Tragedy Plus Time: Capturing Human Unintended Activities from Weakly-Labeled Videos

by

Arnav Chakravarthy

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2021 by the
Graduate Supervisory Committee:

Yezhou Yang, Chair
Hasan Davulcu
Theodore Pavlic

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

In videos that contain actions performed unintentionally, agents do not achieve their desired goals. In such videos, it is challenging for computer vision systems to understand high-level concepts such as goal-directed behavior. On the other hand, from a very early age, humans are able to understand the relation between an agent and their ultimate goal even if the action gets disrupted or unintentional effects occur. Inculcating this ability in artificially intelligent agents would make them better social learners by not just learning from their own mistakes, *i.e.*, reinforcement learning, but also learning from other's mistakes. For example, this could greatly reduce the search space for artificially intelligent agents for finding the correct action sequence when trying to achieve a new goal, since they would be able to learn from others what not to do as well as how/when actions result in undesired outcomes. To validate this ability of deep learning models to perform this task, the **W**eakly Augmented **Oops** (W-Oops) dataset is proposed, built upon the Oops dataset (Epstein *et al.*, 2019). W-Oops consists of 2,100 unintentional human action videos, with 44 goal-directed and 33 unintentional video-level activity labels collected through human annotations. Inspired by previous methods on tasks such as weakly supervised action localization which show promise for achieving good localization results without ground truth segment annotations, this paper proposes a weakly supervised algorithm for localizing the goal-directed as well as the unintentional temporal region of a video using only video-level labels. In particular, an attention mechanism based strategy is employed that predicts the temporal regions which contributes the most to a classification task, leveraging solely video-level labels. Meanwhile, our designed overlap regularization allows the model to focus on distinct portions of the video for inferring the goal-directed and unintentional activity, while guaranteeing their temporal ordering. Extensive quantitative experiments verify the validity of our localization method.

DEDICATION

This is dedicated to my family, Monica Chakravarthy, Kumar Chakravarthy, Rohan Chakravarthy, Usha Bhatia and Suresh Bhatia who inspire and have supported me throughout my life, and without who none of this would have been possible

ACKNOWLEDGMENTS

I express my sincere gratitude to Dr Yezhou Yang, who gave me the chance to be part of the Active Perceptions Group (APG). His enthusiasm and passion has been an inspiration for me since I first approached him to be part of his research group. I have learnt how to communicate my ideas better, and turn a very rusty idea to the work I have presented in this thesis, due to his guidance. I would also like to thank Dr Hasan Davulcu and Dr Steven Corman, who gave me the opportunity to work on extremely interesting projects, and apply my Deep Learning knowledge to practical scenarios. It was an honor to work with them. I would also like to take this opportunity to thank Dr Theodore Pavlic for agreeing to be a presiding member of the thesis defense committee.

I would like to thank John Micco who was my mentor during my summer internship at VMware. It has been my best work experience so far, and working with someone of his experience and intellect is something I am extremely grateful to.

I have had the opportunity to work with some of the most intelligent people at Active Perceptions Group (APG) and Cognitive Information Processing Systems (CIPS) labs. A special thanks to Zhiyuan Fang, a colleague turned friend, for helping me and guiding me, and without whom this project would have not been possible. Another special thanks to Tejas Gokhale for the various illuminating conversations, as well as helping me with my paper submissions. I would also like to thank other members Aadhavan Sadasivam, Maryam Mousavi, Dylan Weber with whom I have the wonderful opportunity to interact and work with.

I am deeply grateful to Tarini Sawant, Bhavishya Pratap, Rishil Lala, Disha Khurana, Sarthake Choudhary and Vinit Acharya for the emotional support and helping lift my spirits. A special thanks to Dwiti Shah for supporting me throughout my Master's, never letting me feel demoralized and always inspiring me to do better.

Finally, I would like to thank my family for everything and anything good in my life, and for whom any amount of appreciation would not be enough.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

The word Teleology, coined by the German philosopher Christian Wolff, originates from two Greek terms 1).*telos*-which is a term used by philosopher Aristotle to refer to the full potential or inherent purpose or objective of a person or thing and 2). *logia* which refers to 'study of' or 'branch of learning'. Teleology is therefore a reason or explanation of something as a function of its end, purpose or goal as opposed to as a function of cause.

Providing a teleological explanation for human action hence involves viewing human action as goal-directed, *i.e.*, performing an action in order to achieve a goal. Previous psychology research (Csibra *et al.*, 1999; Csibra, 2003; Gergely *et al.*, 1995; Csibra, 2008; Brandone and Wellman, 2009; Gergely and Csibra, 2003) provides extensive evidence that children in their first year form this kind of teleological representations of actions. Viewing an action under a teleological lens offer atleast three advantages which are critical for coordinated social interaction since it helps social agents to evaluate others behaviours (Malle *et al.*, 2001). One such advantage includes predicting actions in situations where the environment changes but the goal does not, since understanding a goal would help predict how an agent would adjust its action to the modified environment (Csibra and Gergely, 2013). Another important advantage is that a novel action could be viewed as a means to achieve a goal state. This is especially useful when these novel actions provide a more efficient means to achieve a goal-state than the means action we formerly had access to (Csibra and

Gergely, 2007). And lastly, it provide humans the ability to predict and attribute a goal to an action even before the outcome is fully realized (the goal being partially fulfilled or not fulfilled at all) (Brandone and Wellman, 2009; Gergely and Csibra, 2003). One such experiment is also shown in Fig. 1.1.

Unfortunately, current computer vision models are not able to provide these advantages. Many of them focus on action recognition (Kalfaoglu *et al.*, 2020; Davoodikakhki and Yin, 2020; Wang *et al.*, 2016; Zhou *et al.*, 2018a; Lin *et al.*, 2019; Carreira and Zisserman, 2017; Tran *et al.*, 2018), which focuses on predicting the physical motion and atomic actions in a video. However, this captures only the means of the action and not the underlying goal, which requires a deeper penetration of what is happening in the video. For example viewing Fig. 1.2, we see that the agent is not able to fulfill his goal-directed action of shooting the ball. Us as humans are able to infer this goal-directed action due to our commonsense knowledge and teleological understanding of actions. It is however challenging for a deep learning model to be able to infer this, as 'shooting a ball' is not entailed on the surface appearance of the video. This would not be possible without penetrating deeper than the surface appearance of the action.

If we were to inculcate this abilities in artificially intelligent agents it would provide them with a new lens under which they can view and understand actions, making them better social learners. (Malle *et al.*, 2001) by not just learning from their own mistakes, i.e., reinforcement learning, but also learning from other's mistakes. For example, this could greatly reduce the search space for AI agents for finding the correct action sequence when trying to achieve a new goal, since they would be able to learn from others what not to do as well as how/when actions result in undesired outcomes.

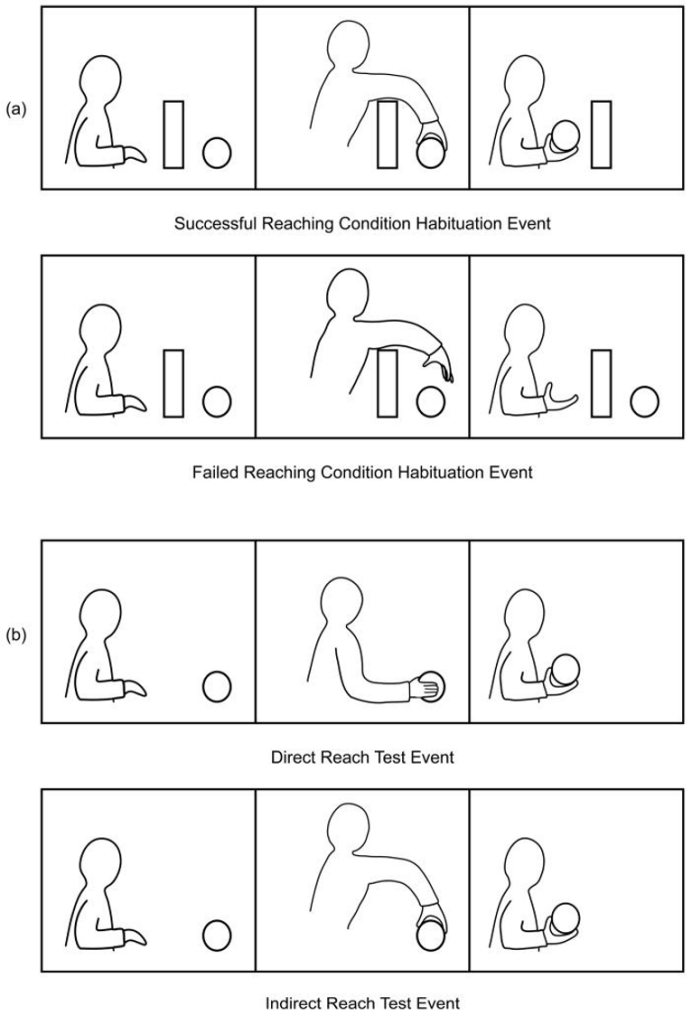

Figure 1.1: Experiment Performed in (Brandone and Wellman, 2009). Children Are Initially Habituation to One of the Two Events in (a), Which Involved a Successful Attempt of a Man Reaching for a Ball over a Barrier as Well as a Failed Attempt. They Are Then Exposed to Both the Test Events in (b), Where There Exists No Barrier. The First Test Event Involves a Man Directly Reaching for the Ball, and the Second Event Involves Him Taking an Arcing Motion to Reach for the Ball Which Is Not Consistent with the Goal of Reaching the Ball. Children Habituated to the Failed Action Look Longer at the Indirect Test Event Which Is Not Consistent with the Goal, Which Concludes That Infants Understand Goal-directed Behavior in Failed Actions.

Figure 1.2: In This Video, an Agent's Ultimate Goal Is to Help His Team Win the Game. In Order to Do This He Performs a Goal-directed Action of Running Towards the Basket to Shoot the Ball. However, He Slips and Falls, Not Being Able to Fulfill His Action.

We also know that in the real world, more severe penalties are assigned to intentional misdeeds rather than unintentional ones, such as intentional fouling in basketball and intentional grounding in football (Malle and Knobe, 1997). There do exist few works (Epstein *et al.*, 2019; Synakowski *et al.*, 2021) to discriminate between an intentional and unintentional action. However, in case of an unintentional misdeed only understanding that it is unintentional is not enough. AI agents should also be able to extract the goal of the action as well as the temporal region in which it occurs in order to make it's decision more justifiable as well as explainable, and as a result improve human's trust in artificially intelligent agents.

Due to these numerous advantages offered by being able to understand an action under the lens of teleology, and the challenges imposed due to understanding the goal-directedness of actions in situations especially where the goal is partially or completely not fulfilled, *i.e.*, unintentional actions, teleological action understanding of unintentional actions in computer vision becomes an interesting topic to delve into.

1.2 Motivation

Learning socially, by observing other agents in the same environment is crucial to discover new behaviors that would be difficult to obtain if explored individually (Laland, 2004; Henrich and McElreath, 2003). Watching other's mistakes and learning from them would enable artificially intelligent agents to not make those same mistakes when trying to achieve the same goal.

Another important practical scenario where teleological action understanding could be helpful is Law. People attempting to commit a crime but failing to do so can still be charged with it as long as they had the intent to do so. On the other hand, a person who commits a crime unintentionally is charged with innocent conduct rather than criminal conduct. This is known as Criminal Intent (Mens Rea), an element often used by the Supreme Court to distinguish charges made on a person (Men, 2011; Martens, 2018). However, since there is rarely any direct evidence of the defendant's intent the case is usually argued by the process of reasoning based on the specific scenario and environment in which the incident occurred. Here as well, it is important for the judge, jury as well as the defendant to attribute a goal to an action even if performed unintentionally. Hence, in today's world where we are evolving towards more human-like AI agents, inculcating these abilities in them is important as well.

There are few previous works which have taken initial steps towards teleological action understanding. (Epstein *et al.*, 2019) builds a dataset rich in unintentional human action, as well as single point transition times manually labeled by human annotators which helps separate the intentional and unintentional regions of the video. They also train models to classify an action as intentional or unintentional, as well localize points where an intentional action transitions into an unintentional action. However, it does not contain well defined classes for the goal-directed action or why

this goal gets disrupted. (Synakowski *et al.*, 2021) too focuses on predicting whether an activity was intentional or unintentional, but again do not focus on understanding the underlying goal of an unintentional action. (Fang *et al.*, 2020; Lei *et al.*, 2020; Zellers *et al.*, 2019) have tried to speculate about all possible effects and intents of actions, but do not focus on which effects are undesirable. We can see that though these works focus on detecting the intentionality of an action, or predicting the possible intents of an action, there is no work yet which focuses on finer-grained understanding of unintentional actions yet

Hence propose a novel dataset dubbed as **W**eakly Augmented **Oops** (W-Oops), which is built upon the original Oops (Epstein *et al.*, 2019) dataset, and contains unintentional action scenes as well high quality video-level annotations which describes the goal-directed action as well as the unintentional action occurring in the video. We further propose a weakly supervised framework which is able to infer the goal-directed and unintentional actions from the video as well as localize their respective temporal regions using only video-level action labels.

1.3 Challenges

For fine-grained understanding of unintentional actions, it is important to know 1) what is the goal of the action? 2) why was it not fulfilled? 3) when (in time) and what part of the video did the action start transitioning away from its goal?. This is a challenging task for deep learning models since it requires the model to understand high level concepts such as goal-directed behavior which is not directly visible on the surface appearance of the video. As an example, we can refer to Fig. 1.3 where an agent is trying to hit the ball but ends up falling and landing on his face. State of the art action recognition models such as an I3D (Carreira and Zisserman, 2017) trained on the Kinetics-600 dataset (Carreira *et al.*, 2018) view the whole scene as "*faceplant*"

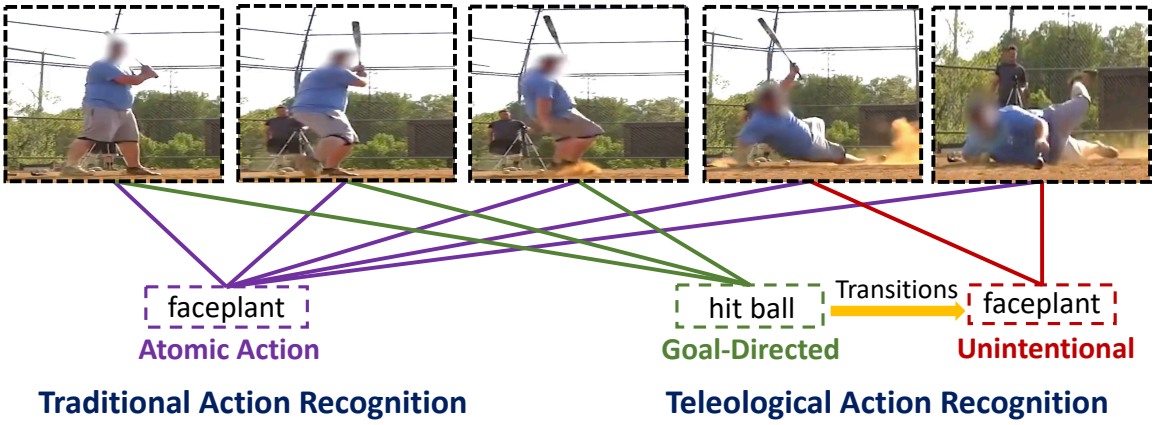

Figure 1.3: State-of-the-art Action Recognition Models Trained on Traditional Video Activity Datasets View an Unintentional Action Scene as an Atomic Action. Although This Scene Involves a Man Falling on His Face, the Man's Ultimate Goal Was to Hit the Ball. Green Lines Indicate the Regions of the Video Which Indicate the Man's Goal, Red Lines Indicate the Regions Where the Action Deviates from the Goal, and Purple Lines Indicate the Region the Action Recognition Model Focuses On.

without paying attention to the goal-directed behavior which was to "*hit the ball*".

