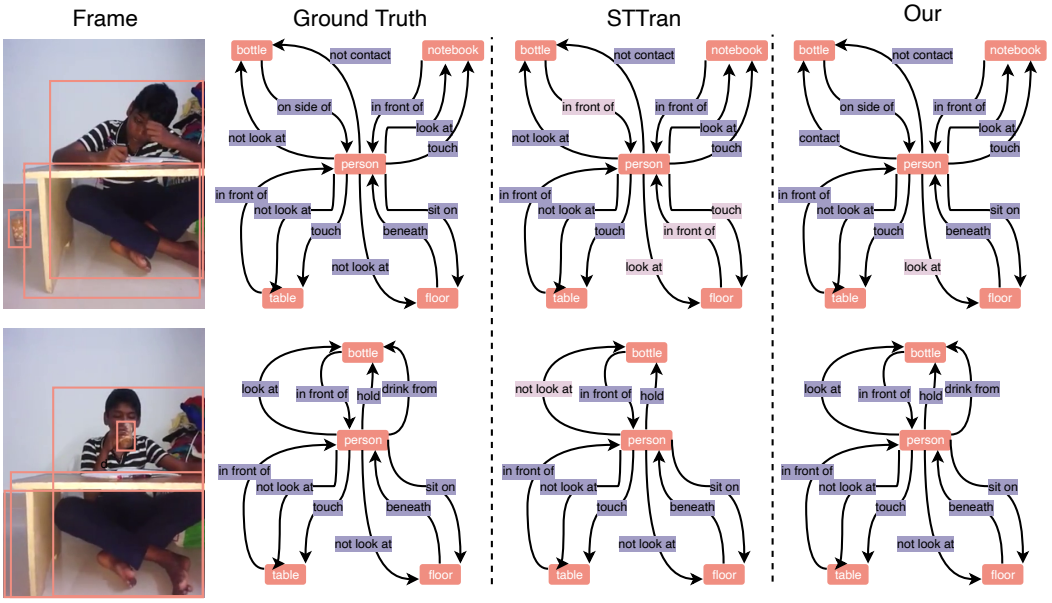


Figure 28: Qualitative comparison on Action Genome predicate classification task. The gray box is the wrongly recognized predicates. Our method performs better than STTran [28] on recognizing the tail and the head both.

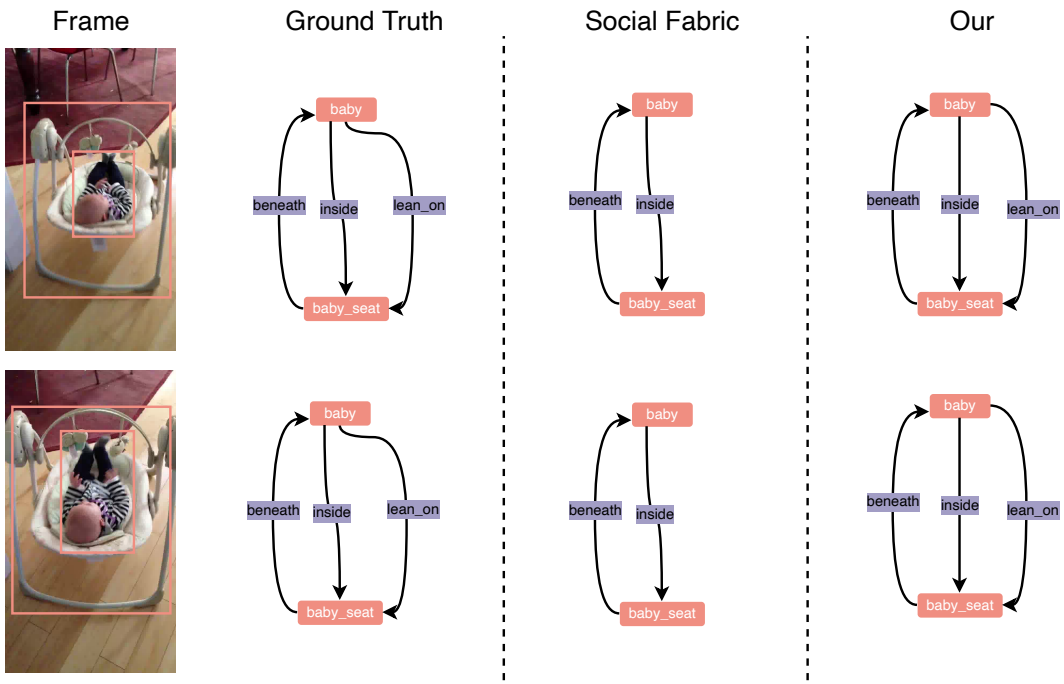


Figure 29: Qualitative comparison on VidOR predicate classification. The Social Fabric baseline [24] misses the *lean_on* predicate, while our method detects it correctly.

