博　士　論　文

Modeling Human Behaviors from First-Person Perspectives

(一人称視点映像解析による人物行動のモデリング)

東京大学大学院
情報理工学系研究科
電子情報学専攻

48-187409　黄　逸飛

指導教員　　佐藤 洋一　教授

令和 2 年　12 月

© Copyright by Yifei Huang 2020.
All rights reserved.

Abstract

The commercial success of wearable cameras allows people to acquire a huge quantity of data about their daily lives and activities from a first-person point of view: being able to observe what camera wearers see and where they are looking. This human-centric perspective is naturally suited to gathering visual information about our everyday observations and interactions, which in turn can reveal our attention and activities. Augmenting the wearable camera with eye-trackers, we can furthermore consider using the measured human gaze to better understand the user's activities and even intention. Such unique characteristics of videos captured by wearable cameras facilitate the automatic modeling of human behavior in the first-person point-of-view paradigm, which has a wide range of applications from human-to-robot imitation learning to developmental psychology.

In this thesis, I present machine learning based models for automatic human behavior modeling that focus on two types of human behavior: human gaze and human action. These models learn to leverage the rich high-level semantic information enclosed in the first-person videos to tackle several major challenges such as occlusion and rapid head motion. The thesis work is composed of three main components that address the modeling of human behavior from different aspects: (1) A graph-based method for localizing and recognizing human actions from videos using the temporal relation among actions; (2) A gaze prediction approach for first-person videos that uses task-dependent attention transition; (3) A unified framework for jointly recognizing human action and predicting human gaze, since human action and gaze are deeply correlated.

The modeling of human actions has always been one of the fundamental research problems of computer vision. Different from the third-person perspective, actions from the first-person perspective are more difficult to capture due to the camera motion and limitations on the field of view. To alleviate these problems, I design a method based on Graph Convolution Networks (GCNs) to leverage the relation of multiple action segments in various time spans to help with better

localization and classification of human actions. By applying graph convolution, we can update each node's representation based on its relation with neighboring nodes. The updated representation can be used for improved action segmentation.

The study of human gaze plays a vital role in understanding the attention mechanism of humans since gaze is one of the most direct representation of human attention. Since human gaze is not always attracted by the salient regions but also dependent on the undergoing task, I propose a hybrid model for gaze prediction based on deep neural networks that integrates task-dependent attention transition with bottom-up saliency estimation.

Building on the work of human gaze prediction and action segmentation, a further step is taken to study the mutual influence of the two human behaviors. My assumption is that during the procedure of performing a manipulation task, on the one hand, what a person is doing determines where the person is looking at. On the other hand, the gaze location reveals gaze regions that contain important and information about the undergoing action and also the non-gaze regions that include complimentary clues for differentiating some fine-grained actions. To validate the assumptions, I use a mutual context network (MCN) that jointly learns action-dependent gaze prediction and gaze-guided action recognition in an end-to-end manner. Experiments on multiple public first-person video datasets demonstrate that our MCN achieves the state-of-the-art performance of both gaze prediction and action recognition. Our experiments also show that action-dependent gaze patterns could be learned with our method.

Acknowledgements

I could never have completed this work without the support and assistance of many people. First and foremost, I would like to express the deepest gratitude to my adviser, Prof. Yoichi Sato, for his kind advising, valuable suggestions, and good-hearted encouragement in academics. Without his tolerant and open-minded style, I could not have a chance to start my Ph.D. pursuit in computer vision without any experience. With his help, I learned how to read an academic paper; how to formalize a research problem; and how to write a paper and present the work. His wide curiosity inspires me to think about what a researcher really is. I also would like to express grateful thanks to Prof. Yusuke Sugano and Prof. Minjie Cai. With their help, I learned how to tackle research problems, how to compose and express an idea more logically, and most importantly how to think like a scientist rather than an engineer. Their insight and passion for the research encourage me to face and overcome difficulties and setbacks with perseverance and optimism. Once again, I thank them from the bottom of my heart for their consistent and generous support which helps me grow as a researcher.

This thesis would not have been possible without generous financial support from the Graduate Program for Social ICT Global Creative Leaders (GCL) of The University of Tokyo by MEXT (Ministry of Education, Culture, Sports, Science and Technology). GCL program also provides a chance for me to make good friendships with other recipients, which makes my life in Japan happy and memorable. These supports are gratefully acknowledged.

I would also like to thank all members of the Sato lab. The time I spent in Sato lab was very exciting and wonderful thanks to them. I learned a lot of things through weekly meetings and everyday discussions with them. Last but not least, I would like to express my deepest love and gratitude to

my parents and all my family members. This thesis would not have been possible without their continued patience and endless support.

December 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, with the rapid development of modern technology, wearable cameras have become affordable and user-friendly for a mass amount of people. The increase of wearable cameras brings out a large collection of first-person videos, for example, life-logging videos. Different from traditional third-person videos that are taken by a fixed camera, first-person videos recorded by wearable or head-mounted cameras capture human behavior from a natural, egocentric perspective: they capture exactly what the camera wearers see (Figure 1.1). Since this unique perspective reflects the way humans observe and interact with the surrounding, the modeling of human behavior in first-person videos has attracted a great deal of interest from researchers as a new research paradigm on exploring how humans process the rich and complex environmental information as intelligent creatures. Furthermore, the knowledge of human behavior can be valuable for HCI applications [DDMF+18] and could be transferred to robots via techniques such as imitation learning [NCA+17], which are both indispensable prerequisites for the next-generation intelligent robots. Thus, in this thesis I aim to automate the modeling of human behavior in daily activities using videos taken from first-person wearable cameras.

Human behavior can be decomposed as a series of perceptions and actions: people first perceive the world using receptors such as eyes and ears, and then take actions based on their goal and the analysis of the surrounding environment. Based on this phenomenon, this thesis first addresses the auto-

Figure 1.1: (a) Examples of first-person cameras: (1) Google glasses [goo], (2) Gopro Hero7 [gop], (3) Tobii Pro 2 [tob], (4) ETMobile [etm]. (b) Examples of first-person videos. (c) Examples of third-person videos. Compared with third-person videos, first-person videos recorded by first-person cameras naturally represents the camera wearer's view, which is an ideal perspective for analyzing and modeling human behaviors.

matic modeling of the two types of human behaviors separately: and (1) the automated modeling of human actions, as actions can be seen as the most explicit demonstration of human behavior, and (2) the automatic modeling of human gaze, since gaze is one of the most important manners that human visually perceive the world, Building upon the knowledge of modeling human gaze and actions separately, this thesis takes a further step to consider their mutual context for jointly modeling human gaze and actions.

When we humans want to achieve our goals, we take actions, which are of the most explicit human behaviors, based on our perception and goals. From the first-person perspective, actions could be interpreted as: "what am I doing?". The modeling of human actions can have increasingly important consequences such as automatic video labeling [MMH+08], highlight extraction [YMR16], augmented reality [SR17], robotics [KKUG07], etc. Therefore, in this thesis the first focus is on the action modeling from first-person perspective. The objective is to not only detect when the action happens but also what kind of action happens.