One main reason for this challenge is that existing datasets (Kay *et al.*, 2017; Carreira *et al.*, 2018; Gu *et al.*, 2018a; Monfort *et al.*, 2019; Schuldt *et al.*, 2004; Blank *et al.*, 2005; Kuehne *et al.*, 2011; Simonyan and Zisserman, 2014; Wang *et al.*, 2014; Karpathy *et al.*, 2014; Caba Heilbron *et al.*, 2015; Abu-El-Haija *et al.*, 2016; Nguyen *et al.*, 2016; Sigurdsson *et al.*, 2016; Goyal *et al.*, 2017; Fouhey *et al.*, 2018; Gu *et al.*, 2018b) mostly focus on detecting only intentional actions where the goal of the agent is realized by the action performed, unlike unintentional actions. There does exist the Oops dataset (Epstein *et al.*, 2019) which contains transition point labels (point where an action starts deviating from its goal) and human level sentence annotations of the goal and the unintended actions in unintentional videos. However it contains redundant sentence representations to address the same goal or unintentional actions

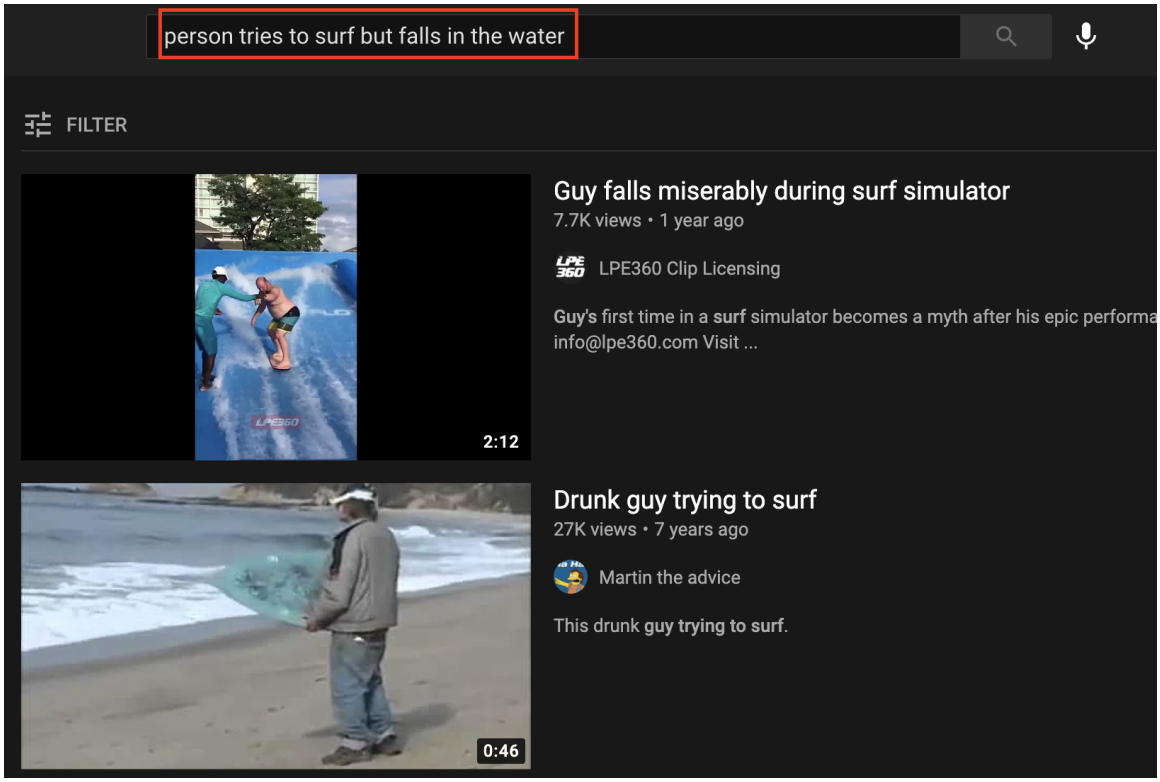

Figure 1.4: Image Showing the Top Two Results on Youtube When Querying "a Person Tries to Surf but Falls in the Water", Which Shows It Is Possible to Easily Obtain Specific Unintentional Videos, by Inputting the Goal-directed and Unintentional Action in a Template Sentence.

and many of these sentences are ambiguous due to the diverse vocabulary of the human annotators.

In order to build a model to be able to tackle these questions, a dataset containing well defined goals and why these goals get disrupted is crucial. Though the results might contain some noise we can also obtain specific unintentional videos on platforms like YouTube by querying "person trying to ____ but ends up ____"" or "person fails to ____"", and replace the blanks by the specific goal and unintended actions we wish to query (example shown in Fig. 1.4), and create a dataset from this. Additionally, in

order to localize these regions in time, one may manually label the transition point as in [9] and fully supervise the training. However, these annotations are prohibitively expensive to collect and suffer from human error and bias.

Previous works such as (Min and Corso, 2020; Shi *et al.*, 2020; Zhai *et al.*, 2020; Liu *et al.*, 2019; Paul *et al.*, 2018; Lee *et al.*, 2020; Shou *et al.*, 2018; Nguyen *et al.*, 2018), which focus on segmenting atomic action scenes from untrimmed videos in a weakly-supervised manner tackle this problem of expensive manual labelling by training a model in a weakly supervised manner using only video level action labels. Though this task differs from our task, as it involves segmenting an atomic action from an untrimmed video, whereas our task involves segmenting goal-directed and unintentional regions from a single unintentional action scene, it still provides encouragement to solve our task in a weakly supervised manner.

1.4 Contributions

In order to tackle the challenges addressed above, we bring W-Oops, an novel human activities dataset which contains "*fail*" videos, building upon Oops (Epstein *et al.*, 2019) but also contains high quality video-level annotations which describes the goal-directed as well as the unintentional actions in the video. To approach this challenge, we develop an algorithm which allows the model to attend to contextualized visual cues to localize the temporal regions of the goal-directed action, unintentional action as well as classify these actions in the video. Our weakly supervised framework includes an encoder to encode the joint representation of the goal-directed and unintentional action from the video as well as temporal attention modules which help the model focus on the respective regions of interest in the video. We also introduce a novel optimization target known as Overlap Regularization which allows the model to pay attention to distinct parts of the video for inferring both types of actions

while ensuring their temporal ordering. In addition we use Multiple Instance Learning Loss Zhou (2004) in order to end-to-end train the model for a classification task. Finally, we use the class-agnostic (bottom-up) as well as class-specific (top-down) attention mechanisms to localize both types of actions. We believe we are the first to make a step towards fine-grained understanding of unintentional actions.

Summarizing our contributions,

1. We curate W-Oops, a novel video dataset containing high quality video level labels for the goal-directed action as well as the unintentional action.

2. We propose a method, which incorporates attention mechanism to focus on relevant temporal regions of the video important to the classification task, while enforcing the model to pay attention to distinct parts of the video when inferring the goal-directed and unintentional action as well as ensuring the temporal ordering of these actions.

3. Finally, we provide in-depth and comprehensive experimental analysis, and it shows that our model achieves competitive results compared to several weakly supervised action localization models.

1.5 Outline

In **Chapter 2**, we discuss some relevant work which has been done in the domain of intent recognition and action localization.

Chapter 3 focuses on how we collected the W-Oops dataset. We further provide detailed statistics and analysis about the dataset.

In **Chapter 4** we discuss our weakly supervised framework for solving this challenge, and discuss all the modules used in the framework in detail. We also discuss about the loss functions used, especially the novel Overlap Regularization proposed

in this work. And finally we discuss the inference procedures.

Chapter 5 focuses on comprehensive and detailed experimental analysis which validates the effectiveness of our proposed approach. We also provide qualitative results, to compare our approach with previous competitive weakly supervised action localization methods.

Chapter 2

RELATED WORK

2.1 Intent Recognition in Computer Vision

There has been an increase in research focusing on intention recognition of agents in videos. (Fang and López, 2019) uses 2D pose estimation to predict the cyclist's intent of turning or stopping or a pedestrian's intent of crossing. (Rasouli *et al.*, 2020) uses a multi-task learning framework to simultaneously predict a pedestrian's trajectory, action and final location. (Varytimidis *et al.*, 2018) too focuses on pedestrain behavior estimates uses pedestrian's motion and head estimates to predict the behavior. Our work differs from this as 1) it focuses on predicting the past and not the future. 2) it is generalized to more diverse environmental settings.

(Wei *et al.*, 2018) proposes a hierarchical model and adopts a multi task learning approach to predict the intention, the attention of an agent's eye gaze, as well as the task being performed by an agent from a RGB-D video. (Vondrick *et al.*, 2016) focuses on predicting the action, motivation and scenes from an image by leveraging a commonsense third order factor graph built from text. (Synakowski *et al.*, 2021) discriminate between an intentional and unintentional action in performed by an agent in realistic videos, using an unsupervised algorithm built using common knowledge concepts of self-propelled motion, Newtonian motion and their relationship. (Epstein *et al.*, 2019) too focuses on discriminating between an intentional and unintentional action, as well as predicting the point in an unintentional video when the action deviates from it's original goal using a supervised algorithm. Our work differs from these as we focus on discriminating between the different goal-directed and unintentional

action categories in unintentional videos, as well as localizing these action regions in a weakly supervised manner. Action anticipation can also be relevant to predicting an unintentional action or the onset of it. (Furnari and Farinella, 2019; Sadegh Aliakbarian *et al.*, 2017; Miech *et al.*, 2019; Ryoo, 2011; Hoai and De la Torre, 2012) focus on forecasting an event or action based on a small snippet of a video. (Vondrick *et al.*, 2015; Tran *et al.*, 2019) focus on self supervised learning approaches to predict future action representation using unlabeled videos.

2.2 Action Localization

Action localization, unlike action recognition involves segmenting time intervals in untrimmed videos across the spatio-temporal axis which have a high probability of containing an action. There exist fully supervised as well as weakly supervised methods to address this problem. Fully supervised methods contain the ground truth time intervals for actions during training. Works such as (Zhao *et al.*, 2017; Soomro *et al.*, 2015; Shou *et al.*, 2016; Gkioxari and Malik, 2015; Yeung *et al.*, 2016) focus on fully supervised methods for action localization.

However, these time intervals are prohibitively expensive to collect and hence this data collection cannot scale due to human cost. In order to solve this problem, researchers started focusing towards weakly supervised methods which involve only action level labels during training. There also exists a third method which considers a special case where the availability of temporal ordering of actions during training. Papers such as (Bojanowski *et al.*, 2014, 2015; Huang *et al.*, 2016; Richard *et al.*, 2017; Kuehne *et al.*, 2017) localize action regions using information containing the temporal ordering of activities.

Weakly supervised Action Localization (WSAL) involves localizing action regions in an untrimmed video by training a model using only video level action labels,

without considering any temporal ordering of activities. STPN (Nguyen *et al.*, 2018) trains a classification model using features weighted by a class-agnostic attention weights, which it learns using a sparsity loss on the attention weights. It then performs the localization by using both the classification activation as well as these class-agnostic weights and threshold them to select action locations. WTALC (Paul *et al.*, 2018) forces the foreground action features from the same action class to be similar and the background features pertaining to an action class to be dissimilar from its foreground feature, and finally localizes the action by threshold the classification activation. A2CL-PT (Min and Corso, 2020) uses foreground and background features to form triplets and apply the Angular Triplet Center Loss (Li *et al.*, 2019) to separate the foreground and background features, as well as use an adversarial branch in order to find supplementary activities from non-localized parts of the video. DGAM (Shi *et al.*, 2020) propose to separate action frames from context frames by modeling the frame representation conditioned on the bottom-up attention. TSCN (Zhai *et al.*, 2020) fuse the attention sequences from the RGB and optical flow stream and use it as pseudo ground truth to supervise the training.

Our work differs from the above mentioned efforts as we do not focus on extracting atomic activity sequences from untrimmed videos, but rather focus on extracting goal-directed and unintentional regions from unintentional action sequences. This, as far as we know, is the first attempt to identify both goal-directed and unexpected elements in unintentional human activity videos.

Chapter 3

W-OOPS DATASET

In this section we talk about how we collected W-Oops, the annotation tool used to collect it, as well as discuss important statistics of this dataset.

3.1 Collecting the Dataset

The original Oops Dataset (Epstein *et al.*, 2019) consists of 20,338 videos containing human unintentional actions obtained by collating "fail" videos from different users on Youtube. Amazon Mechanical Turk workers are then asked to label the time at which the video starts transitioning from the goal-directed action to the unintentional action, as well as indicate whether a video does not indicate an unintentional action.

In order to create our dataset, which is built upon the labeled portion of the Oops dataset, we follow a similar pre-processing step as in (Epstein *et al.*, 2019) by 1). Removing videos that do not contain an unintentional action 2). Removing videos more than 30 seconds which are likely to contain multiple scenes, as well as removing those less than 3 seconds which are not likely to contain one full scene 3). Removing those videos where the transition time occurs in the initial 1% or ending 1% of the video, since there would not be enough context to understand the goal-directed action or unintentional action respectively. Post this process, we were left with a total of about 7,800 labeled videos.

Oops Dataset (Epstein *et al.*, 2019) also provide the annotations in natural language descriptions, which were obtained by asking Amazon Mechanical Turkers to watch the video and answer: "*what was the goal?*" and "*what went wrong?*". Since

we want to collect a distinct set of goal-directed and unintentional actions, we followed a technique similar to the Epic Kitchens Dataset (Damen *et al.*, 2018), by extracting the verbs and associated noun using the SpaCy dependency parser and concatenating them to form an action. We replace all compound nouns by it's second noun: *e.g.*, "*ride mountain bike*" is replaced with "*ride bike*" and so on. Due to the diversity of the worker's vocabulary, we find that the resulting action are of low quality, with many of them having ambiguous meanings such as "*fly bike*" as well as many of them having redundant meanings. In order to overcome this, we manually go over each of these extracted action and remove those with ambiguous meanings as well as merge the redundant ones, *i.e.*, "*jump over fence*" and "*jump over chair*" into a more general "*jump over obstacle*" category.

We finally carry out a human evaluation, going through all the videos manually and ensuring the correctness of the labels, and correcting them if need be. We also give the evaluator an option to discard the video if the goal of the actor was ambiguous. We build an annotation tool in order to make this process easier. The Appendix can be referred to for more information about this tool. We keep a minimum threshold of 15 for the number of videos per goal-directed action as well as unintentional class, and discard the rest of the classes, as well as the videos which were associated with these classes.