5.5 CONCLUSION

Predicate recognition plays a central role in current dynamic scene graph generation methods, but the long-tailed and multi-label nature of the predicate distribution is commonly ignored. We find that rare predicates on popular benchmarks are poorly recovered or even ignored in recent methods. To make the step toward unbiased scene graph generation in videos, we propose a multi-label meta learning framework that learns to weight samples and classes to help optimize any predicate classifier. Our approach is generic and can be plugged into any existing methods. Experiments on two benchmarks and two recent methods show the potential of our meta learning framework, with better overall performance and an improved focus on rare predicates.

6

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

This thesis investigates the spatio-temporal perception of interactivity in videos. Specifically, this thesis focuses on the research question: *how to automate the perception of interactivity in video content?* We start with the definition of interactivity, followed by the recognition of interactivity, then their error analysis, and finally recognizing the rare interactivities. A brief summary of each chapter is provided as follows:

Chapter 2: This chapter introduces spatio-temporal interactivity proposals for video surveillance. Rather than focusing solely on actions performed by subjects, we explicitly include the objects that the subjects interact with. To enable interactivity proposals, we introduce the notion of interactivityness, a score that reflects the likelihood that a subject and object have an interplay. For its estimation, we propose a network containing an interactivity block and geometric encoding between subjects and objects. The network computes local interactivity likelihoods from subject and object trajectories, which we use to link intervals of high scores into spatio-temporal proposals. Experiments on an interactivity dataset with new evaluation metrics show the general benefit of interactivity proposals as well as its favorable performance compared to traditional temporal and spatio-temporal action proposals.

Chapter 3: Here we strive to classify and detect the relationship between object tubelets appearing within a video as a ⟨*subject-predicate-object*⟩ triplet. Where existing works treat object proposals or tubelets as single entities and model their relations *a posteriori*, we propose to classify and detect predicates for pairs of object tubelets *a priori*. We also propose Social Fabric: an encoding that represents a pair of object tubelets as a composition of interaction primitives. These primitives are learned over all relations, resulting in a compact representation able to localize and classify relations from the pool of co-occurring object tubelets across all timespans in a video. The encoding enables our two-stage network. In the first stage, we train Social Fabric to suggest proposals that are likely interacting. We use the Social Fabric in the second stage to simultaneously fine-tune and predict predicate labels for the tubelets. Experiments demonstrate the benefit of early video relation modeling, our encoding and the two-stage architecture, leading to a new state-of-the-art on two benchmarks. We also show how the encoding enables query-by-primitive-example to search for spatio-temporal video relations.

Chapter 4: Video relation detection forms a new and challenging problem in computer vision, where subjects and objects need to be localized spatio-temporally and a predicate label needs to be assigned if and only if there is an interaction between the two. Despite recent progress in video relation detection, overall performance is still marginal and

it remains unclear what the key factors are towards solving the problem. Following examples set in the object detection and action localization literature, we perform a deep dive into the error diagnosis of current video relation detection approaches. We introduce a diagnostic tool for analyzing the sources of detection errors. Our tool evaluates and compares current approaches beyond the single scalar metric of mean Average Precision by defining different error types specific to video relation detection, used for false positive analyses. Moreover, we examine different factors of influence on the performance in a false negative analysis, including relation length, number of subject/object/predicate instances, and subject/object size. Finally, we present the effect on video relation performance when considering an oracle fix for each error type. On two video relation benchmarks, we show where current approaches excel and fall short, allowing us to pinpoint the most important future directions in the field.

Chapter 5: Recognizing the predicate between subject and object pairs is imbalanced and multi-label in nature, ranging from ubiquitous interactions such as spatial relationships (*e.g. in front of*) to rare interactions such as *twisting*. In popular benchmarks such as Action Genome and VidOR, the imbalance ratio between most and least frequent predicates is 3218 and 3408 respectively, far higher even than benchmarks specifically designed to address long-tailed recognition. Due to these long-tailed distributions and label co-occurrences, recent state-of-the-art methods rely heavily on the most often occurring predicate classes, ignoring predicate classes in the long tail. In this chapter, we analyze the limitations of current approaches for scene graph generation in videos and find a one-to-one correspondence between predicate frequency and recall performance. To make the step towards unbiased scene graph generation in videos, we introduce a multi-label meta-learning framework to deal with the biased predicate distribution. Our meta-learning framework learns a meta-weight network for each training sample over all possible label losses. We evaluate our approach on the Action Genome and VidOR benchmarks by building on two current state-of-the-art methods for each benchmark. The experiments confirm that our multi-label meta-weight network improves the performance for predicates in the long tail without hampering performance for head classes, resulting in better overall performance and favorable generalizability.