When we humans perceive the surrounding scene using our eyes, we can quickly process visual information by focusing only on a small region of the whole scene. This is known as the attention mechanism and is mainly driven

by the human gaze. The knowledge of the human gaze, *i.e.*, where people look, is extremely important for various fields both in the research community and in industry. For example, human gaze information can be used for explainable AI [TG20] and improving the performance of multiple research tasks like zero-shot learning [KASB17] and video summarization [XML+15]. The knowledge of human gaze behavior can also be applied in skill assessment [DPSCRDS17] and autism diagnosis [BHL+10]. Although the pupil movement seems subtle from a third-person perspective, in the first-person perspective the gaze movement is enlarged: even a small change in the looking direction could be reflected saliently from the first-person view. Thus, here I also target the automatic modeling of first-person human gaze.

Meanwhile, different kinds of human behavior are not independent of each other but in fact deeply correlated. For example, human take actions based on what they perceive, and meanwhile, the actions alter the environment and thus will affect human perception behavior. To further promote the automatic modeling of human behaviors, it is necessary to consider not only one but multiple kinds of behaviors jointly. Keeping this in mind, in this thesis I also explore the mutual influence of human gaze behaviors and human actions.

1.1 Overview

Figure 1.2 illustrates the structure of this thesis. Under the first-person computer vision paradigm, this thesis work starts with the introduction to first-person videos and the motivation of modeling human behavior using first-person videos. Then this thesis introduces approaches developed to model various aspects of human behavior. Specifically this thesis is organized as the following chapters:

Chapter 2 presents a method for modeling human action in the form of action segmentation. Human action is one of the most explicit forms of human behavior. Other than its applications in augmented reality, human-computer interaction and surveillance, the ability to automatically modeling human actions from the first-person perspective can further be used to enable

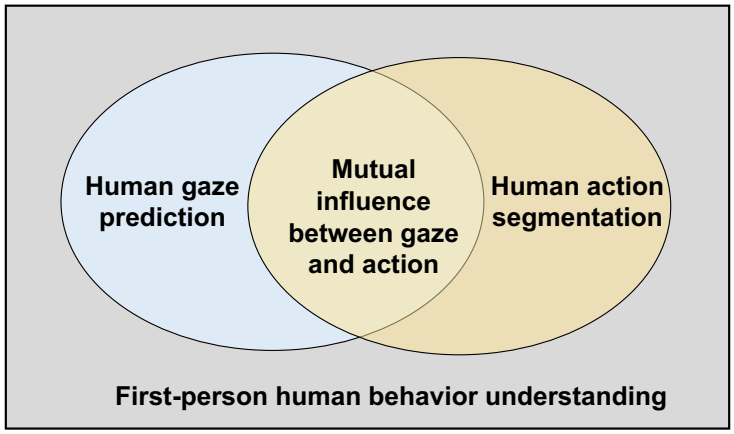

Figure 1.2: Structure of the thesis work. Human behavior modeling is studied under the first-person vision paradigm and plays a central role in this thesis. Human gaze behavior and human actions are separately explored, which is followed by a study of their mutual correlation.

the machine to make its own decisions by techniques such as reinforcement learning, which could pave the way for the future strong artificial intelligence. Although the automatic segmentation of human actions has been well studied, most previous works [CZ17, CHEGCN15] focus on analyzing the videos of third-person perspective, where actions could usually be clearly seen in the video. However as first-person videos naturally record what the camera wearer sees, the action taking place does not always happen within the visible portion of the video, due to the limited field of view and occlusions by unrelated objects. This prevents previous works to achieve optimal performance. Another underlying challenge is that first-person videos are usually natural life-logging videos which tend to be long and complex. To solve the aforementioned challenges, this chapter first describes the observation that we humans can pinpoint the action even if it is not directly visible by reasoning about the previous and the following actions. Based on this observation this chapter presents the idea of a graph-based reasoning module for refining the segmentation result. The module can be built on most existing models for action segmentation and is also computationally friendly when applied to long videos. Experiments on first-person video datasets prove that the proposed module can improve the performance of action segmentation when

applied on most backbone models.

Chapter 3 develops an approach for modeling human gaze behavior in the form of gaze prediction, which is the first ever work to automatically predict human's task-dependent attention transition. Human gaze allows us efficiently perceive the complex visual world by selectively attending to important parts of the scene. This phenomenon can reflect the inner state and intention of human [HUB+19] and thus is one essential part of human behavior. Since human naturally look at visually salient stimuli as it allows us to rapidly detect potential prey, predators, or mates in a cluttered visual world, most prior works use different cues of visual saliency such as color, contrast, and objectness for the automatic prediction of human gaze [PLN02]. Different from previous works, in this chapter I introduce a hybrid method for gaze prediction that incorporates not only visual saliency but also the task-dependent influence on human gaze. Experiments on multiple datasets demonstrate that this method can significantly outperform previous gaze prediction models by a large margin.

Chapter 4 extends the previously presented methods in Chapter 2 and Chapter 3 by jointly modeling human gaze and actions. In a natural dynamic scene, the human perception via gaze and the human actions are not isolated but deeply correlated: human actions alter the environment which requires gaze to dynamically change [Vic09], while human gaze highlights the important regions for more precise action recognition. Based on this observation, this chapter proposes to solve the coupled task of gaze prediction and action recognition jointly by a mutual context network, such that the knowledge of one task can improve the performance of the other. Using the proposed model trained in an alternative fashion, we can get improved performance in both tasks in multiple datasets, which strongly supports that the mutual influence of human gaze action could be leveraged for better human behavior modeling.

Chapter 5 first summarizes the contributions of this thesis and then offers discussion on the vision of future work. Other than improving the robustness

and efficiency of the existing works, this chapter further discuss the feasible applications and use cases for human behavior modeling in various scenarios. The vision of research theme to facilitate the applications is also presented in this chapter.

Finally, a chapter of publications concludes this thesis.

Chapter 2

Human Action and Human Reasoning Modeling from First-Person Perspective

2.1 Background

In the previous chapter, I introduced the background knowledge of first-person human behavior modeling. To begin with, in this chapter I introduce the method for modeling human actions from the first-person perspective. For human action modeling, it is important to know what action is happening in a video, as well as when the action is happening. The task of knowing when and what type of action is observed in a given video is called action segmentation. In this chapter, I introduce the method for action segmentation from the first-person perspective.

Because of the various potential applications such as human behavior analysis [VW87], anomaly detection [CBK09] and robot learning [KKUG07], the study of video action segmentation has attracted increasing research attention [RAAS12, SMJ+16]. Video action segmentation aims at both temporally locating each foreground action segment in the full untrimmed video and recognizing the action category of each segment. Since it can be seen as a combination of action localization [PCF19] and action recognition [TCSU08], most early approaches address this problem by first try to segment the video

into slices and then apply temporal classifiers on top of the video features of each slice. The methods for slicing the videos can be divided into several categories: The first type is sliding window [KSDB14]. These methods typically have very limited temporal receptive fields and thus cannot capture the full pattern of action and non-action. The second group of works designs segmental models [LRVH16, PR14] however these methods often fail in capturing the action patterns of a longer range since one action is only conditioned on its previous action. Another type of work makes use of the recent recurrent networks [SMJ+16, HFFN16]. These methods are proved to have only limited span of attention [SMJ+16] which could be harmful to performance. Recent works leverage temporal convolutional networks [LT18] to capture the long-range dependency within the frames of the video [DX18, LFV+17, FG19], and demonstrated promising results especially on the third-person videos taken from a fixed viewpoint.