3.2 Statistics and Analysis

The final W-oops dataset contains 1,582 train samples and 526 testing samples, containing a total of 44 diverse goal-directed and 30 unintentional action classes, which can be seen in Fig. 3.1. We have also provided the distribution of the goal-directed and unintentional segment lengths, as well as the total video lengths in Fig. 3.2. It shows that the goal-directed and unintentional segment lengths are well

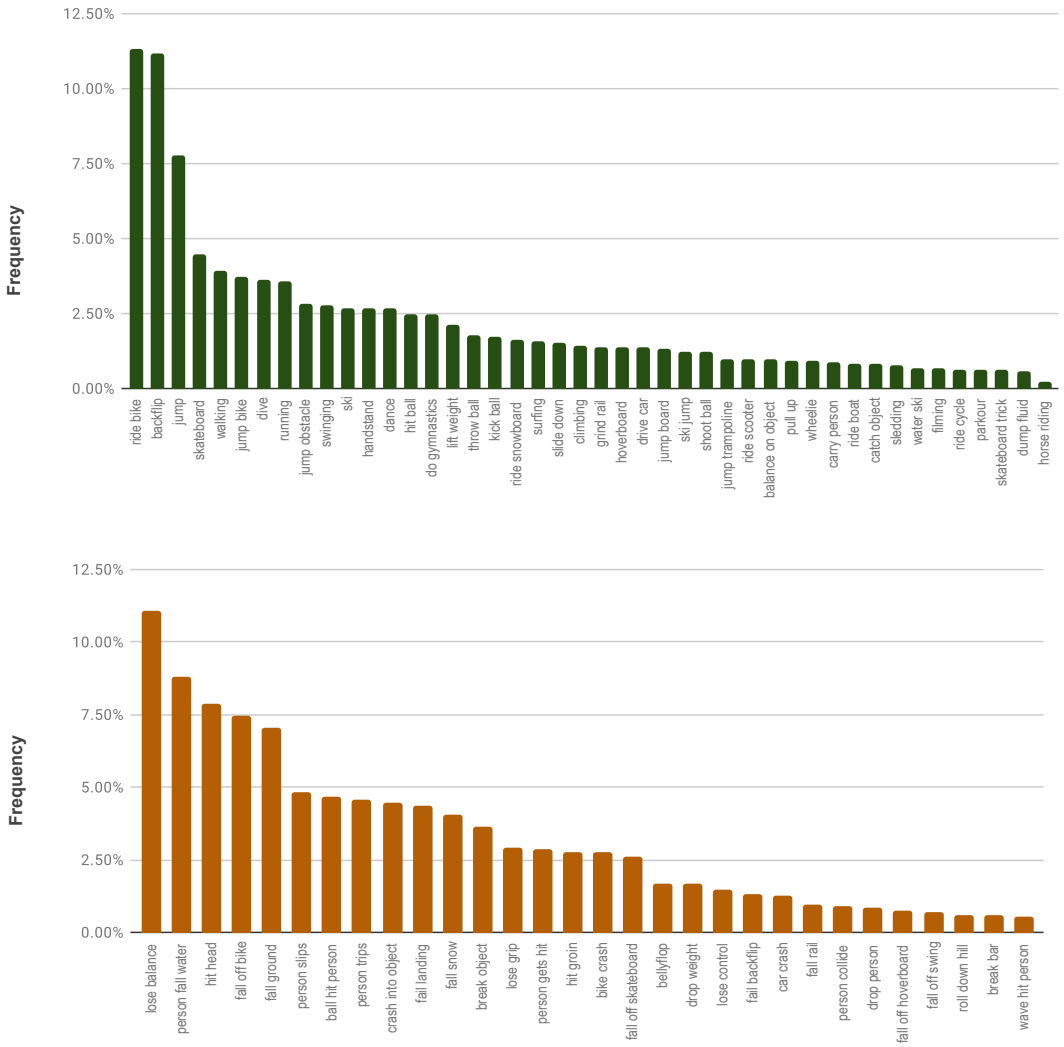

Figure 3.1: Distributions of the Goal-directed and Unintentional Actions Present in Our Dataset.

diversified over then entire length of the video. The lengths of the video are short in general, with a majority of them ranging from 6.2 - 7.7 seconds. This makes the task of identifying these sub-regions in the video challenging. In our benchmark, train samples contain only video-level labels whereas the test samples contain both the video-level labels as well as the unintended activity transition points (taken from the

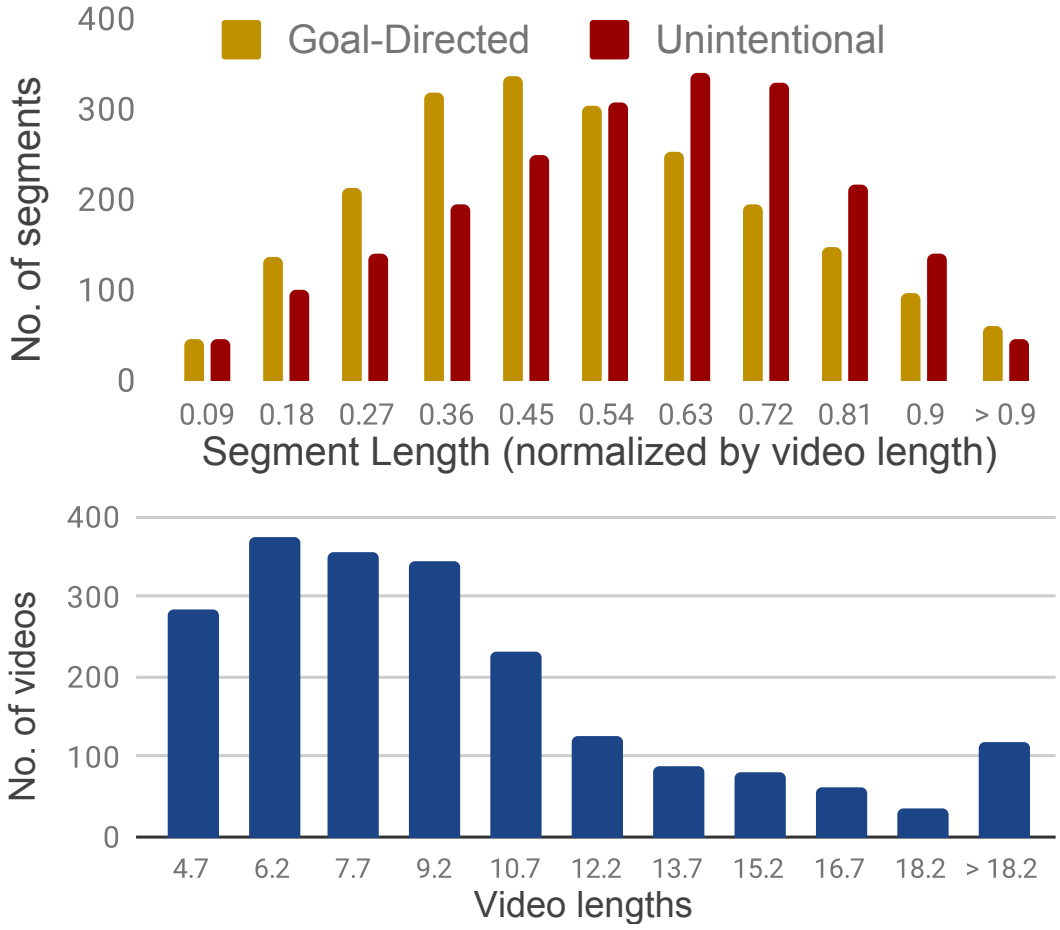

Figure 3.2: (Top) Distribution over Goal-directed and Unintentional Segment Lengths (Normalized by the Video Length). (Bottom) Distribution over the Entire Video Length.

original Oops dataset), in order to conduct evaluation.

Additionally, it is also interesting to know how much information does knowing about a goal-directed action give us when inferring the unintentional action. In order to analyse this, we calculate a probability distribution of the unintentional actions conditioned on the goal-directed actions and calculate their entropy. An entropy of 0 would indicate that the unintentional action can predicted from the goal-directed action alone. On the other hand, an entropy of $4.91(-\log_2(30))$ indicates that the

unintentional actions are uncorrelated with the goal-directed action. Fig. 3.3 shows us that the conditional entropy of unintentional actions lies between these two values, suggesting that they are correlated but are not completely predictable knowing the goal-directed action.

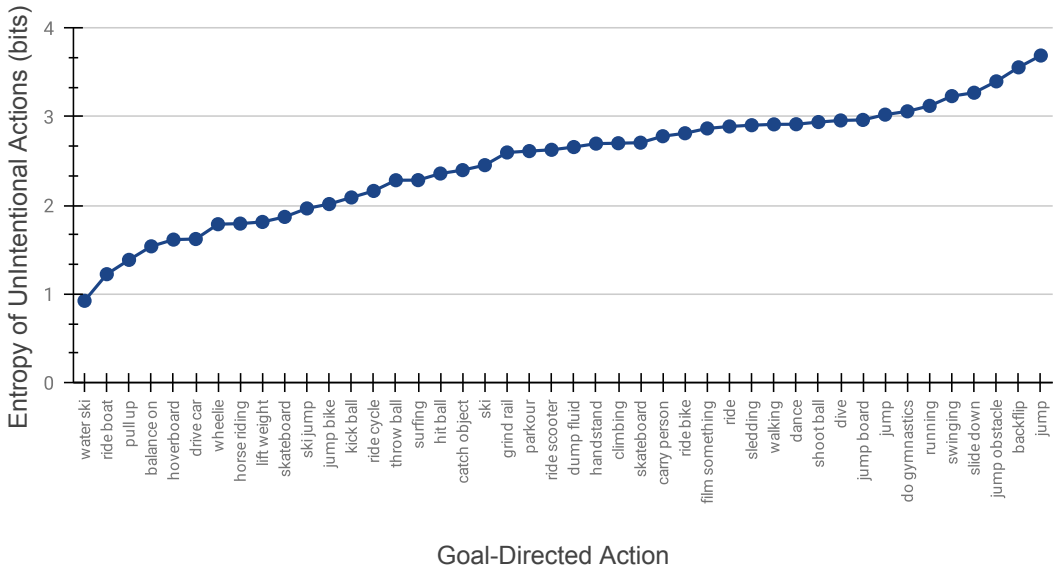

Figure 3.3: Entropy (in Bits) of the Unintentional Actions Conditioned on the Goal-directed Actions. We Can See That the Unintentional Actions Are Correlated to the Goal-directed Actions but Are Not Completely Predictable.

Chapter 4

OUR APPROACH

4.1 Proposed Architecture

We intend to identify the goal-directed and unintended human activities, as well as their corresponding moment of occurrence from an unintentional video in a weakly-supervised manner. To be specific, given the video $\mathcal{V}$ and its categorical labels representing the goal-directed activity, y^{IA}, and the unintended activity, y^{UA}, we expect the model to predict the triplets $\langle s^{\mathrm{IA}}, e^{\mathrm{IA}}, c^{\mathrm{IA}} \rangle$ and $\langle s^{\mathrm{UA}}, e^{\mathrm{UA}}, c^{\mathrm{UA}} \rangle$, containing the starting point, end point and action class associated with this segment by leveraging only the video-level annotations as weak supervision. We formulate this challenge as a weakly supervised action localization (WSAL) task, and address it using an attention mechanism based approach. We start this section by providing an overview of our model, followed by the details of formulations and our proposed learning objective.

To encode the videos, pre-trained 3D neural networks are exploited to extract a set of clip-level representations $\mathbf{X}$. We find that in order to encode the goal-directed and unintentional features from the video, directly using static features is not sufficient. Hence, we encode the clip embedding by an encoder network $\mathcal{F}$, which outputs a joint representation for the goal-directed and unintentional action:

$$\mathbf{O} = \mathcal{F}(\mathbf{X}), \tag{4.1}$$

where $\mathbf{O} \in \mathbb{R}^{l \times d}$ denotes the representations in d dimensions for l clips. Here, encoder network $\mathcal{F}$ can either be a bidirectional Gated Recurrent Unit or a Transformer Encoder (Vaswani *et al.*, 2017). On this basis, we introduce two bottom-up attention modules, which outputs the temporal attention weights that reflect the temporal

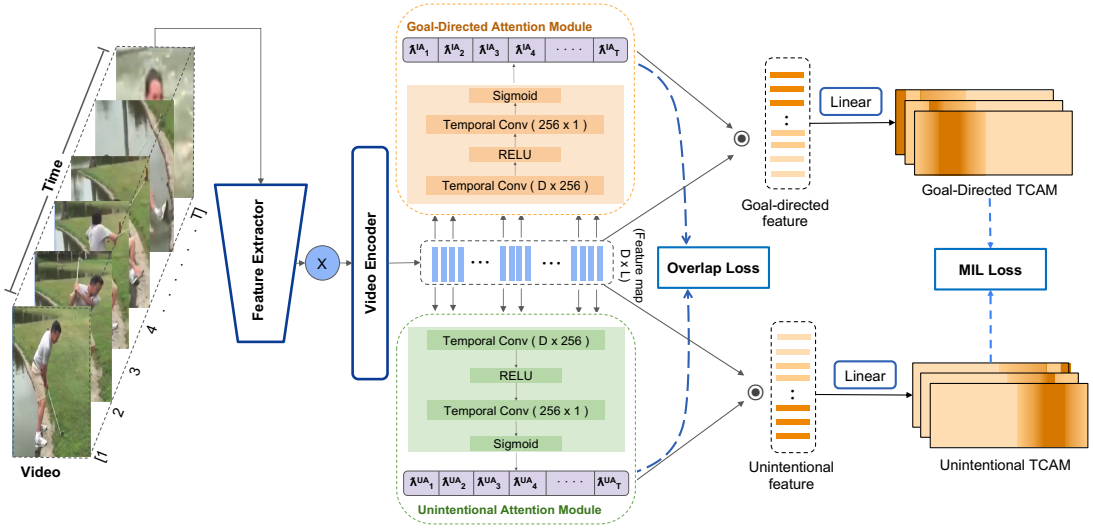

Figure 4.1: Illustration of Our Overall Architecture. A Backbone Feature Extractor Is Used to Convert Raw Videos into Features, *i.e.*, $\mathbf{X}$ and Is Kept Frozen Throughout the Training Process. $\mathbf{X}$ Is Then Passed to a Video Encoder Which Can Be Either a GRU (Chung *et al.*, 2014) or a Transformer Encoder (Vaswani *et al.*, 2017), to Extract High Level Features $\mathbf{O}$. The Two Attention Modules Use $\mathbf{O}$ to Predict the Bottom-up Attention Weights λ^{IA} and λ^{UA} for the Goal-directed and Unintentional Action Respectively, Which Are Used for the Overlap Regularization. We Calculate the Goal-directed, *i.e.*, $\mathbf{O}^{\text{IA}}$ and Unintentional Feature, *i.e.*, $\mathbf{O}^{\text{UA}}$ by Computing a Dot Product Between $\mathbf{O}$ and Their Respective Bottom-up Attention Weights. Finally We Pass the Goal-directed and Unintentional Feature Through Weight-shared Linear Layers to Extract Their Respective TCAMs $\mathbf{C}^{\text{IA}}$ and $\mathbf{C}^{\text{UA}}$. These TCAMs Are Used for the MIL Loss.

importance of clip representations for the goal-directed/unintentional activity respectively. This is achieved by training the model with a classification loss, *e.g.*, multiple instance learning loss. Note that these attention weights are agnostic to the specific action, and are used to identify generic regions of interest. A stack of 1-D Convolution

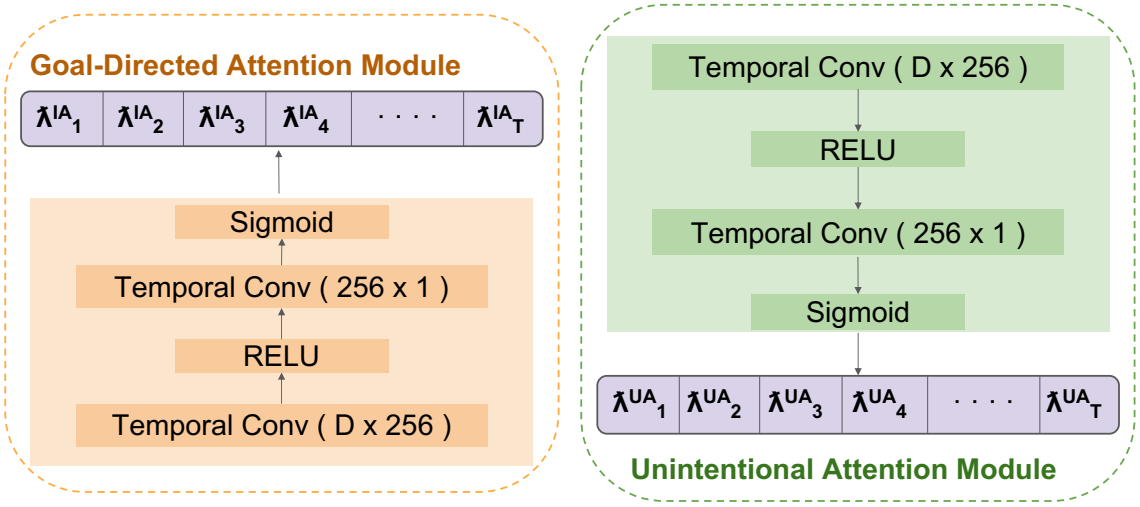

Figure 4.2: Class-agnostic Attention Modules for Capturing the Temporal Attention Weights for the Goal-directed Region (Left) and Unintentional Region (Right).

layers with RELU activation between layers, followed by a Sigmoid function is used to obtain the attention weights λ^{IA}, $\lambda^{\text{UA}} \in \mathbb{R}^l$ with a scale between 0 and 1.