6.2 CONCLUSIONS

This thesis investigated the machine perception of interactivity in videos, addressing aspects such as detection, prediction, and analysis. Despite progress, machines still require substantial human-annotated training data to effectively perceive an interactivity, unlike humans. Future work could explore several promising directions to advance detecting interactivity in videos with less dependence on annotations. One possibility is through weakly-supervised learning, reducing reliance on richly annotated data. Current methods require dense annotations for each frame, but single frame supervision could greatly reduce manual annotations. Moreover, leveraging the temporal correlation between frames through self-supervised learning for interactivity feature representations offers potential. Finally, incorporating multi-modal data sources, such as audio and text, could enhance the understanding of interactivity by providing complementary information.

BIBLIOGRAPHY

[1] A. Agrawal, D. Batra, and D. Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 2012.

[3] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018.

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *CVPR*, 2015.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[6] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, J. Magalhaes, D. Semedo, and S. Blasi. TRECVID 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *TRECVID*, 2018.

[7] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[8] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *ECCV Workshops*, 2014.

[9] B. L. Bhatnagar, X. Xie, I. A. Petrov, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. BEHAVE: dataset and method for tracking human object interactions. In *CVPR*, 2022.

[10] O. Bohdal, Y. Yang, and T. Hospedales. Evograd: Efficient gradient-based meta-learning and hyperparameter optimization. In *NeurIPS*, 2021.

[11] D. Bolya, S. Foley, J. Hays, and J. Hoffman. TIDE: A general toolbox for identifying object detection errors. In *ECCV*, 2020.

[12] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles. SST: single-stream temporal action proposals. In *CVPR*, 2017.

[13] P. Byvshev, P. Mettes, and Y. Xiao. Heterogeneous non-local fusion for multimodal activity recognition. In *ICMR*, 2020.

[14] Q. Cao, H. Huang, X. Shang, B. Wang, and T.-S. Chua. 3-d relation network for visual relation recognition in videos. *Neurocomputing*, 2021.

[15] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

[16] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann. A comprehensive survey of scene graphs: Generation and application. *PAMI Trans. Pattern Anal. Mach. Intell.*, 2023.

[17] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. In *WACV*, 2018.

[18] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. HICO: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015.

[19] J. Chen, J. Liu, J. Liang, T.-Y. Hu, W. Ke, W. Barrios, D. Huang, and A. G. Hauptmann. Minding the gaps in a video action analysis pipeline. In *WACV Workshops*, 2019.

[20] R. Chen, R. Little, L. Mihaylova, R. Delahay, and R. Cox. Wildlife surveillance using deep learning methods. *Ecology and evolution*, 2019.

[21] S. Chen, Y. Du, P. Mettes, and C. G. M. Snoek. Multi-label meta weighting for long-tailed dynamic scene graph generation. In *ICMR*, 2023.

[22] S. Chen, P. Mettes, T. Hu, and C. G. M. Snoek. Interactivity proposals for surveillance videos. In *ICMR*, 2020.

[23] S. Chen, P. Mettes, and C. G. M. Snoek. Diagnosing errors in video relation detectors. In *BMVC*, 2021.

[24] S. Chen, Z. Shi, P. Mettes, and C. G. M. Snoek. Social fabric: Tubelet compositions for video relation detection. In *ICCV*, 2021.

[25] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.

[26] M. Cheng, K. Cai, and M. Li. RWF-2000: an open large scale video database for violence detection. In *ICPR*, 2020.

[27] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 2020.

[28] Y. Cong, W. Liao, H. Ackermann, B. Rosenhahn, and M. Y. Yang. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, 2021.

[29] Q. Cui, H. Sun, and F. Yang. Learning dynamic relationships for 3d human motion prediction. In *CVPR*, 2020.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[31] A. Desai, T.-Y. Wu, S. Tripathi, and N. Vasconcelos. Learning of visual relations: The devil is in the tails. In *ICCV*, 2021.

[32] D. Di, X. Shang, W. Zhang, X. Yang, and T.-S. Chua. Multiple hypothesis video relation detection. In *BigMM*, 2019.

[33] X. Dong, T. Gan, X. Song, J. Wu, Y. Cheng, and L. Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, 2022.

[34] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *ICCV*, 2013.

[35] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016.

[36] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[37] H. Fan, F. Yang, P. Chu, Y. Lin, L. Yuan, and H. Ling. Tracklinic: Diagnosis of challenge factors in visual tracking. In *WACV*, 2021.