However, if the video contains partially occluded actions or the action is beyond the video's field of view, it remains difficult for existing methods to perform well in such cases [ZAOT18]. For example, in Fig. 2.1 we showcase an example video from the EPIC-Kitchens dataset [DDMF+18]. This video is a first-person video and thus the field of view is changing. Nevertheless, we as human beings can easily infer the action after *take bottle* and *pour water* to be *drink water* by observing the sample images, although this drinking action is not directly shown. The major cause for this is our ability of reasoning: based on our observation that the camera wearer first takes the bottle to fill the glass and then puts down an empty glass after an up-and-down head motion, we can reason about the relation of actions to find out what happens in the middle. Without this reasoning ability, existing methods based on convolutional neural networks cannot perform well in these limited observation cases.

In this chapter, we target the task of modeling human action and human reasoning, by enabling the reasoning ability of machines for the task of action segmentation. As for the machine reasoning part, we use Graph Convolutional Networks (GCNs) [KW17, DBV16] as a key tool to perform reasoning. With the help of GCNs, we propose a novel model called Graph-based Temporal Reasoning Module (GTRM) that can be built on top of most existing

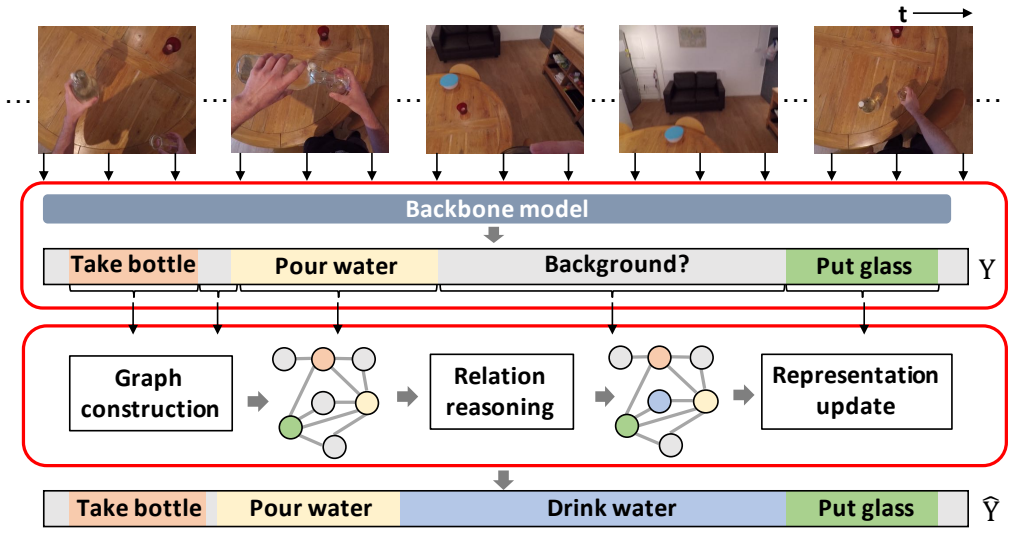

Figure 2.1: This figure showcase an example first-person video. Using the backbone model, since the action of *drink water* cannot be directly observed within the field of view, it detects the segment after *pour water* to be *background*. In this work, we propose a module called GTRM that can be built on top of the backbone model to refine the action segmentation. The result after refinement can successfully detect this segment to be *drink water*. This is because the proposed GTRM learns the temporal relation between the actions. In this figure, we also show that our proposed GTRM can adjust the temporal localization of actions by changing the action boundaries.

deep-learning based action segmentation models (we call it *backbone models* in the following of this chapter) to refine a better action segmentation result. The proposed GTRM learns to explicitly leverage the relation among multiple actions for the refinement of the initial action segmentation result of the backbone model. In the GTRM we first represent each segment as a node of the graph and then construct two types of graphs dedicated to refining the features of the nodes. The refinement is done by training the graphs on node classification task and the node temporal boundary regression task using GCN. One good property of this graph construction scheme is that since a node can represent an action segment of arbitrary length. This al-

lows the GCNs to operate on a flexible temporal receptive field, which makes it possible to model both short and long-range temporal relations.

We choose to use two publicly available dataset for validating the effectiveness of our proposed method: the EGTEA dataset [LLR18] and the EPIC-Kitchens dataset [DDMF+18], mainly based on two reasons. Firstly, compared with other datasets for action segmentation where the videos are captured from a fixed, third-person point of view, the videos from these two datasets are all first-person videos which makes the action segmentation problem a lot more challenging. The major challenge lies in the limitation of observation: the limited and moving field of view results in the invisibility of many actions. Also, severe occlusions caused by the camera wearer's hand or other interacting objects further aggravate the limitation in observation. Secondly, these datasets contain long videos ($e.g.>$ 30000 frames) and large numbers of action instances ($e.g.>$ 100) appear in one video. This severely affects the existing action segmentation models to function properly and accurately and also enlarged the demand for capturing long-range temporal relations between actions. Extensive experiments on the two datasets strongly demonstrate that our GTRM is capable of refining the performance of multiple backbone action segmentation models. We additionally show by experiments that our model cooperates better with recurrent backbones. To further test the usefulness of our GTRM, we also conduct experiments on general third-person datasets that are widely used for action segmentation, $i.e.$the 50Salads [SM13] dataset and the Breakfast [KAS14] dataset. The results demonstrate that our proposed model is also helpful on the general datasets, although the increase brought by our proposed model is not as large as that in first-person datasets. The main contributions of this chapter are summarized as follows: Firstly, this chapter is the first work that explicitly leverages the relations among more than two actions for the task of action segmentation. Secondly, we propose a novel method for modeling the relation and do the reasoning, by constructing graphs using the backbone output and applying GCNs for reasoning on the graph. The GCNs are trained to update the node representation based on the relations with its neighbors to predict a better action segmentation. Lastly, our experiments on two first-person datasets prove that our GTRM can strongly improve the

action segmentation result of several state-of-the-art backbone models. Our additional experiments on third-person datasets also supplement our claim.

The following content of this chapter is organized as follows: In Section 2.2 I introduce some related work. Section 2.3 offers the proposed graph-based method GTRM on action segmentation. Experiment evaluation of the proposed method is shown in Section 2.4. Finally, Section 2.5 concludes this chapter and points out some future directions.

2.2 Related Works

2.2.1 Action Segmentation

Action segmentation methods predict a dense action label at every frame of the video [LFV+17], which is different from action detection methods that focus on outputting only a sparse set of foreground actions. The topic of action segmentation has attracted much research attention [BKSS14, DX17, LFV+17, GYDD18] because of the potential segmentation range from human computer interaction to anomaly detection [CBK09]. Early work of action segmentation can be traced back to the line of work by Fathi *et al*. [FFR11, FRR11, FR13] that used the cue of object state changes. They design segmental models to ensure the temporal consistency between actions. After that, Cheng *et al*. [CFPC14] represent the video features as bag of visual words and use hierarchical Bayesian non-parametric model as a key tool for segmenting events in videos. One major drawback of the previously mentioned works is that the optimization of the segmental models is typically slow, which makes these works inappropriate for processing long videos. If it is safe to assume a strict ordering of actions, several works [KRG17, DX18, RKG17, SY18] focus on weakly supervised temporal action segmentation. But the assumption of ordering harms the generalization of the models.