In order to obtain goal-directed and unintentional features, we compute a dot product between the joint representation $\mathbf{O}$ and each of the bottom-up attention weights λ^{IA} and λ^{UA}. These features would represent those parts of the joint representation $\mathbf{O}$ which correspond to the goal-directed and unintentional region respectively. Formally,

$$\mathbf{O}^{\text{IA}} = \mathbf{O} \cdot \lambda^{\text{IA}}, \qquad \mathbf{O}^{\text{UA}} = \mathbf{O} \cdot \lambda^{\text{UA}}. \tag{4.2}$$

We then compute Temporal Class Activation Maps (TCAM) (Nguyen *et al.*, 2018), $\mathbf{C}^{\text{IA}} \in \mathbb{R}^{l \times N_{\text{IA}}}$, $\mathbf{C}^{\text{UA}} \in \mathbb{R}^{l \times N_{\text{UA}}}$ for the goal-directed as well as unintentional actions, with N_{IA} and N_{UA} corresponding to the number of goal-directed and unintentional classes, by employing two weight-sharing linear transformation layer on $\mathbf{O}^{\text{IA}}$ and $\mathbf{O}^{\text{UA}}$ respectively. These are one dimensional class-specific activations that signify classification scores over time for both the types of actions for each segment (as illustrated

in Fig. 4.1). These class-specific distributions, along with the class-agnostic distributions are used to predict the triplets $\langle s^{\text{IA}}, e^{\text{IA}}, c^{\text{IA}} \rangle$ and $\langle s^{\text{UA}}, e^{\text{UA}}, c^{\text{UA}} \rangle$ associated with the goal-directed and unintentional activities respectively.

4.1.1 Feature Extraction

4.1.1.1 3D-CNN Architectures

We extract RGB features by creating chunks of 16 consecutive and non-overlapping frames and using the I3D (Carreira and Zisserman, 2017) as well as R(2+1)D (Tran *et al.*, 2018) pretrained architectures to extract clip-level features from these chunks. We follow previous work (Epstein *et al.*, 2019) and down-sample all raw videos at 25 FPS. We then create chunks of 16 consecutive and non-overlapping frames. In order the extract the I3D and R(2+1)D features, we pass these chunks to the respective backbone networks and obtain the features as the output of their global pooling layers. We use the following libraries to extract R(2+1)D[1] and I3D[2] features from the videos.

I3D: For the I3D (Carreira and Zisserman, 2017) features, we re-scale all frame pixels between -1 and 1, after which we resize the frames preserving aspect ratio such that the smallest dimension is 256 pixels. We then apply center crop to obtain 224×224 frames. Chunks of 16 non-overlapping frames are then passed through the RGB stream of a I3D (Carreira and Zisserman, 2017) backbone pretrained on the Kinetics dataset (Kay *et al.*, 2017) to obtain features $\mathbf{X_i} \in \mathbb{R}^{1024 \times l_i}$ from the global pooling layer.

R(2+1)D: For the R(2+1)D (Tran *et al.*, 2018) network, we re-scale frame pixels between 0 and 1, after which we resize all frames to 128×171. We then normalize these frames and finally apply center crop to obtain 112×112 frames. We the chunk

[1] `https://pytorch.org/vision/0.8/models.html`

[2] `https://github.com/deepmind/kinetics-i3d`

the frames in the same way and pass it through the R(2+1)D (Tran *et al.*, 2018) backbone pretrained on Kinetics to obtain features $\mathbf{X_i} \in \mathbb{R}^{512 \times l_i}$ from the global pooling layer.

4.1.1.2 Human Skeleton Extraction and Vectorization

Successful attempts at using human skeleton features for activity recognition.(Luvizon *et al.*, 2018; Wang *et al.*, 2013; Yan *et al.*, 2018), fall prediction (Solbach and Tsotsos, 2017; Hua *et al.*, 2019) and action localization (Miki *et al.*, 2020) provides encouragement to use them for our task as well. However human skeleton features alone would not be enough as it does not capture the surrounding environment information which the RGB features do. Hence we concatenate both the RGB features and skeleton features to use as our backbone features.

In order to test this hypothesis, for each video we extract 2D keypoint coordinates of human(s) from each observed frame using OpenPose (Cao *et al.*, 2019). Since OpenPose is able to capture multiple human(s) in a frame, we use DeepSort (Wojke *et al.*, 2017) to cluster the keypoints of the same person across frames, as shown in Fig. 4.4. We denote the sequence of observed keypoints from the i^{th} person in the video as $\mathbf{K^i} = (\mathbf{k}_1^i, \mathbf{k}_2^i, ..., \mathbf{k}_t^i)$, where $\mathbf{k}_j^i$ denotes the keypoint coordinates of the i^{th} person in frame j, with t being the total number of frames.

Using the COCO model of OpenPose, we obtain 18 keypoint coordinates for each observed person in a frame, which include coordinates for the nose, neck, left and right shoulders, hips, elbows, wrists, knees, ankles, eyes and ears. More formally, each $\mathbf{k}_j^i = (x_{j,1}^i, y_{j,1}^i, x_{j,2}^i, y_{j,2}^i, ..., x_{j,18}^i, y_{j,18}^i)$. Since these coordinates do not capture the correlation between different keypoints, we follow the process in (Hua *et al.*, 2019) to vectorize these coordinates to incorporate these correlations. We ignore the face keypoints (eyes, ears and nose), since we want to focus only on the body pose.

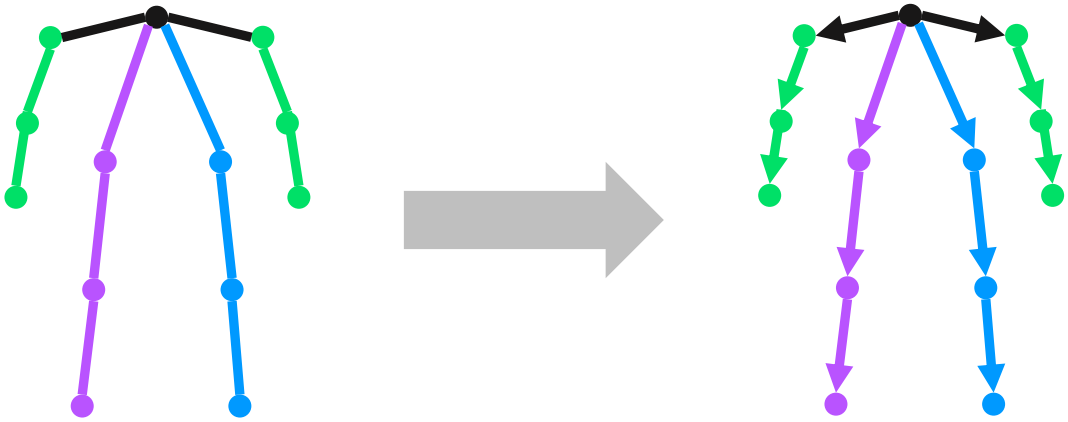

Figure 4.3: An Illustration of the Keypoints Vectorization Method Proposed in (Hua *et al.*, 2019). The Arrows Indicate the 12 Vectors Constructed from the 13 Keypoint Coordinates of the Human Skeleton Extracted from Openpose (Cao *et al.*, 2019). The Vectors Are Normalized to Unit Length While Preserving the Direction Information Which Corresponds to the Correlation Between the Different Body Keypoints.

We then transform the remaining 13 coordinates into vectors connecting the adjacent keypoints as illustrated in Fig. 4.3. The shoulders are connected to the neck, elbows are connected to the corresponding shoulders, wrists are connected to corresponding elbows, hips to the neck, knees to the corresponding hips and finally the ankles to the corresponding knees. Following this process as followed in (Hua *et al.*, 2019), we obtain 12 keypoint vectors from the 13 keypoint coordinates, and normalize them to unit length. For the m^{th} connection pointing from the p^{th} keypoint to the q^{th} keypoint, the keypoint vector $(\overline{x^i_{j,m}}, \overline{y^i_{j,m}})$ for the i^{th} person in frame j is calculated as:

$$(\overline{x^i_{j,m}}, \overline{y^i_{j,m}}) = \frac{(x^i_{j,q} - x^i_{j,p}, y^i_{j,q} - y^i_{j,p})}{\sqrt{(x^i_{j,q} - x^i_{j,p})^2 + (y^i_{j,q} - y^i_{j,p})^2}} \tag{4.3}$$

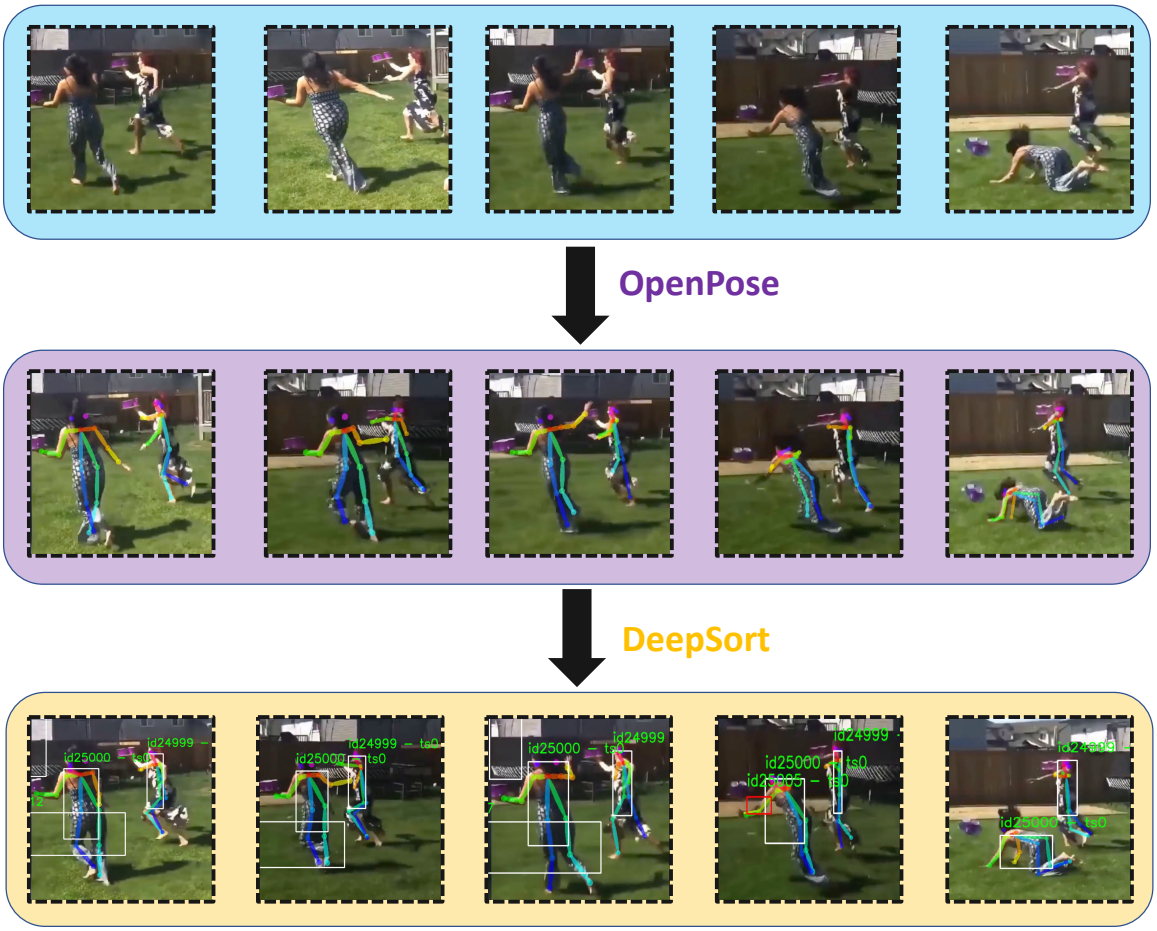

Figure 4.4: An Example of Extracting Body Keypoint Coordinates of Multiple Agents in Videos Using Openpose (Cao *et al.*, 2019), Followed by Deepsort (Wojke *et al.*, 2017) to Cluster the Keypoints of the Same Person Across the Frames.

We calculate this for each of the 12 connections, and concatenate them to get:

$$\overline{\mathbf{k}_j^i} = (\overline{x_{j,1}^i}, \overline{y_{j,1}^i}, \overline{x_{j,2}^i}, \overline{y_{j,2}^i}, ..., \overline{x_{j,12}^i}, \overline{y_{j,12}^i}) \tag{4.4}$$

Videos involving action such as two people colliding with another person, or a person carrying another person, requires features of multiple people in order to understand these actions. Hence we concatenate the keypoints of the two most frequently occurring people l and r as detected by DeepSort, and concatenate them to get the final

feature vector for frame j as $\overline{\mathbf{k}_j} = \overline{\mathbf{k}_j^l} \oplus \overline{\mathbf{k}_j^r}$.

Note, that there may be partially missing or completely missing keypoint coordinates for a person in a certain frame. In the case of partially missing keypoints we set a keypoint vector containing a connection to a missing keypoint to (0,0). In the case of completely missing keypoints we set all the keypoint vectors to (0,0) in the case the person had not been detected yet, or else set all the keypoint vectors to the corresponding last observed keypoint vectors of the person.

RGB features are extracted by passing non-overlapping chunks of 16 frames to a pretrained 3D CNN architecture. Since the skeleton feature are extracted for each frame, we concatenate skeleton features extracted from consecutive and non-overlapping chunks of 16 frames. We convert $\overline{\mathbf{k}} = (\overline{\mathbf{k}_1}, \overline{\mathbf{k}_2}, ..., \overline{\mathbf{k}_t})$ to $\widetilde{\mathbf{k}} = (\widetilde{\mathbf{k}_1}, \widetilde{\mathbf{k}_2}, ..., \widetilde{\mathbf{k}_{t/16}})$, where $\widetilde{\mathbf{k}_h}$ for the h^{th} chunk is given by :

$$\widetilde{\mathbf{k}_h} = \overline{\mathbf{k}_{16(h-1)+1}} \oplus \overline{\mathbf{k}_{16(h-1)+2}} \oplus ... \oplus \overline{\mathbf{k}_{16(h)}} \tag{4.5}$$

We finally concatenate the RGB features X and the skeleton features $\widetilde{\mathbf{k}}$ to obtain $X_{cat} = (X_1 \oplus \widetilde{\mathbf{k}_1}, X_2 \oplus \widetilde{\mathbf{k}_2}), ..., X_l \oplus \widetilde{\mathbf{k}_l})$, where l is the total number of 16 frame chunks (clips) in the video.

4.1.2 Video Embedding Module

4.1.2.1 Bidirectional Gated Recurrent Unit

In order to learn a joint representation for inferring the goal-directed and unintentional actions, we use a Bidirectional Gated Recurrent Unit (Chung *et al.*, 2014) as the video encoder. 3D-CNN architectures like R(2+1)D (Tran *et al.*, 2018) and I3D (Carreira and Zisserman, 2017) capture very short clip level information. However, capturing information which helps discriminate between the goal-directed region and an unintentional region requires longer temporal context which can be modeled

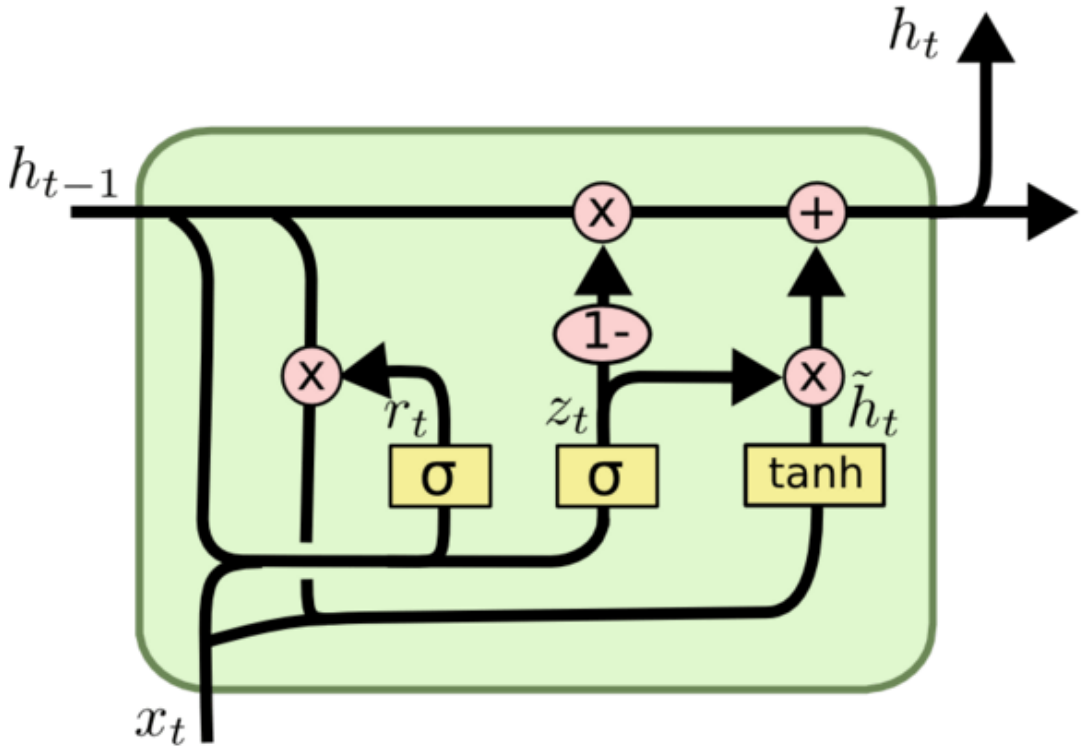

Figure 4.5: Example of a Gated Recurrent Unit. x_t Is Our Input at Timestep t.

by a GRU. Specifically, our GRU, shown in Fig. 4.5, consists of a reset gate r which controls how much importance to give the previous hidden state h^{t-1} in order to calculate the current hidden state h^t, and an update gate u which determines how much of the previous hidden state h^{t-1} should be carried on to the current hidden state h^t. Given the backbone feature $\mathbf{X}$, we compute the hidden state at each time-step t using the following equations:

$$\begin{aligned} z^t &= \sigma(\mathbf{W}^z\mathbf{X}^t + \mathbf{U}^z h^{t-1)}) && \text{Update Gate} \\ r^t &= \sigma(\mathbf{W}^r\mathbf{X}^t + \mathbf{U}^r h^{t-1}) && \text{Reset Gate} \\ \tilde{h}^t &= \tanh(r^t \cdot \mathbf{U}h^{t-1} + \mathbf{W}\mathbf{X}^t) && \text{New Memory} \\ h^t &= (1 - z^t) \cdot \tilde{h}^t + z^t \cdot h^{t-1}, && \text{Hidden State} \end{aligned} \tag{4.6}$$

where **U** and **W** correspond to learnable parameters of this module. In order to capture the forward information flow $\overrightarrow{h^{(t)}}$ as well as backward information flow $\overleftarrow{h^{(t)}}$ we use a Bidirectional-GRU and obtain the final representation **O** by concatenating these features from the final hidden layer.