[38] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *PAMI*, 2013.

[39] C. Gao, Y. Zou, and J.-B. Huang. ican: Instance-centric attention network for human-object interaction detection. In *BMVC*, 2018.

[40] J. Gao, K. Chen, and R. Nevatia. CTAP: complementary temporal action proposal generation. In *ECCV*, 2018.

[41] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. TURN TAP: temporal unit regression network for temporal action proposals. In *ICCV*, 2017.

[42] K. Gao, L. Chen, Y. Niu, J. Shao, and J. Xiao. Classification-then-grounding: Reformulating video scene graphs as temporal bipartite graphs. In *CVPR*, 2022.

[43] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. M. Snoek. Actor-transformers for group activity recognition. In *CVPR*, 2020.

[44] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *CVPR*, 2019.

[45] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

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

[47] J. Gleason, R. Ranjan, S. Schwarcz, C. Castillo, J.-C. Chen, and R. Chellappa. A proposal-based solution to spatio-temporal action detection in untrimmed videos. In *WACV*, 2019.

[48] E. B. Goldstein. *Cognitive psychology: Connecting mind, research and everyday experience*. Cengage Learning, 2018.

[49] C. Graber and A. G. Schwing. Dynamic neural relational inference. In *CVPR*, 2020.

[50] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[51] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *PAMI*, 2009.

[52] J. He, Z. Deng, M. S. Ibrahim, and G. Mori. Generic tubelet proposals for action localization. In *WACV*, 2018.

[53] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

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

[55] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[56] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015.

[57] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.

[58] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018.

[59] Z. Hu, G. Feng, J. Sun, L. Zhang, and H. Lu. Bi-directional relationship inferring network for referring image segmentation. In *CVPR*, 2020.

[60] X. Huang, J. Wicaksana, S. Li, and K.-T. Cheng. Automated vision-based wellness analysis for elderly care centers. In *AAAI Workshops*, 2022.

[61] S. Inayoshi, K. Otani, A. Tejero-de Pablos, and T. Harada. Bounding-box channels for visual relationship detection. In *ECCV*, 2020.

[62] M. Jain, J. van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek. Tubelets: Unsupervised action proposals from spatiotemporal super-voxels. *IJCV*, 2017.

[63] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[64] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013.

[65] J. Ji, R. Desai, and J. C. Niebles. Detecting human-object relationships in videos. In *ICCV*, 2021.

[66] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, 2020.

[67] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015.

[68] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.

[69] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[70] T. S. Kim, M. Peven, W. Qiu, A. Yuille, and G. D. Hager. Synthesizing attributes with unreal engine for fine-grained activity analysis. In *WACV Workshops*, 2019.

[71] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *IJRR*, 2013.

[72] A. Kortylewski, Q. Liu, A. Wang, Y. Sun, and A. Yuille. Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion. *IJCV*, 2020.

[73] A. Kukleva, M. Tapaswi, and I. Laptev. Learning interactions and relationships between movie characters. In *CVPR*, 2020.

[74] R. Li, S. Zhang, B. Wan, and X. He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, 2021.

[75] W. Li, H. Zhang, Q. Bai, G. Zhao, N. Jiang, and X. Yuan. PPDL: predicate probability distribution based loss for unbiased scene graph generation. In *CVPR*, 2022.

[76] Y. Li, X. Yang, and C. Xu. Dynamic scene graph generation via anticipatory pre-training. In *CVPR*, 2022.

[77] Y.-L. Li, S. Zhou, X. Huang, L. Xu, Z. Ma, H.-S. Fang, Y. Wang, and C. Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019.

[78] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *CVPR*, 2019.

[79] Y. Liao, S. Liu, F. Wang, Y. Chen, C. Qian, and J. Feng. PPDM: parallel point detection and matching for real-time human-object interaction detection. In *CVPR*, 2020.

[80] R. Lin, J. Xiao, and J. Fan. Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In *ECCV*, 2018.

[81] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[82] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. BSN: boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

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

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

[85] C. Liu, Y. Jin, K. Xu, G. Gong, and Y. Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *CVPR*, 2020.

[86] H. Liu, N. Yan, M. Mortazavi, and B. Bhanu. Fully convolutional scene graph generation. In *CVPR*, 2021.

[87] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, 2019.

[88] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

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

[90] L. Mi and Z. Chen. Hierarchical graph attention network for visual relationship detection. In *CVPR*, 2020.

[91] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. In *CVPR Workshops*, 2017.

[92] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR Workshops*, 2013.

[93] I. Misra, A. Shrivastava, and M. Hebert. Watch and learn: Semi-supervised learning for object detectors from video. In *CVPR*, 2015.