Since frame-wise features can result in fragmented output results, many approaches apply classifiers with temporal smoothing. For example, some works [KGS16, TFFK12, VB14] used probabilistic models to refine the boundaries of the segmented actions. Recently, temporal convolution networks

(TCN) are proposed for action segmentation by Lea *et al*. [LFV+17] and have shown very promising results. Because of the large temporal receptive field brought by stacking temporal convolution layers, methods based on TCNs surpass the traditional sliding window based methods [KSDB14, RAAS12]. Lei *et al*. [LT18] further ensembles deformable convolution and residual connections on TCNs and achieves better performance. However, these two models [LFV+17, LT18] only work on a downsampled temporal resolution. Recently, Farha *et al*.proposed a modified version of TCN using dilated convolution kernels. They further stack multiple TCN blocks and showed that the dilated TCN with multi-stage refinement is able to capture longer range temporal dependencies [FG19]. The use of dilated convolution can avoid the video to be downsampled by temporal pooling operations and thus could operate on full temporal resolution. This method benchmarks the state-of-the-art performance in action segmentation of third-person videos. While remarkable progress has been made, all of the existing approaches are still difficult in capturing the actions without direct observation, since none of the existing methods have the relational reasoning ability to explicitly leverage the relations among more than two actions for action segmentation.

In this work, we address this problem by assembling the human reasoning ability into the task of action segmentation. We propose to use graph convolution network for reasoning by constructing the segmented video into two graphs, in which each action segment represents a node in the graph. With the graph representation, the node features can be fine-tuned based on their neighbors connected by the graph edges. Therefore, the relation among multiple actions is leveraged for a better action segmentation.

2.2.2 Graph Convolution Networks

Graph convolution networks (GCNs) are first proposed in [KW17] and soon dominate the research field of reasoning because of their effectiveness in modeling relation via their non-grid structures [LG18, LHZ+18]. After that, GCNs are widely applied in various computer vision topics such as image captioning [YPLM18], video action recognition [WZKX18, WG18, YXL18, ZTHS19, ZSXS19] and semi-supervised learning [LHW18]. For instance,

Pan *et al.* [PGZ19] used the natural graph structure of human joints and applied GCN for the task of action assessment. Zeng *et al.* [ZHT+19] proposed a model using GCN for finding more accurate action localization. Technically our method is inspired by these works. Our proposed GTRM exploit the reasoning ability of GCNs to explicitly model the temporal relations of multiple actions for improving video action segmentation.

2.3 Proposed Method

Given an untrimmed video containing a total of T frames, action segmentation targets to infer one action label for each frame. The ground-truth of frames can be represented as $Y^{gt} = \{\boldsymbol{y}_1^{gt}, \cdots, \boldsymbol{y}_T^{gt}\}$, in which each element $\boldsymbol{y}_t^{gt} \in \{0,1\}^C$ is given as an one-hot vector where the 1 in the vector indicates the human labeled action class. Including the background class, we denote C as the total number of classes. Our proposed module can be built on top of most backbone model for action segmentation for fine-tuning the output of the backbone model using graph-based temporal reasoning. We name our proposed Graph-based Temporal Reasoning Module (GTRM).

We will explain the details of our proposed GTRM in the following part of this section together with its training process and the implementation details. Before going into technical details of the model we would like to first describe the notations of the graph. In the following of this chapter, $\mathcal{G}(\mathcal{V}, \mathcal{E})$ denotes a graph with a set of N nodes $\mathcal{V}$. As for the weight of the edge that connects the nodes i and j, we denote it as $e(i,j) \in \mathcal{E}$.

2.3.1 Overview of the Model

We illustrate the architecture of our GTRM in Fig. 2.2. In the figure we show the backbone model as a three layer GRU, but it can be replaced by most existing models for action segmentation. The backbone model takes the frame-wise features $F = \{\boldsymbol{f}_1, \cdots, \boldsymbol{f}_T\}$ extracted by feature extractors like TSN [WXW+16] as input, and it will output a initial action segmentation result represented by frame-wise class predictions $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T\}$ where $\boldsymbol{y}_t \in [0,1]^C$. The input to our model is both the final output of the

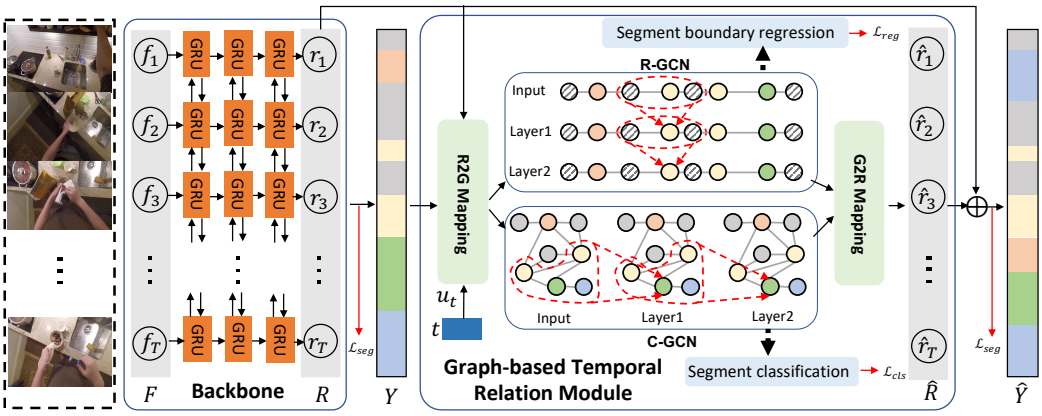

Figure 2.2: Illustration of our proposed Graph-based Temporal Relation Module (GTRM) built on top of a 3-layer GRU backbone model. Our GTRM construct graph nodes by mapping the backbone encoded representation of each segment in the initial segmentation. The two graphs have different types of edges and are respectively responsible for the segment boundary regression task and the segment classification task. After representation refinement by the GCNs, the node features are mapped back to frame-wise representations for an improved action segmentation.

backbone model Y and the frame-wise d-dimensional hidden representations $R = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_T\}$ encoded by the backbone model.

The proposed GTRM is inspired by the recent success of using GCNs for relational reasoning [ZHT+19, CRY+19, HGS19, HCK+17]. To learn the temporal relations of actions, our proposed module makes use of two GCNs called R-GCN and C-GCN. The nodes of the two graphs are identically created using the hidden representations R from the backbone model. We use the consecutive predictions in Y with the highest likelihood on the same action category to construct one node in the graph. The graph edges and loss on the graph training differentiate the two graphs by forcing the two graphs to learn different relations between the nodes. For R-GCN, a segment boundary regression loss is applied and a node classification loss is applied for C-GCN.