4.1.2.2 Transformer Encoder

As opposed to a GRU which learns feature representations at each time step in a sequential manner by using the hidden state in the previous timestep, a transformer encoder, shown in Fig. 4.6, uses multiheaded self attention to calculate the dependency of each token in the sequence to encode the token at the current timestep. We provide a brief explanation of self-attention and multihead self-attention.

The self-attention module of a transformer consists of mapping a set of Queries (Q), Values (V) and Keys (K) to an output. In the context of self-attention the Queries, Values and Keys are the same and correspond, in our cause it corresponds to the output of our feature extractor, *i.e.* **O**. We calculate the output using the following equation:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.7}$$

where d_k is the dimensionality of the query, key and values.

Multi-head self attention allows the transformer encoder to learn different representation sub-spaces by computing the scaled dot product multiple times in parallel over the input. This is done in practice by separating the input into N_{heads} heads along its dimensionality axis. Finally we concatenate these outputs from each of these heads, and compute a dot product with a learnable weight matrix. More formally,

$$\begin{aligned} MultiHead(Q, K, V) &= (head_1 \oplus head_2 .... \oplus head_{N_{\text{heads}}}) \cdot W^o \\ where \qquad head_i &= Attention(Q_i W_i^Q, K_i W_i^K, V_i W_i^V) \end{aligned} \tag{4.8}$$

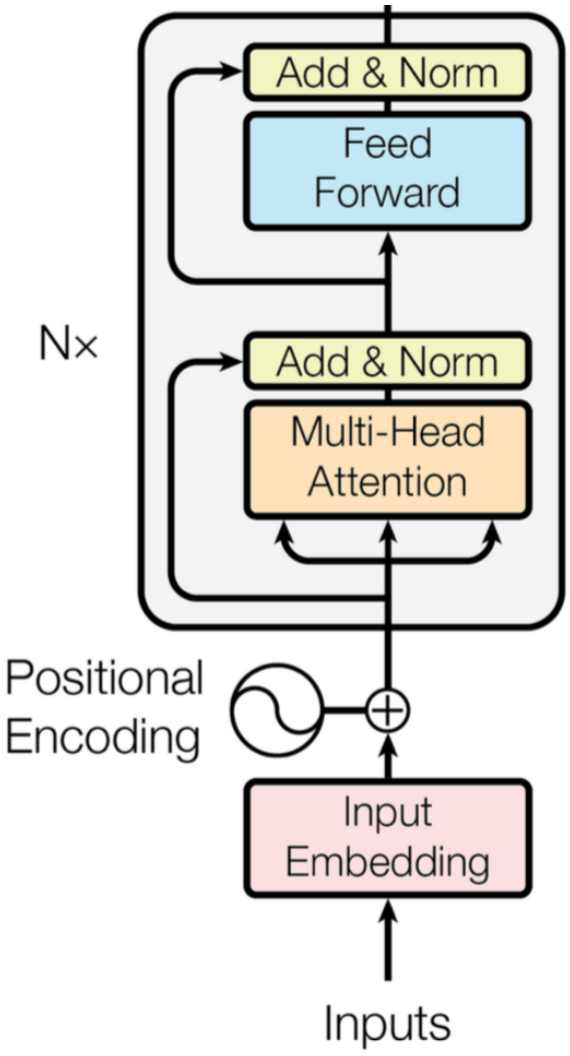

Figure 4.6: Architecture of a Transformer Encoder (Vaswani *et al.*, 2017)

The module diagram of scaled-dot product attention and multi-head attention can be viewed in Fig. 4.7

Since this architecture does not naturally incorporate the sequence of the data such as a GRU, we need to incorporate the sequence embeddings along with the

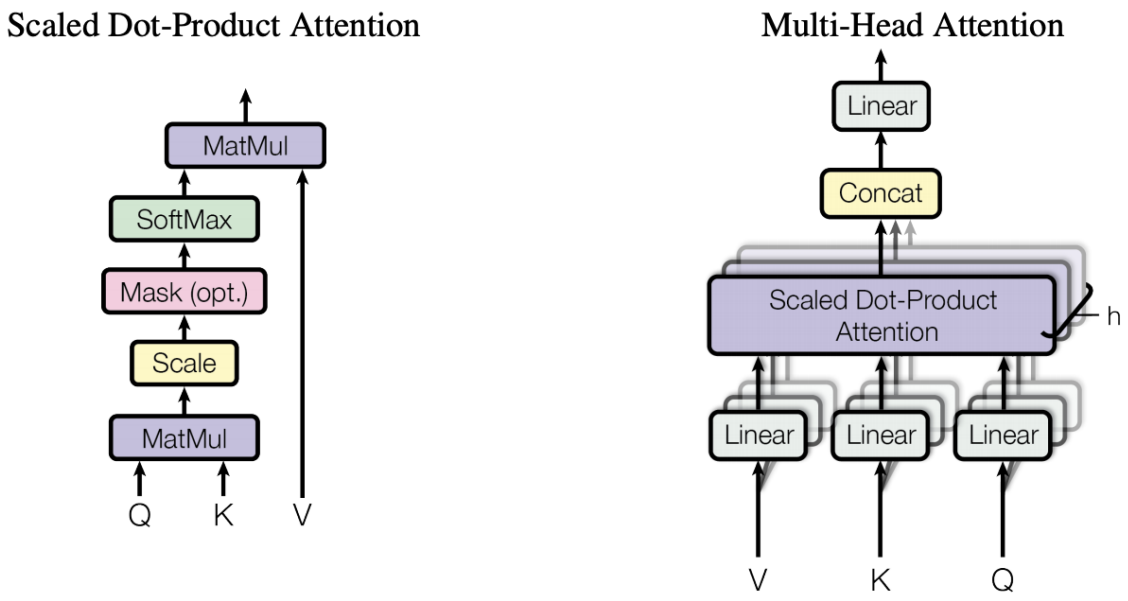

Figure 4.7: (Left) Scaled-dot Product Attention. (Right) Multi-head Attention

embeddings from our feature extractor. We can do is using the following formula:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}}) \tag{4.9}$$

where *pos* corresponds to the postion of the token, and i corresponds to the dimension. d_{model} corresponds to the dimension of the input and output.

4.1.3 Temporal Class Activation Maps (Class-Specific)

(Zhou *et al.*, 2016), which used Class Action Maps (CAM) produced using the global average pooling layer (GAP) in convolutional neural networks (CNN). The Class Activation Maps signifies the regions in an image which the convolutional neural network pays attention to when classifying the image into one of the classes. This is often used in weakly supervised object localization where a CNN is trained for a classification task, and during inference, a class activation map is generated and used for localizing the object in the image.

Taking inspiration from this concept (Nguyen *et al.*, 2018), extends this to videos by generating a Temporal Class Activation Map (TCAM), which signifies the regions along the temporal axis which are the most activated when classifying a video into a specific class. TCAMs are often used in weakly supervised action localization, where a model is trained for a classification task, such as detecting the activities present in videos, and then generating a temporal class activation map to localize these activities.

In our case we train a model to predict the goal-directed action class and unintentional action class present in it. We generate TCAMs specific to both these types of actions and use them along with the class-agnostic activation weights, *i.e.* λ, to localize these respective regions in the video.

More formally, let $\mathbf{w}^{c^{\mathrm{IA}}}(k)$ and $\mathbf{w}^{c^{\mathrm{UA}}}(k)$, be the weight parameter which maps the k-th element in goal-directed feature $\mathbf{O}^{\mathrm{IA}}$ and unintentional feature $\mathbf{O}^{\mathrm{UA}}$ to the goal-directed class c^{IA} and unintentional class c^{IA} respectively. The Temporal Class Activation maps for the goal-directed and unintentional action can be computed using the following formulation:

$$\begin{aligned} \mathbf{C}_t^{\mathrm{IA}}(c^{\mathrm{IA}}) &= \sum_{k=1}^{m} \mathbf{w}^{c^{\mathrm{IA}}}(k)\mathbf{O}_t^{\mathrm{IA}} \\ \mathbf{C}_t^{\mathrm{UA}}(c^{\mathrm{UA}}) &= \sum_{k=1}^{m} \mathbf{w}^{c^{\mathrm{UA}}}(k)\mathbf{O}_t^{\mathrm{UA}} \end{aligned} \tag{4.10}$$

where m refers to the latent dimension of the output of the video encoder.

4.2 Loss Formulation

4.2.1 Multiple Instance Learning Loss

Following previous works in weakly supervised action localization (Liu *et al.*, 2019; Paul *et al.*, 2018; Min and Corso, 2020), we use the k-max Multiple Instance Learning

(MIL) (Zhou, 2004) loss function for classifying the goal-directed and unintentional activities in the video. For each video, we average out the top-k elements of the TCAMs, *i.e.*, $\mathbf{C}^{\text{IA}}$ and $\mathbf{C}^{\text{UA}}$ along the temporal axis for each class to obtain the video-level classification scores $A^{\text{IA}} \in \mathbb{R}^{N_{IA}}$ and $A^{\text{UA}} \in \mathbb{R}^{N_{UA}}$. Here, k is set by $\left[\frac{l}{s}\right]$ where s is a hyper-parameter that regulates the number of clips to consider when making the classification. We then apply a softmax function over class scores, in order to obtain a probability mass function (pmf) over the goal-directed as well as unintentional classes, *i.e.*, p^{IA} and p^{UA}. Let y^{IA} and y^{UA} be the ground truth label vectors for a video. We then l_1-normalize them to obtain ground-truth pmfs q^{IA} and q^{UA}. Finally we conduct cross entropy between these two pmfs.

$$\begin{aligned}
\mathcal{L}_{cls}^{\text{IA}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N_{\text{IA}}} -q_i^{\text{IA}}(j)\log\left(p_i^{\text{IA}}(j)\right) \\
\mathcal{L}_{cls}^{\text{UA}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N_{\text{UA}}} -q_i^{\text{UA}}(j)\log\left(p_i^{\text{UA}}(j)\right) \\
\mathcal{L}_{cls} &= \mathcal{L}_{cls}^{\text{IA}} + \mathcal{L}_{cls}^{\text{UA}},
\end{aligned} \tag{4.11}$$

where N corresponds to the total number of training videos.

4.2.2 Overlap Regularization

Let λ_t^{IA}, $\lambda_t^{\text{UA}} \in [0,1]\ \forall\ t \in [1,l]$ be the bottom-up attention weights for the goal-directed actions and unintentional action respectively, obtained from the respective attention modules. λ_t signifies the temporal attention weight for a clip t. During training, a trivial solution which could be learned by the model is to pay attention to the entire video when inferring the goal-directed and unintentional action, *i.e.*, $\lambda_t^{IA}, \lambda_t^{UA} = 1\ \forall\ t \in [1,l]$, though these actions take place at two distinct sections of the video. Simply applying the MIL loss cannot guarantee that such distinctions can be learnt from the data. We solve this problem by appending an additional

regularization term on the overlap of these attention weights:

$$\begin{aligned}
\mathcal{L}_{IA} &= \texttt{max}\Big(0, \frac{\sum_{r}^{N_{T^{\text{UA}}}} \lambda_{T_r^{\text{UA}}}^{IA}}{N_{T^{\text{UA}}}} - \frac{l}{p}\Big) \\
\mathcal{L}_{\text{UA}} &= \texttt{max}\Big(0, \frac{\sum_{r}^{N_{T^{\text{IA}}}} \lambda_{T_r^{\text{IA}}}^{\text{UA}}}{N_{T^{\text{IA}}}} - \frac{l}{p}\Big) \\
\mathcal{L}_{overlap} &= \mathcal{L}_{\text{IA}} + \mathcal{L}_{\text{UA}},
\end{aligned} \tag{4.12}$$

where T^{IA} and T^{UA} are the set of temporal indices of the bottom-up goal-directed and unintentional attention weights at which they are more than a predefined threshold. $N_{T^{\text{IA}}}$ and $N_{T^{\text{UA}}}$ are the lengths of the sets of activated temporal indices. p is a design parameter which controls the amount of allowed overlap between these two attention maps. Lower the value of p, lower the penalization of overlaps. In the goal-directed as well as unintentional regions of the video, the attention weights should ideally be low at the borders of their respective ground truth region and high towards the center of this region. Hence we view $\lambda_{\text{IA}}, \lambda_{\text{UA}}$ as Gaussian distributions $\mathbf{P}_{\text{IA}} \sim \mathcal{N}(\mu_{\text{IA}}, \sigma_{\text{IA}}^2)$ and $\mathbf{P}_{\text{UA}} \sim \mathcal{N}(\mu_{\text{UA}}, \sigma_{\text{UA}}^2)$. Every unintentional action begins with an agent performing a goal-directed action in order to achieve it's goal, which then gets disrupted and transitions into an unintentional action. Using this prior that a goal-directed action transitions into an unintentional action, we need to ensure $\mu_{\text{IA}} < \mu_{\text{UA}}$. We approach this by formulating the following regularization:

$$\begin{aligned}
\mu_{\text{IA}} &= \frac{\sum_{t=1}^{l} P_t^{\lambda_{\text{IA}}} \cdot t}{\sum_{t=1}^{l} P_t^{\lambda_{\text{IA}}}} \\
\mu_{\text{UA}} &= \frac{\sum_{t=1}^{l} P_t^{\lambda_{\text{UA}}} \cdot t}{\sum_{t=1}^{l} P_t^{\lambda_{\text{UA}}}} \\
\mathcal{L}_{order} &= \texttt{max}(0, \frac{\mu_{\text{IA}} - \mu_{\text{UA}}}{l} + \frac{l}{q}),
\end{aligned} \tag{4.13}$$

where $P^{\lambda_{\text{IA}}}$ and $P^{\lambda_{\text{UA}}}$ are probability distributions obtained by applying softmax over the temporal axis of λ_{IA} and λ_{UA} respectively. q is a design parameter that helps control the margin by which μ_{UA} has to be greater than μ_{IA}. Our model is end-to-

end trained with the overall loss as follows:

$$\mathcal{L} = \lambda L_{cls} + (1 - \lambda)(\mathcal{L}_{overlap} + \mathcal{L}_{order}), \tag{4.14}$$

where λ is the weighting hyper-parameter that controls the trade-off between MIL loss and overlap regularization.