[94] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[95] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014.

[96] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkilä. Uncovering Hidden Challenges in Query-Based Video Moment Retrieval. *BMVC*, 2020.

[97] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *PAMI*, 2012.

[98] X. Qian, Y. Zhuang, Y. Li, S. Xiao, S. Pu, and J. Xiao. Video relation detection with spatio-temporal graph. In *ACM MM*, 2019.

[99] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He. Precise temporal action localization by evolving temporal proposals. In *ICMR*, 2018.

[100] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[101] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[102] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[103] J. B. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 2000.

[104] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[105] M. L. Sandifort, J. Liu, S. Nishimura, and W. Hürst. An entropy model for loiterer retrieval across multiple surveillance cameras. In *ICMR*, 2018.

[106] M. L. Sandifort, J. Liu, S. Nishimura, and W. Hürst. Visloiter+: An entropy model-based loiterer retrieval system with user-friendly interfaces. In *ICMR*, 2018.

[107] X. Shang, D. Di, J. Xiao, Y. Cao, X. Yang, and T.-S. Chua. Annotating objects and relations in user-generated videos. In *ICMR*, 2019.

[108] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua. Video visual relation detection. In *ACM MM*, 2017.

[109] X. Shang, J. Xiao, D. Di, and T.-S. Chua. Relation understanding in videos: A grand challenge overview. In *ACM MM*, 2019.

[110] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *NeurIPS*, 2019.

[111] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[112] C. G. M. Snoek and M. Worring. *Concept-based video retrieval*. Now Publishers Inc, 2009.

[113] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[114] Z. Su, X. Shang, J. Chen, Y.-G. Jiang, Z. Qiu, and T.-S. Chua. Video relation detection via multiple hypothesis association. In *ACM MM*, 2020.

[115] X. Sun, T. Ren, Y. Zi, and G. Wu. Video visual relation detection via multi-modal feature fusion. In *ACM MM*, 2019.

[116] S. P. R. Sunkesula, R. Dabral, and G. Ramakrishnan. LIGHTEN: learning interactions with graph and hierarchical temporal networks for HOI in videos. In *ACM MM*, 2020.

[117] E. Swears, A. Hoogs, Q. Ji, and K. Boyer. Complex activity recognition using granger constrained DBN (GCDBN) in sports and surveillance video. In *CVPR*, 2014.

[118] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: the new data in multimedia research. *Communications of the ACM*, 2016.

[119] J. Tian, N. C. Mithun, Z. Seymour, H.-P. Chiu, and Z. Kira. Striking the right balance: Recall loss for semantic segmentation. In *ICRA*, 2022.

[120] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *ACL*, 2019.

[121] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, 2019.

[122] J. C. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek. APT: action localization proposals from dense trajectories. In *BMVC*, 2015.

[123] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *PAMI*, 2010.

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

[125] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014.

[126] B. Wan, D. Zhou, Y. Liu, R. Li, and X. He. Pose-aware multi-level feature network for human object interaction detection. In *ICCV*, 2019.

[127] H. Wang, S. Pirk, E. Yumer, V. G. Kim, O. Sener, S. Sridhar, and L. J. Guibas. Learning a generative model for multi-step human-object interactions from videos. *Computer Graphics Forum*, 2019.

[128] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. Actionness estimation using hybrid fully convolutional networks. In *CVPR*, 2016.

[129] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[130] T. Wang, T. Yang, M. Danelljan, F. S. Khan, X. Zhang, and J. Sun. Learning human-object interaction detection using interaction points. In *CVPR*, 2020.

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

[132] X. Wang and Q. Ji. A hierarchical context model for event recognition in surveillance video. In *CVPR*, 2014.

[133] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[134] J. Wu, H. Hsieh, D. Chen, C. Fuh, and T. Liu. Self-supervised sparse representation for video anomaly detection. In *ECCV*, 2022.

[135] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, 2020.

[136] W. Xie, G. Ren, and S. Liu. Video relation detection with trajectory-aware multi-modal features. In *ACM MM*, 2020.

[137] B. Xu, Y. Wong, J. Li, Q. Zhao, and M. S. Kankanhalli. Learning to detect human-object interactions with knowledge. In *CVPR*, 2019.

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

[139] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[140] S. Yan, C. Shen, Z. Jin, J. Huang, R. Jiang, Y. Chen, and X.-S. Hua. PCPL: predicate-correlation perception learning for unbiased scene graph generation. In *ACM MM*, 2020.

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

[142] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[143] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[144] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[145] A. L. Yuille. Towards a theory of compositional learning and encoding of objects. In *ICCV Workshops*, 2011.