The two GCNs will refine the node representation by graph convolution operations. After the refinement, we use the nodes to form a new frame-wise

representation $\hat{R}$, and combine with the backbone representation R to predict a better frame-wise segmentation. There are in total three loss functions for the proposed framework. The first is the cross-entropy segmentation loss on the overall segmentation outputs. As stated before, there is one loss for each of the GCN so we jointly train the whole framework including the backbone model using the combined loss functions. We offer the details of the proposed GTRM in the following of this section.

2.3.2 Representation-to-Graph (R2G) Mapping

The outputs of the backbone model are the action class likelihood Y and the hidden representation R. Using Y and R for the graph construction is the key step in our proposed model. We aim to map the representation R of the backbone model to the graph nodes and refer this step Representation-to-Graph (R2G) mapping for simplicity. The number of nodes N is determined by the number of temporally ordered segments in Y. Let $(t_{i,s}, t_{i,e})$ represent the i-th action segment, where $t_{i,s}$ and $t_{i,e}$ respectively represent the timestamp of the starting and ending frames of this segment. Each segment of Y is summarized into one node in the graph and the node feature $\boldsymbol{a}_i$ is obtained by using pooling operation (max pooling used in this work) over the set of hidden representations corresponding to the action segment $\{\boldsymbol{r}_{t_{i,s}}, \cdots, \boldsymbol{r}_{t_{i,e}}\}$. In addition, since the temporal location of each segment contains useful information such as ordering, we also encode the time information to a d_t-dimensional vector $\boldsymbol{u}_i$ by feeding the time vector $(t_{i,s}, t_{i,e})$ to a multi-layer perceptron. The representation $\boldsymbol{x}_i$ for the i-th node is obtained by concatenating $\boldsymbol{a}_i$ and $\boldsymbol{u}_i$ in a channel-wise manner.

Defining fully connected graph edges to model the temporal relations of all action segments [WG18] can potentially result in noisy message passing between unrelated actions that are temporally far apart. To better address the action segmentation task, which essentially can be viewed as finding the class label and temporal boundary of all action instances including the background (no action), we construct different types of edges for the two graphs where the edges of R-GCN correspond to the boundary regression task and the edges of C-GCN to the classification task.

R-GCN The target task of the R-GCN is segment boundary regression, and its edges are defined to model the relation between neighboring segments which directly determine the temporal boundary (*i.e.*the start and the end frames) of the corresponding action segment. To this end, we only connect each segment with the segments right next to it by computing the temporal proximity between two segments. Defining $p(i, j)$ as the temporal proximity (inverse of the distance) between the middle frames of the i-th and j-th segment normalized by the length of the video, the edges $e_r(i, j)$ between the i-th and j-th nodes in R-GCN are defined as

$$e_r(i,j) = \begin{cases} p(i,j) & |i-j| \leq 1 \\ 0 & otherwise. \end{cases} \tag{2.1}$$

C-GCN In contrast, the target task of the C-GCN is segment classification, and the edges have to take into account the relations among multiple actions as they influence or condition on each other. For example, if we see a *take knife* action and then a *take potato* action, it is highly likely that a *cut potato* action will happen in the next few segments. We can infer the *cut potato* action even when the potato is occluded by leveraging such temporal relations. However, if two actions have a long temporal gap, they are unlikely to influence each other. Thus, we define edges $e_c(i, j)$ in C-GCN based on temporal proximity between the two nodes as

$$e_c(i,j) = \begin{cases} p(i,j) & |j-i| \leq 1, c_i \vee c_j = bg \\ p(i,j) & |j-i| \leq k, c_i \neq bg, c_j \neq bg \\ 0 & otherwise, \end{cases} \tag{2.2}$$

where bg represents the background class where no action happens. In other words, each background node is linked only to its nearest neighbors, while each of other nodes is also linked to k neighboring nodes.

Reasoning on Graphs

In both GCNs, all of the edge weights form the adjacency matrix $\boldsymbol{A}_c$ or $\boldsymbol{A}_r$ with $N \times N$ dimensions. Following [WG18], we normalize the adjacency

matrix by using the softmax function as

$$\boldsymbol{A}(i,j) = \frac{\exp g(i,j)}{\Sigma_{j=1}^{N} \exp g(i,j)}. \tag{2.3}$$

For reasoning on the graphs, we perform M-layer graph convolution for refining the node representation. Graph convolution enables message passing based on the graph structure, and multiple GCN layers further enable message passing between non-connected nodes [KW17]. In an M-layer GCN, the graph convolution operation of the m-th layer ($1 \leq m \leq M$) could be represented as

$$\boldsymbol{X}^{(m)} = \sigma(\boldsymbol{A}\boldsymbol{X}^{(m-1)}\boldsymbol{W}^{(m)}), \tag{2.4}$$

where $\boldsymbol{X}^{(m)}$ are the hidden representation of all the nodes with $N \times d_m$ dimensions at the m-th layer. $\boldsymbol{W}^{(m)}$ is the weight matrix of the m-th layer, and σ denotes the activation function. Following prior work [WG18], we apply two activation functions namely Layer Normalization [BKH16] and ReLU after each GCN layer. After the graph convolution operations, we obtain updated node representations $\hat{\boldsymbol{x}}_i^c$ and $\hat{\boldsymbol{x}}_i^r$ for nodes in the C-GCN and R-GCN, respectively.

We apply an FC layer on each node after the final GCN layer to perform segment classification on the C-GCN and segment boundary regression on the R-GCN. This operation is also known as *readout* operation [QWJ+18, WLS+19] as it maps the refined node representation to the desired output. The output of each C-GCN node is the class likelihood $\hat{\boldsymbol{c}}_i$ for the corresponding segment. Following previous works on boundary regression [RHGS15, GYN17], the output of each node in R-GCN is an offset vector $\hat{\boldsymbol{o}} = (\hat{o}_{i,c}, \hat{o}_{i,l})$ relative to the input segment. $\hat{o}_{i,c}$ is the offset of the segment center (normalized by the length of the segment), and $\hat{o}_{i,l}$ is the offset of the length of a segment in log scale. Given these offsets, it is trivial to compute the predicted boundary $\hat{t}_{i,s}, \hat{t}_{i,e}$.

2.3.3 Graph-to-Representation (G2R) Mapping

After the graph convolution operations, the representation of each node is updated by information propagation from its neighboring nodes. To perform

action segmentation based on the updated representations, we inversely map the updated graph node representations to frame-wise representations $\hat{R} = \{\hat{\boldsymbol{r}}_1, \cdots, \hat{\boldsymbol{r}}_T\}$. We fuse the representations from two GCNs via node-wise summation, and then reconstruct $\hat{\boldsymbol{r}}$ by mapping the node representation to all of the corresponding frames:

$$\hat{\boldsymbol{r}}_t = \hat{\boldsymbol{x}}_i^c + \hat{\boldsymbol{x}}_i^r, \ \forall t \in \{\hat{t}_{i,s}, \cdots, \hat{t}_{i,e}\}, \tag{2.5}$$

where $\hat{t}_{i,s}, \hat{t}_{i,e}$ are the temporal starting and ending frames of the i-th segment predicted by the R-GCN. Similarly to previous work [ZXW+17, ZHT+19], we concatenate $\hat{\boldsymbol{r}}$ with the original latent representation $\boldsymbol{r}$ from the backbone model for obtaining the final action segmentation results. We apply a 1×1 convolution layer on the concatenated representation followed by softmax as activation function to obtain the final frame-wise action likelihood $\hat{\boldsymbol{y}}$.