4.3 Classification and Localization

After training our network, we use it to classify goal-directed and unintentional actions as well as localize the regions in which they occur. For a single video, after obtaining the pmf p^{IA} and p^{UA} over each of the classes, as mentioned in Section 4.2.1, we use mean average precision (mAP) to conduct evaluation for the classification task. For localization of the goal-directed and unintentional regions, we consider only categories having classification scores *i.e.*, A^{IA} and A^{UA} above 0. For each of these categories, we first scale the respective TCAM outputs to [0,1] using a Sigmoid function and weight these using the bottom-up attention weights. This can be formally expressed by:

$$\begin{aligned} \psi^{\text{IA}}(c^{\text{IA}}) &= \lambda_{\text{IA}} \cdot \texttt{Sigmoid}(C^{\text{IA}}(c^{\text{IA}})) \qquad c^{\text{IA}} \in [1, N_{\text{IA}}], \\ \psi^{\text{UA}}(c^{\text{UA}}) &= \lambda_{\text{UA}} \cdot \texttt{Sigmoid}(C^{\text{UA}}(c^{\text{UA}})) \quad c^{\text{UA}} \in [1, N_{\text{UA}}], \end{aligned} \tag{4.15}$$

where $\psi^{\text{IA}}(c^{\text{IA}})$, $\psi^{\text{UA}}(c^{\text{UA}}) \in \mathbb{R}^l$ are the weighted TCAMs, for the respective classes c^{IA} and c^{UA}. We finally threshold $\psi^{\text{IA}}(c^{\text{IA}})$ and $\psi^{\text{UA}}(c^{\text{UA}})$ to obtain the triplets $\langle s^{\text{IA}}, e^{\text{IA}}, c^{\text{IA}} \rangle$ and $\langle s^{\text{UA}}, e^{\text{UA}}, c^{\text{UA}} \rangle$.

Chapter 5

EXPERIMENTS AND RESULTS

5.1 Implementation Details

We extract RGB features by creating chunks of 16 consecutive and non-overlapping frames and using the I3D (Carreira and Zisserman, 2017) as well as R(2+1)D (Tran *et al.*, 2018) pretrained architectures to extract clip-level features from these chunks (details provided in Sec. 4.1.1.1). This backbone feature extractor is kept frozen throughout the entire training process. The kernel-size of all the 1-D convolutional layers for the bottom-up attention modules are set to 1. The learning rate and loss weighting function λ is set to 10^{-3} and 0.8 respectively. We set the MIL loss hyper-parameter s to 3. The parameters of the Overlap Regularization, p and q, are set to 1000 and 10 respectively. Finally we set the number of layers of our bidirectional GRU to 3. for the transformer encoder, we set the number of layers as 3 and number of heads as 8. Our network is implemented and trained on a machine with a single Tesla X Pascal GPU for 10,000 iterations using the Adam Optimizer (Kingma and Ba, 2014) with a batch size of 16.

5.2 Evaluation Metrics

We use interpolated Average Precision (AP) as the evaluation metric for evaluating the results on each action class. Given a descending score rank of videos for test class c, the $AP(\mathrm{c})$ is computed as :

$$AP(\mathrm{c}) = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\sum_{k=1}^{n} rel(k)} \tag{5.1}$$

where n is the total number of videos, $P(k)$ is the precision at cut-off k of the list, $rel(k)$ is an indicator function whose output is 1 if the video ranked at k is a true positive, and zero otherwise. Hence the denominator becomes the total of true positives in the list.

In order to get a single performance metric across all the classes, we calculate the Mean Average Precision (mAP) across all the classes, expressed formally as:

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c) \tag{5.2}$$

where C is the total number of classes, which is 44 for the goal-directed classes and 30 for the unintentional classes.

We can use this formula in a straightforward way to calculate the classification mAP (cMAP) where the class scores for each video can be given by probability mass functions over the class scores, p^{IA} for the goal-directed classes and p^{UA} for the unintentional classes.

However for calculating the detection mAP (dMAP), we calculate the mAP@IoU where IoU is the intersection over union calculated as:

$$IoU = \frac{R_p \cap R_{gt}}{R_p \cup R_{gt}} \tag{5.3}$$

where R_p is the predicted time range and R_{gt} is the ground truth time range.

Hence, for a IoU of threshold 0.5, this detection would be considered correct if the IoU ≥ 0.5.

The Mean Average Precision (mAP) is obtained by averaging the Average Precision (AP) scores across each class (goal-directed and unintentional). The Average Precision gives us the score that measures how good is our model in sorting video samples (cMAP) or localized segments (dMAP) for a certain class according to a score function (Classification scores for the cMAP and Intersection over Union for the dMAP).

5.3 Results and Analysis

5.3.1 Localization

Model	Feature	Segment	mAP @ IoU									
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	**Avg**
STPN	R(2+1)D	Goal	44.9	41.7	33.0	25.7	18.3	10.0	5.0	3.7	1.2	20.4
		UnInt	30.9	26.6	21.8	15.7	9.9	5.2	1.8	1.0	0.1	12.5
WTALC	R(2+1)D	Goal	45.1	41.8	36.1	28.9	22.8	15.9	10.4	8.1	2.0	23.5
		UnInt	25.5	21.2	15.3	12.6	7.7	4.3	2.3	1.0	0.5	10.1
Ours	R(2+1)D	Goal	45.3	45.1	44.0	41.8	39.1	29.5	21.9	13.9	3.5	31.6
		UnInt	34.6	33.4	28.4	23.6	19.5	15.0	10.0	3.4	1.0	18.8
STPN	I3D	Goal	44.8	42.8	34.9	27.8	19.9	11.1	6.1	4.0	1.6	21.5
		UnInt	36.3	31.3	26.1	19.5	13.0	6.8	1.7	0.6	0.02	15.0
WTALC	I3D	Goal	38.8	36.4	30.4	26.3	18.6	13.1	7.2	4.5	1.8	19.7
		UnInt	22.9	18.4	14.2	11.0	6.8	3.6	1.2	0.5	0.1	8.8
Ours	I3D	Goal	**51.5**	**51.3**	**49.9**	**44.9**	**41.1**	**32.5**	**24.3**	**14.4**	**5.0**	**35.0**
		UnInt	**39.4**	**39.0**	**36.4**	**32.2**	**30.0**	**26.6**	**17.6**	**10.2**	**2.8**	**26.0**

Table 5.1: Performance Comparison of Our Model with Competitive Weakly Supervised Action Localization (WSAL) Models. We Adjust the WSAL Models by Attaching Two Classification Heads to Compute Two TCAMs (for the Goal-directed and Unintentional Action). We Then Retrain It on Our Dataset (W-Oops). We Can See That Our Model Significantly Outperforms the Other Methods.

Our model should be able to focus on the correct regions of the video in order to infer the goal-directed and unintentional action segments, hence understanding the

transition between these two.

In order to evaluate our model on the task of localizing goal-directed as well as unintentional segments, we follow the standard evaluation protocol for temporal localization tasks by calculating the mean average precision (mAP) over different intersection over union (IoU) thresholds for both the types of actions. Since there are no quantitative results reported on our dataset, we use competitive models from the traditional weakly supervised action localization task as baselines. Since these models are trained using only one classification head which is used to identify the atomic actions in the video, we repurpose these models by adding an additional classification head (for the goal-directed and unintentional action) and bottom-up attention module (in the case of STPN (Nguyen *et al.*, 2018)) to adapt it to our task. We then retrain these models on our dataset and report quantitative results for comparison in Tab. 5.1.

It may be noted that our method performs significantly better than other weakly supervised methods on this task, when using the same backbone. For example, the average mAP@IoU score of our method outperforms STPN by 13.5% for the goal-directed action and 11% for the unintentional action, when using an I3D backbone. We conjecture that this localization improvement is due to our overlap regularization on the bottom-up attention weights since it enforces the model to focus on distinct portions of the action scene while ensuring the temporal order of the actions, which is a crucial property for solving this task. The qualitative results shown in the Fig. 5.5, Fig. 5.6 and Fig. 5.7 show how the WSAL models focus on overlapping regions when inferring the goal-directed/unintentional action which reduces it's localization performance.

Figure 5.1: Example of a Video Where Openpose Is Giving Missing as Well as Wrong Results. In This Video, and Agent Is Driving a Car and Is Not Directly Seen in the Video. Hence, Openpose Is Not Able to Extract the Keypoint Coordinates in the First Few Frames, and Even in the Latter Frames Where It Is Extracted the Keypoint Coordinates Are Wrong, as They Do Not Correspond to the Driver.

Feature	Segment	**mAP@IoU**			
		0.3	0.5	0.9	Avg(0.1:0.9)
RGB (I3D)	Goal	49.9	41.1	5.0	35.0
	UnInt	36.4	30.0	2.8	26.0
RGB (I3D) + Skeleton	Goal	45.4	40.2	4.1	32.5
	UnInt	31.9	24.8	2.2	22.2

Table 5.2: Analysis of the Effect of Skeleton Features

5.3.1.1 Analysis of Addition of Skeleton Features

We now provide analysis of our model's performance using skeleton features extracted in Sec. 4.1.1.2 along with RGB features, compared with only RGB features in Tab. 5.2. We can see that the performance decreases, from 35.0% to 32.5% for the goal-directed mAP@IoU and from 26.0% to 22.2% for the unintentional mAP@IoU. We conjecture that this performance decrease could be due to the noise introduced by the incorrect/missing keypoint coordinates at certain frames, as well as due to some of the

$\mathcal{L}_{cls}$	$\mathcal{L}_{order}$	$\mathcal{L}_{overlap}$	Segment	mAP @ IoU			
				0.3	0.5	0.9	Avg(0.1:0.9)
✓	-	-	Goal	34.7	17.6	0.9	21.2
			UnInt	31.1	14.4	0.1	17.4
✓	✓	-	Goal	46.3	35.2	2.7	30.1
			UnInt	31.7	17.9	0.7	19.0
✓	✓	✓	Goal	49.9	41.1	5.0	35.0
			UnInt	36.4	30.0	2.8	26.0

Table 5.3: Ablation Study on Contributions of Different Losses in Our Model.

videos which involve an agent driving a vehicle and hence the agent is partially or completely not seen in the video, an example shown in Fig. 5.1

5.3.1.2 Analysis of the Contribution of Overlap Regularization

We conduct an ablation study to analyse various components of our model. We analyse the significance of the overlap regularization introduced in Section 4.2.2. We observe very clearly in Tab. 5.3 that only using $\mathcal{L}_{cls}$ is not sufficient to localize the goal-directed and unintentional actions, and our final model performs the best. This implies that all components are necessary in order to achieve the best performance and each one is effective.

5.3.1.3 Analysis of Hyper-Parameters of Overlap Regularization

We further analyse the importance of the hyper-parameters p and q used in the overlap regularization in Fig. 5.2. We can see that increasing p from 1 to 10^3 results in a significant increase in the goal-directed as well as unintentional average mAP@IoU.

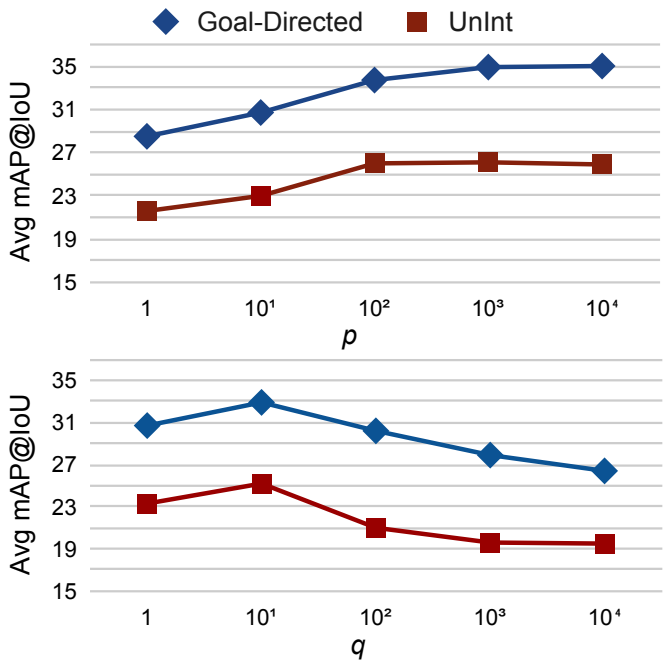

Figure 5.2: Effect on Average mAP@IoU for the Goal-directed and Unintentional Action When Changing p (Top) and q (Bottom).

This shows that the localization performance increases by penalizing the overlap of the bottom-up attentions more, but plateaus after the 10^3 mark.

Analysing the q hyper-parameter, we notice that increasing the value of q decreases the performance. Since increasing the value of q results in a lower margin of separation between the expectations of the goal-directed and unintentional bottom-up attention weights, we can conclude that a lower value of q, *i.e.*, higher margins of separation helps achieve a better localization performance. However, $q = 1$ signifies the extreme case when the margin is equal to the length of the clips, forcing the attention maps to be at two separate ends of the temporal axis, thereby hurting the performance. Fig. 5.4 shows qualitative examples of localizing the goal-directed and unintentional segments on our W-Oops dataset. We further provide more qualitative examples in Fig. 5.5, Fig. 5.6 and Fig. 5.7 which compare our method with previous WSAL

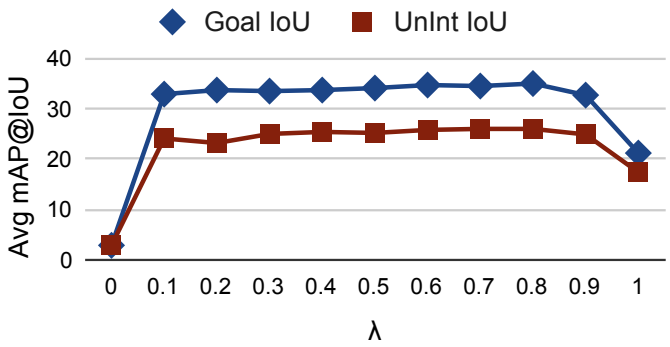

Figure 5.3: Effect on Average mAP@IoUfor the Goal-directed and Unintentional Action When Changing Weight Tradeoff Parameter λ.

methods.

5.3.1.4 Analysis of Weight Trade-Off Parameter λ

λ is the scalar parameter used to control the tradeoff between the Multiple Instance Learning Loss (MIL) and the Overlap Regularization. We study the effects of changing this parameter in the range of [0,1], where λ=0 corresponds to purely MIL Loss and λ=1 corresponds to purely Overlap Regularization. As seen in Fig. 5.3, we observe that for all values of $0.2 \leq \lambda \leq 0.8$, the average mAP@IoU, is very similar, but on closer observation $\lambda = 0.8$ achieves the best performance for both the goal-directed action and unintentional action.

5.3.1.5 Analysis of Video-Embedding Module

We now analyse the effectiveness of our video embedding module, by removing the module and using only the raw features from the frozen feature extractor. We also replace the GRU with a transformer encoder, which is another encoding module for sequential data, originally a sub-component of the original transformer architecture (Vaswani *et al.*, 2017), which has achieved state of the art results on many vision

(Zhang *et al.*, 2020a; Carion *et al.*, 2020; Zeng *et al.*, 2020; Dosovitskiy *et al.*, 2020; Zhou *et al.*, 2018b; Zhang *et al.*, 2020b) as well as NLP (Devlin *et al.*, 2018; Yang *et al.*, 2019; Keskar *et al.*, 2019; Wu *et al.*, 2020) tasks. As seen in table 5.4, we can see that using static backbone features result in a very poor localization performance. Additionally it is also interesting to observe that the GRU performs better than the transformer encoder.

Embedding Module	Segment	**mAP@IoU**			
		0.3	0.5	0.9	Avg(0.1:0.9)
None	Goal	30.2	16.5	1.3	18.7
	UnInt	18.6	9.4	0.02	11.1
Transformer Encoder	Goal	49.1	41.5	2.7	34.9
	UnInt	31.7	17.9	0.7	22.7
GRU	Goal	49.9	41.1	5.0	35.0
	UnInt	36.4	30.0	2.8	26.0

Table 5.4: Ablation Study of the Contribution of the Video Embedding Module.