[146] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. N. Asghar, and B. Lee. A survey of modern deep learning based object detection models. *Digit. Signal Process.*, 2022.

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

[148] Y. Zhan, J. Yu, T. Yu, and D. Tao. On exploring undetermined relationships for visual relationship detection. In *CVPR*, 2019.

[149] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. How far are we from solving pedestrian detection? In *CVPR*, 2016.

[150] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

[151] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[152] Z. Zhao, X. Li, X. Du, Q. Chen, Y. Zhao, F. Su, X. Chang, and A. G. Hauptmann. A unified framework with a benchmark dataset for surveillance event detection. *Neurocomputing*, 2018.

[153] S. Zheng, X. Chen, S. Chen, and Q. Jin. Relation understanding in videos. In *ACM MM*, 2019.

[154] H. Zhu, S. Lu, J. Cai, and Q. Lee. Diagnosing state-of-the-art object proposal method. In *BMVC*, 2015.

[155] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li. A comprehensive study of deep video action recognition. *arXiv:2012.06567*, 2020.

COMPLETE LIST OF PUBLICATIONS

- **Shuo Chen**, Yingjun Du, Pascal Mettes, and Cees G. M. Snoek, "Multi-Label Meta Weighting for Long-Tailed Dynamic Scene Graph Generation", in submission to **ACM International Conference on Multimedia Retrieval**, 2023.
- Wenjing Chu, **Shuo Chen**, and Marcello Bonsangue. "Non-linear optimization methods for learning regular distributions", **International Conference on Formal Engineering Methods**, 2022.
- **Shuo Chen**, Pascal Mettes, Cees G. M. Snoek, "Diagnosing Errors in Video Relation Detectors", **British Machine Vision Conference**, 2021.
- **Shuo Chen,** Tan Yu, and Ping Li. "MVT: Multi-view Vision Transformer for 3D Object Recognition", **British Machine Vision Conference**, 2021.
- Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek, "Social Fabric: Tubelet Compositions for Video Relation Detection", **IEEE International Conference on Computer Vision**, 2021.
- Wenjing Chu, **Shuo Chen**, and Marcello Bonsangue. "Learning Probabilistic Automata Using Residuals", **International Confederation for Thermal Analysis and Calorimetry**, 2021.
- **Shuo Chen**, Pascal Mettes, Tao Hu, and Cees G. M. Snoek, "Interactivity Proposals for Surveillance Videos", **ACM International Conference on Multimedia Retrieval**, 2020.
- Sarah Ibrahimi, **Shuo Chen**, Devanshu Arya, Arthur Camara, Yunlu Chen, Tanja Crijns, Maurits van der Goes, Thomas Mensink, Emiel van Miltenburg, Daan Odijk, William Thong, Jiaojiao Zhao,and Pascal Mettes, "Interactive Exploration of Journalistic Video Footage through Multimodal Semantic Matching", **ACM International Conference on Multimedia**, 2019.
- **Shuo Chen**, Fei Zhou, and Qingmin Liao, "Visual Domain Adaptation using Weighted Subspace Alignment", **IEEE International Conference on Visual Communications and Image Processing**, 2016.

SAMENVATTING

Dit proefschrift onderzoekt de tijdruimtelijke perceptie van interactiviteit in video's. Dit proefschrift richt zich met name op de onderzoeksvraag: *hoe kan de perceptie van interactiviteit in video-inhoud geautomatiseerd worden?* We beginnen met de definitie van interactiviteit, gevolgd door de herkenning van interactiviteit, vervolgens de bijbehorende foutenanalyse, en ten slotte het herkennen van de zeldzame interactiviteit. Een korte samenvatting van elk hoofdstuk wordt als volgt gegeven:

Hoofdstuk 2: Dit hoofdstuk introduceert tijdruimtelijke interactiviteitsvoorstellen voor videobewaking. In plaats van ons uitsluitend te focussen op de acties die door de onderwerpen worden uitgevoerd, nemen we expliciet de objecten op waarmee de onderwerpen interageren. Om interactiviteitsvoorstellen mogelijk te maken, introduceren we het begrip interactiviteitsgraad, een score die de waarschijnlijkheid weerspiegelt dat een onderwerp en object een wisselwerking hebben. Voor de schatting hiervan stellen we een netwerk voor met een interactiviteitsblok en geometrische codering tussen onderwerpen en objecten. Het netwerk berekent lokale interactiviteitswaarschijnlijkheden uit de trajecten van het onderwerp en het object, die we gebruiken om intervallen met hoge scores te koppelen aan tijdruimtelijke voorstellen. Experimenten op een interactiviteitsdataset met nieuwe evaluatiemetingen tonen het algemene voordeel van interactiviteitsvoorstellen en hun gunstige prestaties in vergelijking met traditionele temporele en tijdruimtelijke actievoorstellen.