2.3.4 Training and Loss Function

We train the whole network including both the backbone model and our GTRM using a combination of multiple loss functions. As for the action segmentation outputs $\boldsymbol{y_t}, \hat{\boldsymbol{y}}_{\boldsymbol{t}}$, we apply the same loss function as [FG19] which is a combination of cross entropy loss $\mathcal{L}_{\text{cls}}$ and a truncated mean squared error $\mathcal{L}_{\text{t-mse}}$ designed to punish local inconsistency by encouraging adjacent predictions to be similar:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{cls}} + \lambda_t \mathcal{L}_{\text{t-mse}}. \tag{2.6}$$

We use the same cross entropy loss $\mathcal{L}_{\text{cls}}$ for C-GCN. The ground truth action category of a segment is defined by the category of the closest ground truth segment measured by temporal intersection over union (tIoU).

For R-GCN, we use smooth L1 loss as the regression loss $\mathcal{L}_{\text{reg}}$. Similarly with the C-GCN, the ground truth time information of a node is defined by the temporally closest segment to this node. Denote $t_{i,c} = (t_{i,s} + t_{i,e})/2$ and $t_{i,l} = t_{i,e} - t_{i,s}$ as the center and length of a segment, respectively, the ground truth offset $\boldsymbol{o}_i^{gt} = (o_{i,c}^{gt}, o_{i,l}^{gt})$ could be represented as:

$$o_{i,c}^{gt} = (t_{i,c} - t_{i,c}^{gt})/t_{i,l}, \quad o_{i,l}^{gt} = \log(t_{i,l}/t_{i,l}^{gt}), \tag{2.7}$$

The combined loss function thus can be defined as

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{T} \mathcal{L}_{\text{seg}}(\boldsymbol{y}_i^{gt}, \boldsymbol{y}_i) + \sum_{i=1}^{T} \mathcal{L}_{\text{seg}}(\boldsymbol{y}_i^{gt}, \hat{\boldsymbol{y}}_i) \\ &+ \lambda_1 \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(\boldsymbol{c}_i^{gt}, \hat{\boldsymbol{c}}_i) + \lambda_2 \sum_{i=1}^{N} \mathcal{L}_{\text{reg}}(\boldsymbol{o}_i^{gt}, \hat{\boldsymbol{o}}_i).\end{aligned} \tag{2.8}$$

2.3.5 Implementation Details

We implement our model using the Pytorch [PGC+17] library. We choose to use $d = 64$ as the dimension of hidden representations. The multi-layer perceptron for encoding the time representation $\boldsymbol{u}_t$ is a fully connected layer with sigmoid activation and 16 output channels. We use 2 layer GCNs in all of our experiments, since we do not observe obvious performance increase when adding more layers.

We adopt the Adam optimizer [KB14] with its default hyper-parameters for training the proposed framework. The training process can be described as follows: we first initialize the backbone model for 50 epochs with learning rate 5×10^{-4}, without other parts of the model. We then fix the backbone and train the GTRM for 50 epochs. After this, we finetune the whole network for 100 epochs with a reduced learning rate of 1×10^{-4}. More details about training can be found in the supplementary material. In all experiments, we set $\lambda_t = 0.15, \lambda_1 = \lambda_2 = 0.5$ for loss functions.

2.4 Evaluation

In this section, we compare the performance of our proposed module built on top of state-of-the-art models on challenging large-scale first-person datasets. We also conduct ablation studies to examine the impact of each part of our model. To further understand our we examine the performance of our GTRM when built on top of existing backbone models on more general third-person datasets.

Datasets Figure 2.3 compares different commonly-used video datasets based on average action instances per video and average video length (in minutes),

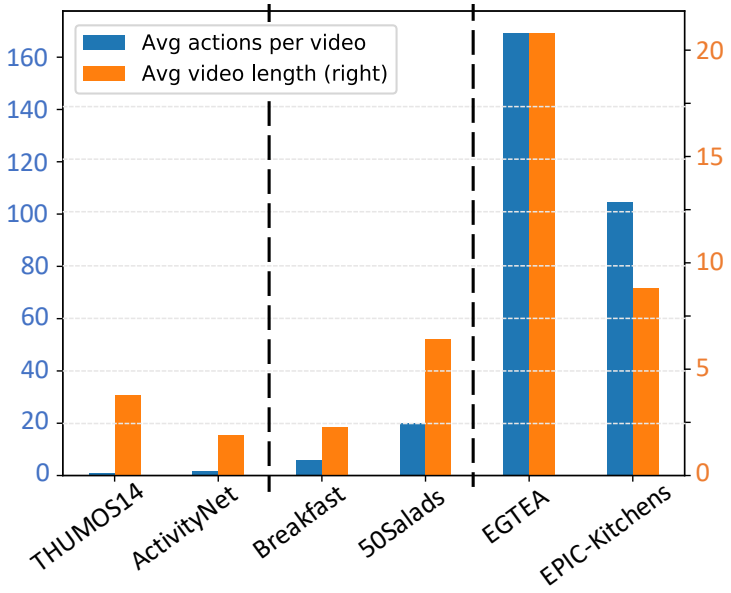

Figure 2.3: Dataset comparison by average action instances per video (blue) and average video length (orange, right axis).

in which we divide them into three groups. The leftmost group are the THUMOS14 [IZJ+17] and ActivityNet [CHEGCN15] dataset. These datasets contain one or two action instances per video and are usually used for the task of action proposal [LLL+19], localization [PCF19] or detection [MJY+19, XGC+19]. The Breakfast [KAS14] and 50Salads [SM13] dataset contain less than 20 actions per video, and are the standard datasets for evaluation of action segmentation methods [LT18]. The rightmost group contains two recent large-scale datasets containing natural daily living activities from an first-person perspective, EGTEA [LLR18] and EPIC-Kitchens [DDMF+18]. Due to the unique perspective of egocentric recording, the actions sometimes happen out of the camera's field of view (*e.g.*in Fig. 2.1), or critical informative region is occluded by the hand. These characteristics make many actions in EGTEA and EPIC-Kitchens not directly observable and they have to be inferred from temporal relations. In the following sections, we mainly conduct the experiments on these two datasets, while later we also show experimental results on the Breakfast and 50Salads datasets.

Evaluation Metrics For evaluating our model, we adopt several evaluation metrics commonly used in action segmentation [LFV+17, LT18, FG19]: frame-wise accuracy, segmental edit score, and the segmental F1 score at overlapping thresholds $\tau/100$ denoted by F1@τ. Frame-wise accuracy is one of the most widely used metrics for evaluation of action segmentation. However, long actions tend to have a higher impact on this metric, while there is no strong penalty on over-segmentation. In contrast, segmental edit score and F1 score are evaluation metrics presented in [LRVH16, LFV+17] and penalize over-segmentation errors. The segmental edit score penalizes the case of over-segmentation, and the segmental F1 scores measure the quality of the prediction.