5.3.2 Classification

Given any video our model is trained to predict the goal-directed action as well as the unintentional action it eventually transitions into. Following previous works (Nguyen *et al.*, 2018; Paul *et al.*, 2018), we use mean average precision (mAP) to evaluate the classification performance of our model on predicting the goal-directed action as well as unintentional action. We report our results in Tab. 5.5. It is interesting to note that our method performs the best on the classification task as

Architecture	Feature	GOAL cMAP	UNINT. cMAP
Chance	-	2.7	3.3
STPN	R(2+1)D	44.0	32.6
WTALC	R(2+1)D	48.5	37.5
Ours	R(2+1)D	50.5	38.4
STPN	I3D	45.3	37.5
WTALC	I3D	50.2	38.2
Ours	I3D	52.6	41.1

Table 5.5: Mean Average Precision of Activity Classification Results Using Different Methods. First Row Shows the mAP of Random Chance.

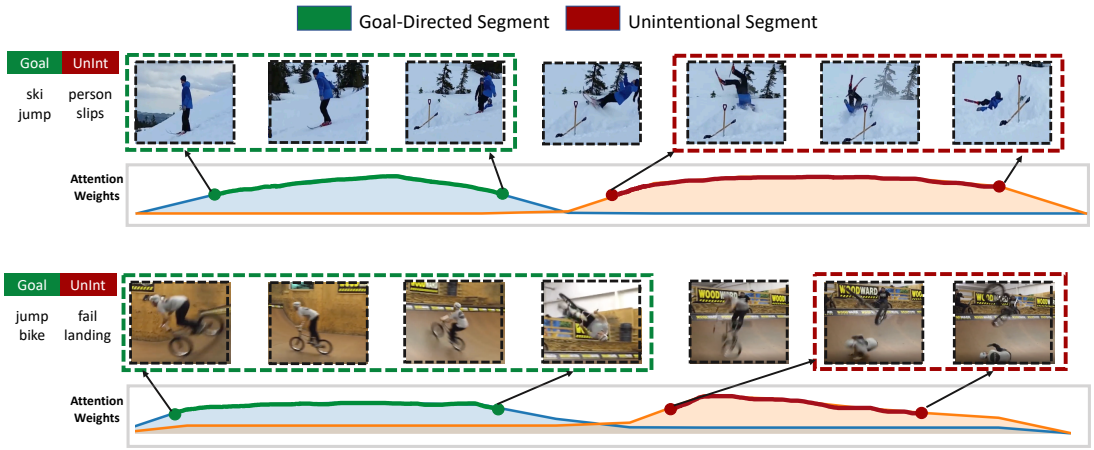

Figure 5.4: Our Method Is Able to Identify the Temporal Regions That Correspond to Goal-directed/Unintentional Activity via the Produced Weighted TCAMs. Blue Attention Maps Correspond to the Goal-directed Action. Orange Attention Maps Correspond to the Unintentional Action.

well. For example, it performs 7.3% higher on the Goal cMAP and 3.6% higher on the Unintentional cMAP than STPN when using an I3D backbone.

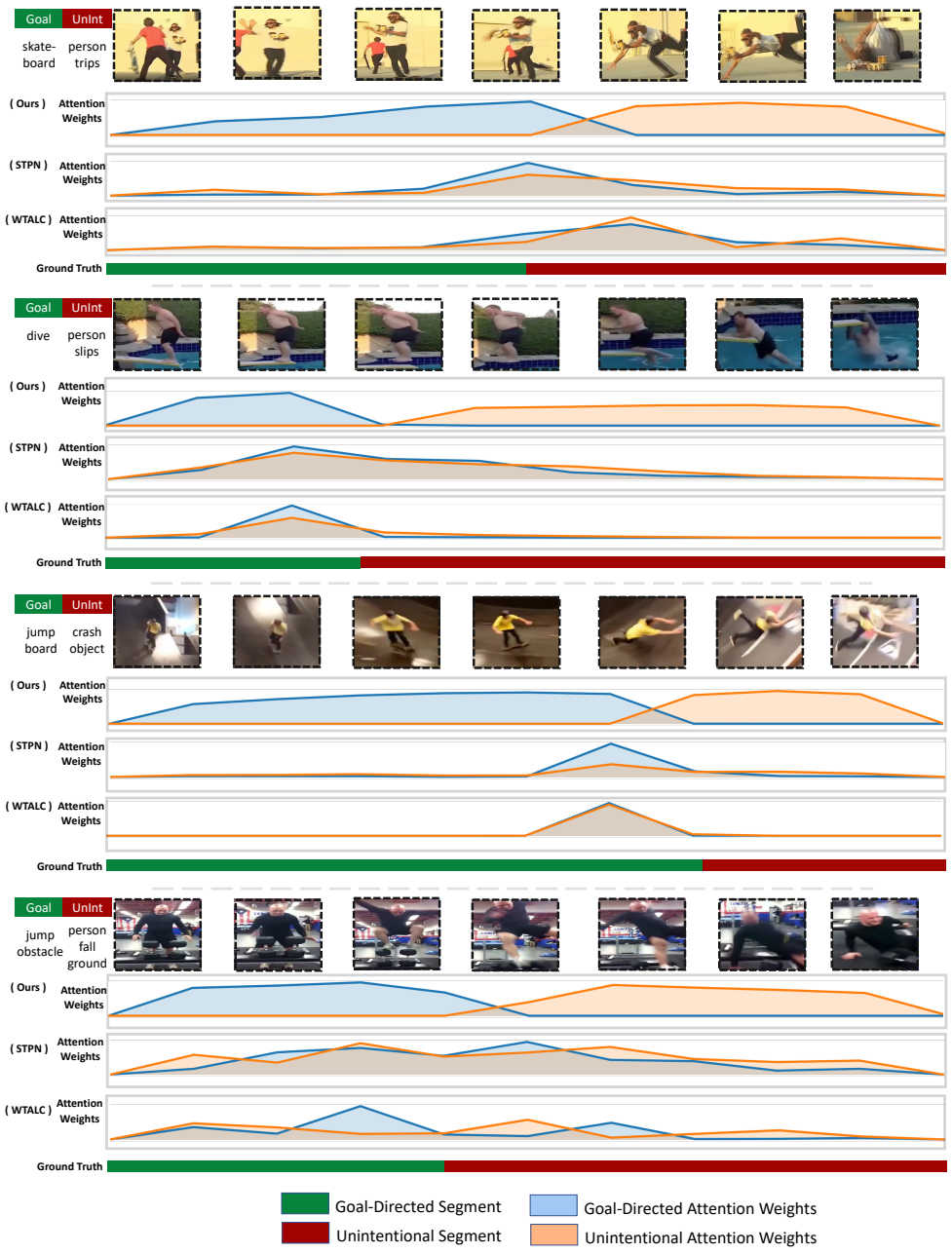

Figure 5.5: Qualitative Results of Our Model's Outputs. We Provide Attention Weights Outputted from STPN Trained on Our Dataset, as Well as the Ground Truth Segments for Comparison.

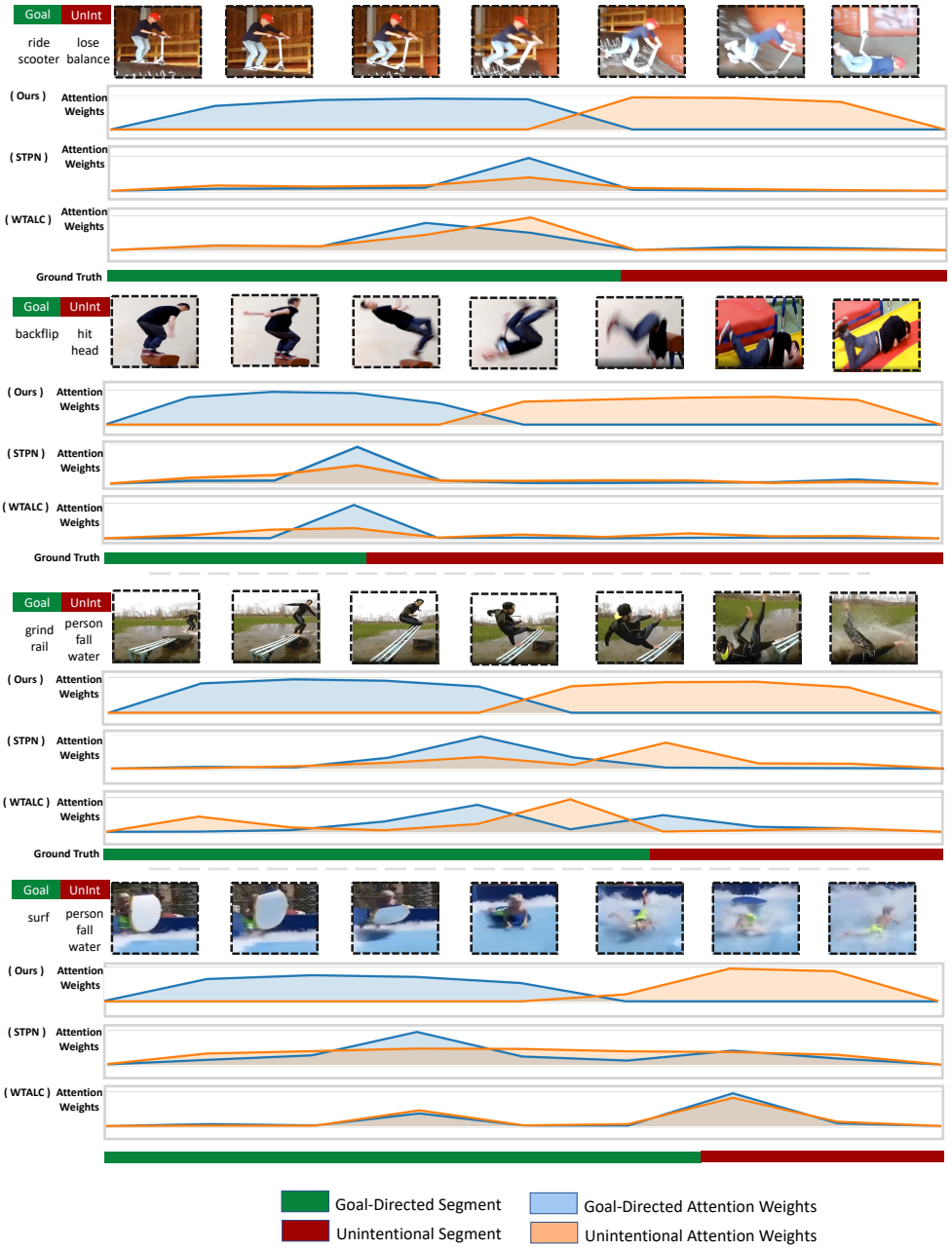

Figure 5.6: Qualitative Results of Our Model's Outputs. We Provide Attention Weights Outputted from STPN Trained on Our Dataset, as Well as the Ground Truth Segments for Comparison.

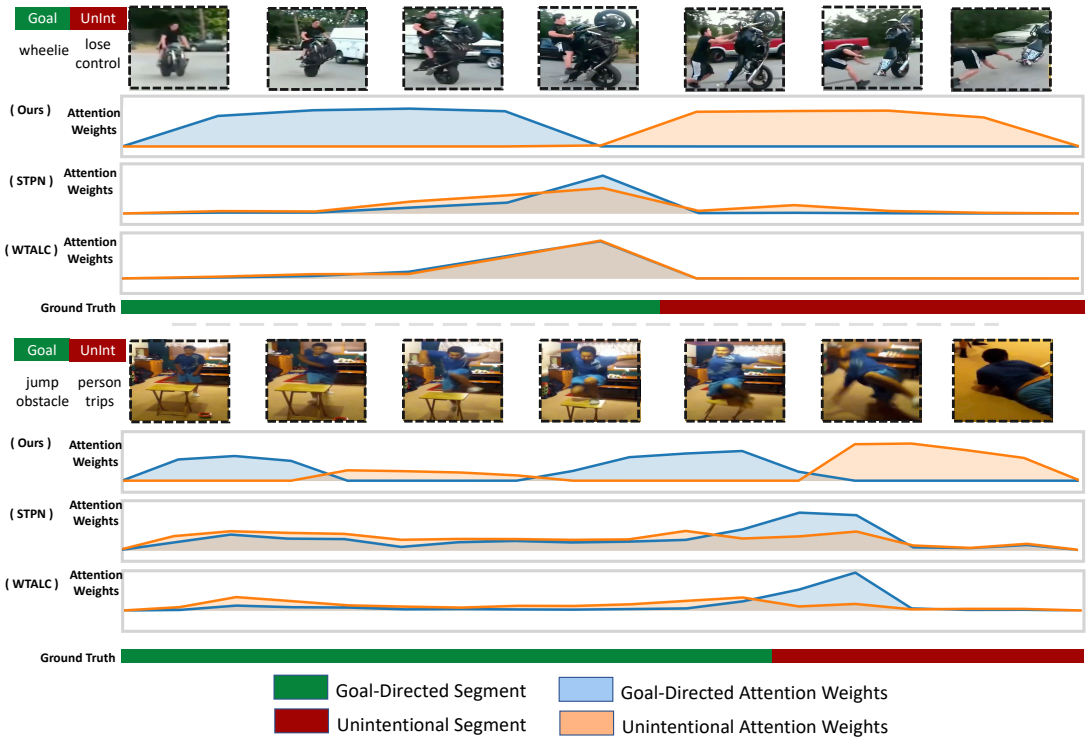

Figure 5.7: Qualitative Results of Our Model's Outputs. We Provide Attention Weights Outputted from STPN Trained on Our Dataset, as Well as the Ground Truth Segments for Comparison.

Chapter 6

CONCLUSION

In this thesis, we propose W-Oops, an augmented unintentional human activity dataset that consists of both goal-directed and unintentional video-level activity annotations, built upon Oops (Epstein *et al.*, 2019). We also shed the importance of fine grained and teleological understanding of unintentional human actions, and how artificially intelligent agents could benefit while incorporating these abilities.

In order to address the expensive temporal labelling process, we consider a weakly supervised task to infer the respective classes as well as the temporal regions in which they occur using only the video-level activity annotations. We further build a neural network architecture which employs a novel overlap regularization on top of the bottom-up attention weights outputted by our attention module, which helps the model focus on distinct parts of the video while maintaining the temporal ordering of these actions when inferring the temporal regions. We conclude from our experiments that our method significantly outperforms previous WSAL baselines on our benchmark.

Finally we provide extensive ablation studies to understand the various components of our architecture, and also experiment with human skeleton features, which has shown promise for detecting human actions.

REFERENCES

"Mens-rea", URL `https://www.law.cornell.edu/wex/mens\_rea` (2011).

Abu-El-Haija, S., N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark", arXiv preprint arXiv:1609.08675 (2016).

Blank, M., L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes", in "Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1", vol. 2, pp. 1395–1402 (IEEE, 2005).

Bojanowski, P., R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid and J. Sivic, "Weakly supervised action labeling in videos under ordering constraints", in "European Conference on Computer Vision", pp. 628–643 (Springer, 2014).

Bojanowski, P., R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce and C. Schmid, "Weakly-supervised alignment of video with text", in "Proceedings of the IEEE international conference on computer vision", pp. 4462–4470 (2015).

Brandone, A. C. and H. M. Wellman, "You can't always get what you want: Infants understand failed goal-directed actions", Psychological science **20**, 1, 85–91 (2009).

Caba Heilbron, F., V. Escorcia, B. Ghanem and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding", in "Proceedings of the ieee conference on computer vision and pattern recognition", pp. 961–970 (2015).

Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields", IEEE transactions on pattern analysis and machine intelligence **43**, 1, 172–186 (2019).

Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-end object detection with transformers", in "European Conference on Computer Vision", pp. 213–229 (Springer, 2020).

Carreira, J., E. Noland, A. Banki-Horvath, C. Hillier and A. Zisserman, "A short note about kinetics-600", arXiv preprint arXiv:1808.01340 (2018).

Carreira, J. and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset", in "proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 6299–6308 (2017).

Chung, J., C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", arXiv preprint arXiv:1412.3555 (2014).

Csibra, G., "Teleological and referential understanding of action in infancy", Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences **358**, 1431, 447–458 (2003).

Csibra, G., "Goal attribution to inanimate agents by 6.5-month-old infants", Cognition **107**, 2, 705–717 (2008).

Csibra, G. and G. Gergely, "'obsessed with goals': Functions and mechanisms of teleological interpretation of actions in humans", Acta psychologica **124**, 1, 60–78 (2007).

Csibra, G. and G. Gergely, "Teleological understanding of actions", Navigating the social world: What infants, children, and other species can teach us pp. 38–43 (2013).