Hoofdstuk 3: Hier streven we ernaar om de relatie tussen object 'tubelets' die binnen een video verschijnen te classificeren en te detecteren als een ⟨*onderwerp-predikaat-object*⟩ triplet. Waar bestaande werken objectvoorstellen of tubelets behandelen als enkele entiteiten en hun relaties *a posteriori* modelleren, stellen we voor om predikaten voor paren van object tubelets *a priori* te classificeren en te detecteren. We stellen ook Social Fabric voor: een codering die een paar object tubelets vertegenwoordigt als een compositie van interactieprimitieven. Deze primitieven worden geleerd over alle relaties, resulterend in een compacte weergave die in staat is om relaties te lokaliseren en te classificeren uit de pool van gelijktijdig voorkomende object tubelets over alle tijdsspannen in een video. De codering maakt ons twee fasen netwerk mogelijk. In de eerste fase trainen we Social Fabric om voorstellen te suggereren die waarschijnlijk interactief zijn. We gebruiken de Social Fabric in de tweede fase om tegelijkertijd te finetunen en predikaatlabels voor de tubelets te voorspellen. Experimenten tonen het voordeel aan van vroege videorelatiemodellering, onze codering en de tweefasenarchitectuur, wat leidt tot een nieuwe stand van zaken op twee benchmarks. We tonen ook aan hoe de codering query-door-primitievevoorbeeld in staat stelt om naar tijdruimtelijke videorelaties te zoeken.

Hoofdstuk 4: Videorelatiedetectie vormt een nieuw en uitdagend probleem in computer vision, waarbij onderwerpen en objecten tijdruimtelijk gelokaliseerd moeten worden en een predikaatlabel moet worden toegewezen als en alleen als er een interactie is tussen de twee. Ondanks de recente vooruitgang in videorelatiedetectie, is de algehele prestatie nog steeds marginaal en blijft het onduidelijk wat de sleutelfactoren zijn om het probleem op te lossen. Volgens voorbeelden in de literatuur over objectdetectie en actielokalisatie, duiken we diep in de foutendiagnose van huidige benaderingen van videorelatiedetectie. We introduceren een diagnostisch hulpmiddel voor het analyseren van de bronnen van detectiefouten. Ons hulpmiddel evalueert en vergelijkt huidige benaderingen verder dan de enkele scalaire metriek van gemiddelde precisie door verschillende fouttypen specifiek voor videorelatiedetectie te definiëren, gebruikt

voor analyses van valse positieven. Bovendien onderzoeken we verschillende factoren van invloed op de prestatie in een analyse van valse negatieven, waaronder relatielengte, aantal onderwerp/object/predikaat instanties, en onderwerp/object grootte. Ten slotte presenteren we het effect op videorelatieprestaties wanneer we een orakelfix overwegen voor elk fouttype. Op twee videorelatie benchmarks tonen we aan waar huidige benaderingen uitblinken en tekortschieten, waardoor we de belangrijkste toekomstige richtingen in het veld kunnen aangeven.

Hoofdstuk 5: Het herkennen van het predikaat tussen onderwerp en object paren is oneven-wichtig en multi-label van aard, variërend van alomtegenwoordige interacties zoals ruimtelijke relaties (*e.g. tegenover*) tot zeldzame interacties zoals *draaien*. In populaire benchmarks zoals Action Genome en VidOR is de disbalansverhouding tussen de meest en minst frequente predicaten respectievelijk 3218 en 3408, veel hoger zelfs dan benchmarks specifiek ontworpen om long-tailed herkenning aan te pakken. Door deze long-tailed distributies en label gelijktijdigheden vertrouwen recente state-of-the-art methoden sterk op de meest voorkomende predikaatklassen, waarbij ze de predikaatklassen in de long-tail negeren. In dit hoofdstuk analyseren we de beperkingen van huidige benaderingen voor scene graaf generatie in video's en vinden een een-op-een correspondentie tussen predikaat frequentie en recall-prestatie. Om de stap naar onbevooroordeelde scene graaf generatie in video's te maken, introduceren we een multi-label meta-learning raamwerk om de bevooroordeelde predikaatdistributie aan te pakken. Ons meta-learning raamwerk leert een meta-weight netwerk voor elke trainings sample over alle mogelijke label verliezen. We evalueren onze benadering op de Action Genome en VidOR benchmarks door voort te bouwen op twee huidige state-of-the-art methoden voor elke benchmark. De experimenten bevestigen dat ons multi-label meta-weight netwerk de prestaties voor predicaten in de long-tail verbetert zonder de prestaties voor hoofdklassen te belemmeren, resulterend in betere algehele prestaties en gunstige algemene toepasbaarheid.