2.4.1 Comparison with the State of the Art

In this section, we compare our model with several state-of-the-art models on EGTEA and EPIC-Kitchens datasets (Table 2.1). The EGTEA dataset contains 86 videos and has a total length of 29 hours. We focus on the segmentation of the 19 action classes (*i.e.*verbs). For the EGTEA dataset we perform a four-fold cross validation by randomly splitting the videos into four partitions. The EPIC-Kitchens dataset contains 55 hours of daily living non-scripted activities with 125 classes of actions. Since the ground-truth labels of the test set are not publicly available, we follow [BNW+18] to split part of the training set as train-test set. The video features for EGTEA and EPIC-Kitchens are extracted by using I3D pretrained on Kinetics dataset [CZ17]. We down-sample the videos to 15 fps.

We use four closely related methods as baseline models. **FC** is a simple baseline that directly add a frame-wise classifier on the I3D-extracted features. **Bi-LSTM** [SMJ+16] is a bi-directional temporal LSTM for action segmentation. **EDTCN** [LFV+17] and **MSTCN** [FG19] are two of the recent competitive models using temporal convolution networks to capture long term frame dependencies.

We also include our own backbone using multi-layer GRU (**m-GRU**) in the comparison. We report the performances of our GTRM built on top of different backbone networks, by adding "+GTRM " as the notation. Since

EGTEA	**F1**@{10,25,50}			**Edit**	**Acc**
FC [CZ17]	8.7	6.7	3.1	9.4	65.4
Bi-LSTM [SMJ+16]	27.0	23.1	15.1	28.5	70.0
EDTCN [LFV+17]	31.1	27.7	19.6	28.6	70.1
MSTCN [FG19]	32.1	28.3	18.9	32.2	69.2
m-GRU	32.6	27.7	17.6	36.0	67.1
Bi-LSTM+GTRM	33.3	29.2	19.9	32.1	**70.7**
EDTCN+GTRM	34.6	31.2	20.7	34.8	70.1
MSTCN+GTRM	36.6	29.7	18.6	32.2	68.4
m-GRU+GTRM	**41.6**	**37.5**	**25.9**	**41.8**	69.5

EPIC	**F1**@{10,25,50}			**Edit**	**Acc**
FC [CZ17]	9.3	5.6	2.2	20.0	42.2
Bi-LSTM [SMJ+16]	19.0	11.7	5.0	29.1	43.3
EDTCN [LFV+17]	21.8	13.8	6.5	27.3	42.9
MSTCN [FG19]	19.4	12.3	5.7	25.3	43.6
m-GRU	20.2	15.2	7.7	30.5	40.3
Bi-LSTM+GTRM	25.1	17.3	8.8	35.9	43.5
EDTCN+GTRM	24.2	15.9	7.2	33.1	42.8
MSTCN+GTRM	24.4	15.4	7.2	32.5	**43.7**
m-GRU+GTRM	**31.9**	**22.8**	**10.7**	**42.1**	43.4

Table 2.1: Quantitative comparison with state-of-the-art models on the EGTEA dataset (left) and EPIC-Kitchens dataset (right).

no previous results on EGTEA and EPIC-Kitchens datasets are available for baseline models, all the reported results are based on our implementation.

As can be seen from and Table 2.1, comparing our model with the backbone models (without adding our GTRM), our model outperforms backbone models by a large margin on F1 score and edit score, while performing comparably well with respect to the frame-wise accuracy metric. The lower parts of Table 2.1 summarize the performance of our proposed GTRM when built on top of different backbones. As can be seen, the performance of all backbone models mostly increases by adding GTRM, except the F1@50 and accuracy of MSTCN in the EGTEA dataset. This shows that our GTRM is capable of refining the backbone results in most cases. Interestingly, we find that the gain of adding our GTRM is the largest with recurrent backbone models (Bi-LSTM and m-GRU). This is possibly because the recurrent backbones have a smaller span of attention, while our GTRM can work complementary since the reasoning is performed with a larger temporal receptive field.

From the qualitative comparison in Fig. 2.4 (a), we can see that the "take", "put" and "close" actions are correctly detected by adding our GTRM. Especially, due to the viewpoint limitation, the "close (fridge)" action is almost not observable in the video (since the camera wearer quickly turns his attention to the location of the next step). The fact that this action is being correctly detected by our model strongly supports our claim that our GTRM can capture the relation of actions (as there is an "open (fridge)" action happened before) for better action segmentation. On the other hand, we can also

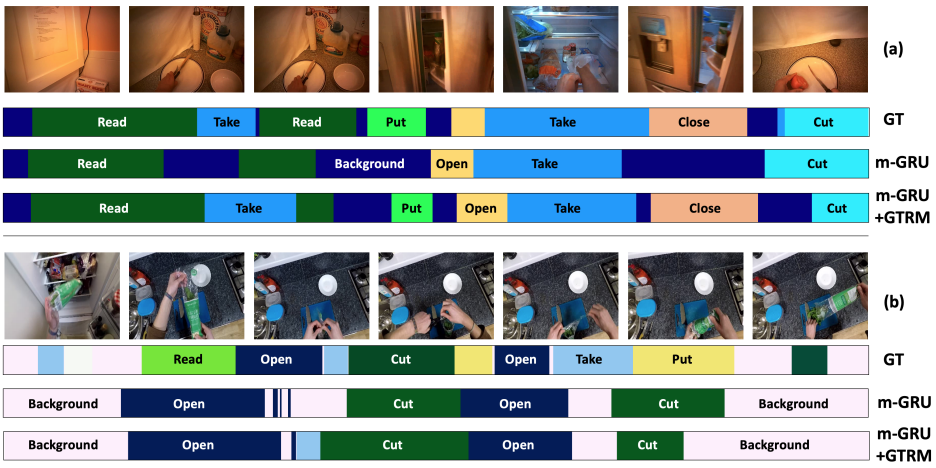

Figure 2.4: Qualitative comparison of results for action segmentation task on (a) EGTEA, and (b) EPIC dataset. Only part of the whole video is shown for clarity. We can see in (a) that the *take*, *put* and *close* actions are correctly detected by adding GTRM.

see weakness of our model in Fig. 2.4 (b) is that our GTRM depends on the initial backbone output. The backbone model could not detect the "read" action, and the "take", "put" actions are predicted as a single "cut" action. Conditioning on this output, it is still difficult for our GTRM to correctly identify those actions. More qualitative results of different backbone models with and without our proposed GTRM are in the supplementary material.

2.4.2 Ablation Studies

To fully understand the effect of each component of our model, we conduct ablation studies on the EGTEA dataset by changing or deleting part of our model and compare their performances. We first examine the impact of each of the graphs in our model. For fair comparison, we replace each of the C-GCN and R-GCN with a small 2 layer fully connected network (denoted by FCN). In this case, each graph node is processed individually by the FCN without considering the relations brought by the graph edges. We also examine the usefulness of time vector $\boldsymbol{u}_t$. Table 2.2 shows the *relative performance gain* compared with using the m-GRU backbone alone. In the table, C-GCN

Gain	**F1**@{10,25,50}			**Edit**	**Acc**
C-GCN + FCN (w/o $\boldsymbol{u}_t$)	4.6	4.4	1.8	4.4	2.5
FCN only	6.2	6.1	4.7	4.5	3.3
R-GCN + FCN	6.8	6.8	4.8	6.0	2.1
C-GCN + FCN	6.4	6.0	4.7	3.5	2.7
C-GCN + R-GCN	10.0	9.8	6.8	7.5	2.8

Table 2.2: Ablation study of our model. We replace GCN with fully connected network (FCN) and report the performance gain in absolute values relative to the m-GRU backbone model.