Csibra, G., G. Gergely, S. Bıró, O. Koos and M. Brockbank, "Goal attribution without agency cues: the perception of 'pure reason'in infancy", Cognition **72**, 3, 237–267 (1999).

Damen, D., H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 720–736 (2018).

Davoodikakhki, M. and K. Yin, "Hierarchical action classification with network pruning", in "International Symposium on Visual Computing", pp. 291–305 (Springer, 2020).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805 (2018).

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

Epstein, D., B. Chen and C. Vondrick, "Oops! predicting unintentional action in video", CoRR **abs/1911.11206**, URL `http://arxiv.org/abs/1911.11206` (2019).

Fang, Z., T. Gokhale, P. Banerjee, C. Baral and Y. Yang, "Video2commonsense: Generating commonsense descriptions to enrich video captioning", Conference on Empirical Methods in Natural Language Processing (2020).

Fang, Z. and A. M. López, "Intention recognition of pedestrians and cyclists by 2d pose estimation", IEEE Transactions on Intelligent Transportation Systems **21**, 11, 4773–4783 (2019).

Fouhey, D. F., W.-c. Kuo, A. A. Efros and J. Malik, "From lifestyle vlogs to everyday interactions", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 4991–5000 (2018).

Furnari, A. and G. M. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention", CoRR **abs/1905.09035**, URL `http://arxiv.org/abs/1905.09035` (2019).

Gergely, G. and G. Csibra, "Teleological reasoning in infancy: The naıve theory of rational action", Trends in cognitive sciences **7**, 7, 287–292 (2003).

Gergely, G., Z. Nádasdy, G. Csibra and S. Bíró, "Taking the intentional stance at 12 months of age", Cognition **56**, 2, 165–193 (1995).

Gkioxari, G. and J. Malik, "Finding action tubes", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 759–768 (2015).

Goyal, R., S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 5842–5850 (2017).

Gu, C., C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 6047–6056 (2018a).

Gu, C., C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 6047–6056 (2018b).

Henrich, J. and R. McElreath, "The evolution of cultural evolution", Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews **12**, 3, 123–135 (2003).

Hoai, M. and F. De la Torre, "Max-margin early event detectors", in "2012 IEEE Conference on Computer Vision and Pattern Recognition", pp. 2863–2870 (2012).

Hua, M., Y. Nan and S. Lian, "Falls prediction based on body keypoints and seq2seq architecture", in "Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops", pp. 0–0 (2019).

Huang, D.-A., L. Fei-Fei and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling", in "European Conference on Computer Vision", pp. 137–153 (Springer, 2016).

Kalfaoglu, M. E., S. Kalkan and A. A. Alatan, "Late temporal modeling in 3d cnn architectures with bert for action recognition", in "European Conference on Computer Vision", pp. 731–747 (Springer, 2020).

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in "Proceedings of the IEEE conference on Computer Vision and Pattern Recognition", pp. 1725–1732 (2014).

Kay, W., J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset", arXiv preprint arXiv:1705.06950 (2017).

Keskar, N. S., B. McCann, L. R. Varshney, C. Xiong and R. Socher, "Ctrl: A conditional transformer language model for controllable generation", arXiv preprint arXiv:1909.05858 (2019).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Kuehne, H., H. Jhuang, E. Garrote, T. Poggio and T. Serre, "Hmdb: a large video database for human motion recognition", in "2011 International conference on computer vision", pp. 2556–2563 (IEEE, 2011).

Kuehne, H., A. Richard and J. Gall, "Weakly supervised learning of actions from transcripts", Computer Vision and Image Understanding **163**, 78–89 (2017).

Laland, K. N., "Social learning strategies", Animal Learning & Behavior **32**, 1, 4–14 (2004).

Lee, P., Y. Uh and H. Byun, "Background suppression network for weakly-supervised temporal action localization", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 34, pp. 11320–11327 (2020).

Lei, J., L. Yu, T. L. Berg and M. Bansal, "What is more likely to happen next? video-and-language future event prediction", arXiv preprint arXiv:2010.07999 (2020).

Li, Z., C. Xu and B. Leng, "Angular triplet-center loss for multi-view 3d shape retrieval", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 33, pp. 8682–8689 (2019).

Lin, J., C. Gan and S. Han, "Tsm: Temporal shift module for efficient video understanding", in "Proceedings of the IEEE/CVF International Conference on Computer Vision", pp. 7083–7093 (2019).

Liu, D., T. Jiang and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 1298–1307 (2019).

Luvizon, D. C., D. Picard and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 5137–5146 (2018).

Malle, B. F. and J. Knobe, "The folk concept of intentionality", Journal of experimental social psychology **33**, 2, 101–121 (1997).

Malle, B. F., L. J. Moses and D. A. Baldwin, "Introduction: The significance of intentionality", Intentions and intentionality: Foundations of social cognition **1**, 24 (2001).

Martens, C., "Criminal intent: What it means and why it matters", URL `https://www.martenslawfirm.com/blog/2018/january/criminal-intent-what-it-means-and-why-it-matters` (2018).

Miech, A., I. Laptev, J. Sivic, H. Wang, L. Torresani and D. Tran, "Leveraging the present to anticipate the future in videos", in "2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)", pp. 2915–2922 (2019).

Miki, D., S. Chen and K. Demachi, "Weakly supervised graph convolutional neural network for human action localization", in "Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision", pp. 653–661 (2020).

Min, K. and J. J. Corso, "Adversarial background-aware loss for weakly-supervised temporal activity localization", in "European Conference on Computer Vision", pp. 283–299 (Springer, 2020).

Monfort, M., A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick *et al.*, "Moments in time dataset: one million videos for event understanding", IEEE transactions on pattern analysis and machine intelligence **42**, 2, 502–508 (2019).

Nguyen, P., T. Liu, G. Prasad and B. Han, "Weakly supervised action localization by sparse temporal pooling network", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 6752–6761 (2018).

Nguyen, P. X., G. Rogez, C. Fowlkes and D. Ramanan, "The open world of micro-videos", arXiv preprint arXiv:1603.09439 (2016).

Paul, S., S. Roy and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 563–579 (2018).

Rasouli, A., M. Rohani and J. Luo, "Pedestrian behavior prediction via multitask learning and categorical interaction modeling", arXiv preprint arXiv:2012.03298 (2020).

Richard, A., H. Kuehne and J. Gall, "Weakly supervised action learning with rnn based fine-to-coarse modeling", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 754–763 (2017).

Ryoo, M. S., "Human activity prediction: Early recognition of ongoing activities from streaming videos", in "2011 International Conference on Computer Vision", pp. 1036–1043 (2011).

Sadegh Aliakbarian, M., F. Sadat Saleh, M. Salzmann, B. Fernando, L. Petersson and L. Andersson, "Encouraging lstms to anticipate actions very early", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 280–289 (2017).

Schuldt, C., I. Laptev and B. Caputo, "Recognizing human actions: a local svm approach", in "Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.", vol. 3, pp. 32–36 (IEEE, 2004).

Shi, B., Q. Dai, Y. Mu and J. Wang, "Weakly-supervised action localization by generative attention modeling", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 1009–1019 (2020).

Shou, Z., H. Gao, L. Zhang, K. Miyazawa and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 154–171 (2018).

Shou, Z., D. Wang and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 1049–1058 (2016).

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

Simonyan, K. and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", arXiv preprint arXiv:1406.2199 (2014).

Solbach, M. D. and J. K. Tsotsos, "Vision-based fallen person detection for the elderly", in "Proceedings of the IEEE International Conference on Computer Vision Workshops", pp. 1433–1442 (2017).

Soomro, K., H. Idrees and M. Shah, "Action localization in videos through context walk", in "Proceedings of the IEEE international conference on computer vision", pp. 3280–3288 (2015).

Synakowski, S., Q. Feng and A. Martinez, "Adding knowledge to unsupervised algorithms for the recognition of intent", International Journal of Computer Vision URL `http://dx.doi.org/10.1007/s11263-020-01404-0` (2021).

Tran, D., H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition", in "Proceedings of the IEEE conference on Computer Vision and Pattern Recognition", pp. 6450–6459 (2018).

Tran, V., Y. Wang and M. Hoai, "Back to the future: Knowledge distillation for human action anticipation", CoRR **abs/1904.04868**, URL `http://arxiv.org/abs/1904.04868` (2019).

Varytimidis, D., F. Alonso-Fernandez, B. Duran and C. Englund, "Action and intention recognition of pedestrians in urban traffic", in "2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)", pp. 676–682 (2018).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is all you need", in "Advances in Neural Information Processing Systems", edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, vol. 30 (Curran Associates, Inc., 2017), URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vondrick, C., D. Oktay, H. Pirsiavash and A. Torralba, "Predicting motivations of actions by leveraging text", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2997–3005 (2016).

Vondrick, C., H. Pirsiavash and A. Torralba, "Anticipating the future by watching unlabeled video", CoRR **abs/1504.08023**, URL http://arxiv.org/abs/1504.08023 (2015).

Wang, C., Y. Wang and A. L. Yuille, "An approach to pose-based action recognition", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 915–922 (2013).

Wang, L., Y. Qiao and X. Tang, "Action recognition and detection by combining motion and appearance features", THUMOS14 Action Recognition Challenge **1**, 2, 2 (2014).

Wang, L., 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 "European conference on computer vision", pp. 20–36 (Springer, 2016).

Wei, P., Y. Liu, T. Shu, N. Zheng and S. Zhu, "Where and why are they looking? jointly inferring human attention and intentions in complex tasks", in "2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 6801–6809 (2018).

Wojke, N., A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric", in "2017 IEEE international conference on image processing (ICIP)", pp. 3645–3649 (IEEE, 2017).

Wu, C.-S., S. Hoi, R. Socher and C. Xiong, "Tod-bert: Pre-trained natural language understanding for task-oriented dialogues", arXiv preprint arXiv:2004.06871 (2020).

Yan, S., Y. Xiong and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition", in "Proceedings of the AAAI conference on artificial intelligence", vol. 32 (2018).

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding", arXiv preprint arXiv:1906.08237 (2019).

Yeung, S., O. Russakovsky, G. Mori and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2678–2687 (2016).

Zellers, R., Y. Bisk, A. Farhadi and Y. Choi, "From recognition to cognition: Visual commonsense reasoning", in "The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", (2019).

Zeng, Y., J. Fu and H. Chao, "Learning joint spatial-temporal transformations for video inpainting", in "European Conference on Computer Vision", pp. 528–543 (Springer, 2020).

Zhai, Y., L. Wang, W. Tang, Q. Zhang, J. Yuan and G. Hua, "Two-stream consensus network for weakly-supervised temporal action localization", in "European Conference on Computer Vision", pp. 37–54 (Springer, 2020).

Zhang, D., H. Zhang, J. Tang, M. Wang, X. Hua and Q. Sun, "Feature pyramid transformer", in "European Conference on Computer Vision", pp. 323–339 (Springer, 2020a).

Zhang, Z., Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu and Z.-J. Zha, "Object relational graph with teacher-recommended learning for video captioning", in "Proceedings of the IEEE/CVF conference on computer vision and pattern recognition", pp. 13278–13288 (2020b).

Zhao, Y., Y. Xiong, L. Wang, Z. Wu, X. Tang and D. Lin, "Temporal action detection with structured segment networks", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 2914–2923 (2017).

Zhou, B., A. Andonian, A. Oliva and A. Torralba, "Temporal relational reasoning in videos", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 803–818 (2018a).

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

Zhou, L., Y. Zhou, J. J. Corso, R. Socher and C. Xiong, "End-to-end dense video captioning with masked transformer", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 8739–8748 (2018b).

Zhou, Z.-H., "Multi-instance learning: A survey", Department of Computer Science & Technology, Nanjing University, Tech. Rep **2** (2004).

APPENDIX A

ANNOTATION TOOL

The annotation tool used for the human evaluation and correction process as described in Chap. 3 is show in Fig. A.1. It is made using the streamlit[1] library. We provide a video to the evaluator along with the actions extracted from the annotations as mentioned in Chap. 3. The evaluator can then view the videos and mark the goal-directed actions as well as unintentional action as either 'Good' (G) or 'Poor' (P), with reference to the video. 'Good' is given to an action which is entailed in the video and 'Poor' otherwise. In case the evaluator marks an action as 'Poor', they can then choose another action from the already present list of total actions, or else add a new action if not contained in the list. The evaluator also has an option to not keep the video in the case the goal of the agent in the video was ambiguous. Once they are done with this process they can then hit submit, which would then load the next video.

[1] `https://streamlit.io/`

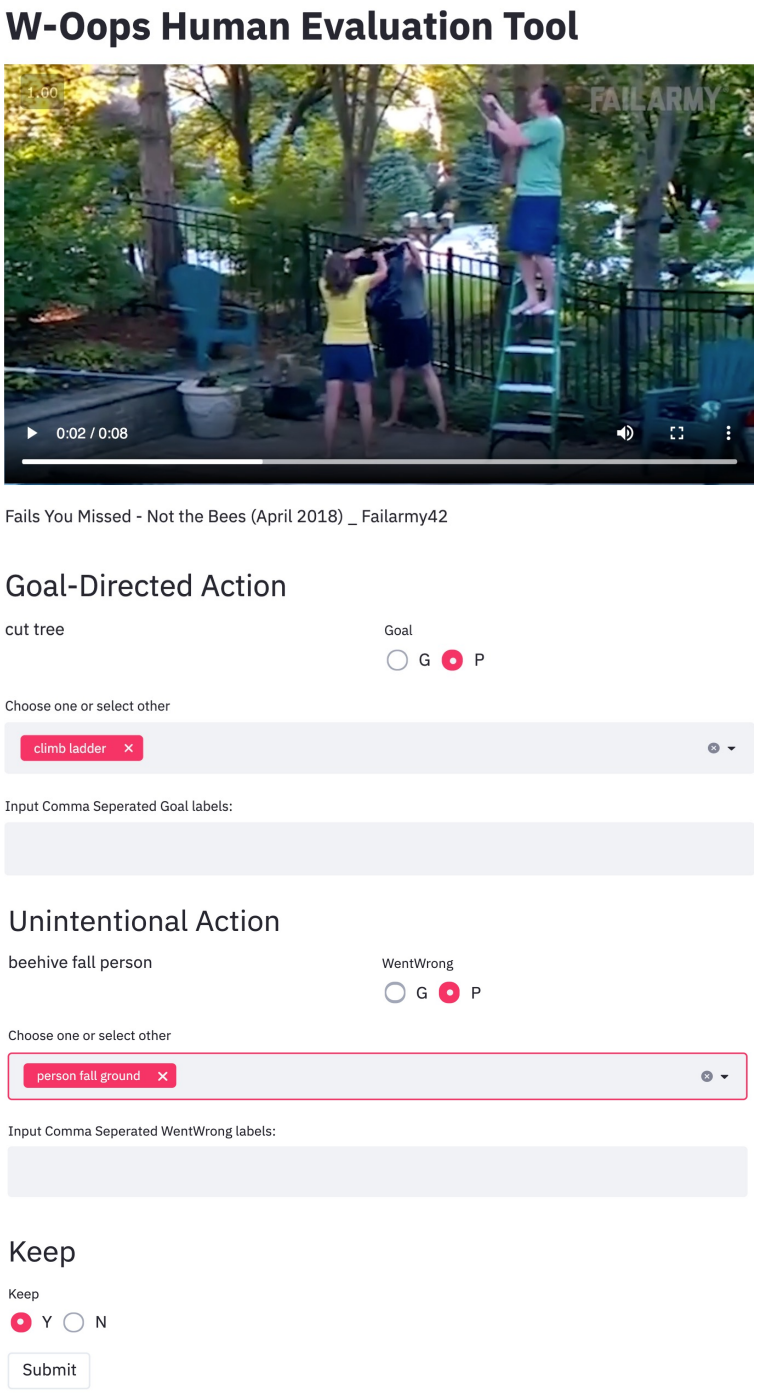

Figure A.1: Interface for W-Oops Annotations, Where We Ask the Annotators to Rate the Semi-automatically Extracted Goal-directed and Unintentional Actions as 'good' or 'poor', and If 'poor', to Choose from a Fixed List of Already Present Actions or Create Their Own. They Also Have an Option to Indicate Whether or Not to Keep They Video in the Case of the Goal in the Video Being Ambiguous.