ACKNOWLEDGMENTS

It is never an easy task to complete a PhD. The journey has been filled with learning experiences and unforgettable moments. There are countless sentiments I hope to show, and many people I wish to express my gratitude towards.

First, I would like to express my deepest appreciation to my supervisor, Cees, who consistently shows patience, humor, and expertise. Each meeting with him is a learning opportunity for me. Some moments from our initial interactions remain vivid. I still recall the fun events during my admissions interview with you and Pascal. I mentioned that Amsterdam is the most bike-friendly in the world, a fact I'd found on Wikipedia. You then asked me about my favorite editor; I initially thought you meant journal editor and began my response, but you quickly clarified that you were asking about code editors. We all had a good laugh at that. During our first meeting in Amsterdam, you asked me about my hometown. I told you I was from Huaibei, which drew a blank look from you. I added that it's a small city in Eastern China with only 2 million inhabitants. While in Amsterdam, the biggest city in the Netherlands, the population is less than 1 million. And Huaibei is not too far from Shanghai. However, in reality, Huaibei is around 600 kilometers from Shanghai, a distance greater than that between Amsterdam to Paris. I also appreciate your reassurances and contingency planning when the project funding for my PhD was canceled during my second year. Your action taught me the importance of always having a backup plan. I have learned so much from you, and for that, I am incredibly grateful.

Next, my sincerest thanks to my co-supervisor, Pascal. I appreciate your patience and kindness when I grappled with my procrastination during the paper submission process. I remember the time when we were submitting to BMVC'21. I was interning in Beijing and had not completed the draft until the last day of the deadline due to my procrastination. You did not blame me but instead dedicated your time and energy to help me polish the paper until the very last hour. Your encouragement was valid throughout my PhD journey.

I am indebted to the committee members for my PhD defense: Prof. dr. Paul Groth, Prof. dr. Marie Rosenkrantz Lindegaard, Prof. dr. Albert Salah, Prof. dr. Rita Cucchiara, and dr. Iris Groen. Your insightful comments and suggestions on my thesis were invaluable.

I would also like to extend my special thanks to my internship mentors, Tan Yu and Ping Li at Baidu Beijing, as well as Erhan Gundogdu and Loris Bazzani at Amazon Berlin. Your mentorship provided me with valuable industry insights and an enriching working experience.

I owe a debt of gratitude to my lab mates and friends in VIS Lab and the old ISIS group. This includes but is not limited to Zenglin Shi, Yunlu Chen, Sarah Ibrahimi, Pengwan Yang, Teng Long, Jia-Hong Huang, Yunhua Zhang, David Zhang, Tao Hu, Jiaojiao Zhao, Shuai Liao, Yingjun Du, Jiayi Shen, Zehao Xiao, Haochen Wang, Jie Liu, William Thong, Riaan Zoetmulder, Devanshu Arya, Gjorgji Strezoski, Sadaf Gulshad, Melika Ayoughi, Mehmet Altinkaya, Mert Kilickaya, Noureldien Hussein, Tom Runia, Berkay Kicanaoglu, Inske Groenen, Mina Ghadimi, Wenzhe Yin, Yongtuo Liu, Fida Thoker, Tom van Sonsbeek, Duy Kien Nguyen, and many others. The times we spent together—talking, drinking sharing meals, cycling, attending shows and concerts, barbecuing—have given me memories I will always cherish. And special thanks to Sarah for polishing my Dutch thesis summary.

Thanks to Dennis and Virginie. When I joined the group, I was advised to maintain a good relationship with you both. Your assistance was always invaluable.

To Xiantong Zhen, Efstratios Gavves, Yuki M. Asano, Marcel Worring and Nanne van Noord, thank you for your insightful feedback on my soos talk. A special thanks go to Arnold for his kindness to organize New Year's dinners for the whole group at his place.

I also want to express my gratitude to my friends in Amsterdam, including Shaojie Jiang, Jiayun Fan, Jiahuan Pei, Shihan Wang, Qi Wang, Shaodi You, Jingfei Xie, Ting Yin, Chang Li, Minghui Fan, Chongxuan Li, among others. The memories we created together are precious.

Lastly, my deepest appreciation goes to my family. To my parents, I would not have survived this PhD journey without your unwavering support and love. And to my grandmothers, I hope you maintain a good healthy body. I am also grateful to Wenjing Chu, who enriched the latter three years of my PhD journey. I feel fortunate to share this journey with you.

Sincerely,
Shuo Chen