+ FCN is the case where R-GCN is replaced by the fully connected network and others follow the same rule. We can see that the performance using GCN in general favors than that without GCN, which validates the usefulness of using relations between actions for action segmentation. Additionally, we find that the time vector $\boldsymbol{u}_t$ provides necessary information to the network as adding $\boldsymbol{u}_t$ improves the performance while without $\boldsymbol{u}_t$ the task for boundary regression cannot converge.

We also investigate the selection of parameter k, which is related to the number of neighbors for each segment to aggregate information from. We variant the value of k and show the experiment result on EGTEA dataset in Fig. 2.5. Overall, the best performance is achieved with $k = 8$, while the performance gain decreases starting from $k = 16$. We suspect this is because of irrelevant information propagation through the edges by connecting the action segments that are too temporally distinct. Further ablation studies on the influence of edge weight and tools for modeling relation (*e.g.*1D convolution on nodes) can be found in the supplementary material.

2.4.3 Results on Third-Person Datasets

To test the effectiveness of our proposed model on other general cases, we also test our model performance on the 50Salads [SM13] and Breakfast [KAS14] datasets. The 50Salads dataset contains 50 videos of salad making activities with 17 action classes. We follow [SM13] to use a 5-fold cross validation

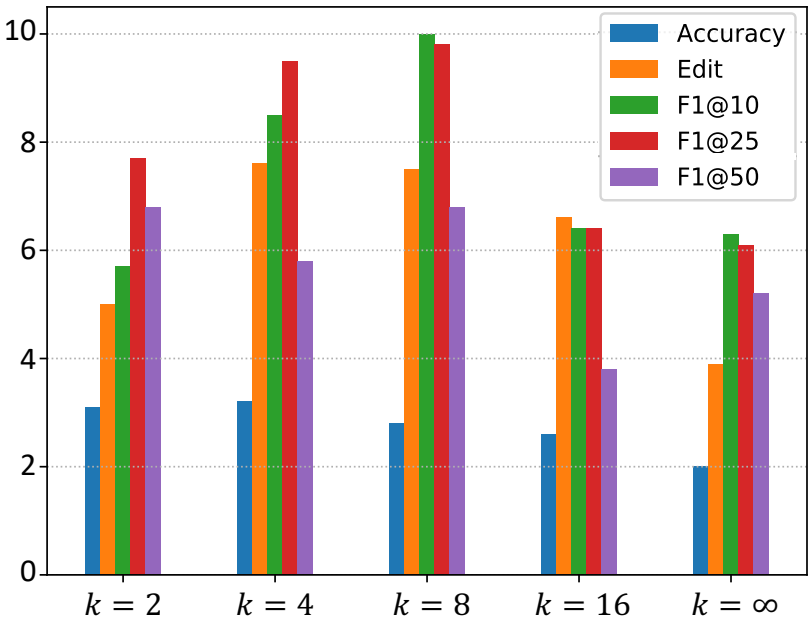

Figure 2.5: Performance gain compared with the m-GRU backbone model with different values of k. $k = \infty$ denotes the case that all nodes are connected.

and report the average performance. The Breakfast dataset contains 1712 videos with a total length of 65 hours. There are 48 different actions while on average 6 actions per video. We use the standard 4 splits [KAS14] and report the average. For a fair comparison, we adopt the features from [FG19] in the following experiments.

We build our GTRM on top of the current state-of-the-art approach MSTCN [FG19]. Since MSTCN is based on temporal convolution networks, we further test the model performance combined with a recurrent backbone Bi-LSTM [SMJ+16]. The performance comparison on 50Salads dataset is shown in Table 2.3, including both the result reported in [FG19] and result with our implementation. Since there are on average 20 actions per video, we adjust the parameter k to be 4. As can be seen, the performance of both backbone models got improved by adding our GTRM. While the performance gain of the MSTCN backbone is relatively marginal, the gain of Bi-LSTM backbone is still significant. This phenomenon is the same as observed in the

50Salads	**F1**@{10,25,50}			**Edit**	**Acc**
MSTCN [FG19]	76.3	74.0	64.5	67.9	80.7
MSTCN(our impl.)	73.4	71.0	61.5	67.2	80.2
MSTCN+GTRM	75.4	72.8	63.9	67.5	82.6
Gain	2.0	1.8	1.4	0.3	2.4
Bi-LSTM [SMJ+16]	62.6	58.3	47.0	55.6	55.7
Bi-LSTM (our impl.)	62.2	61.3	53.7	53.5	70.1
Bi-LSTM+GTRM	70.4	68.9	62.7	59.4	81.6
Gain	8.2	7.6	9.0	5.9	11.5

Table 2.3: Results on the 50 Salads dataset. Performance gain in absolute values by adding our GTRM on top is shown in dark rows.

EGTEA dataset, which shows that our GTRM works better with recurrent backbones.

Since there was no previously reported results from Bi-LSTM, we only use MSTCN as the backbone model for the Breakfast dataset. The performance is summarized in Table 2.4. The breakfast dataset only contains 6 action instances per video, far less than the 50Salads dataset. Similarly with the 50Salads dataset, the performance gain is relatively marginal. Also, modeling relations among more neighbors by increasing k does not improve the segmentation performance.

There could be mainly two reasons why the benefit of our GTRM is limited on these two datasets. Firstly, as the 50Salads and Breakfast dataset are taken from a fixed view camera capturing most of the human activities, there are less cases of unobservable actions due to, *e.g.*, occlusions. Secondly, the number of action instances is relatively small so that temporal patterns can be captured to some extent by only using the backbone model.

2.4.4 Comparison with 1D Convolution

In our GTRM, we use GCN as a way to perform reasoning on the graph structure. An alternative approach to perform reasoning on the temporal re-

Breakfast	**F1**@{10,25,50}			**Edit**	**Acc**
MSTCN [FG19]	52.6	48.1	37.9	61.7	66.3
MSTCN (our impl.)	57.3	53.4	41.4	58.8	60.0
MSTCN+GTRM ($k = 2$)	57.5	54.0	43.3	58.7	65.0
Gain	0.2	0.6	1.9	-0.1	5.0
MSTCN+GTRM ($k = 4$)	57.3	53.6	42.9	58.5	63.8
Gain	0.0	0.2	1.5	-0.3	3.8

Table 2.4: Result on the Breakfast dataset. Performance gain in absolute values by adding our GTRM on top is shown in dark rows.

lation of segments is to perform 1D convolution on the sequence of segments. While it can increase the computational cost and potentially aggregate irrelevant information (from, *e.g.*, background segments containing no action), 1D convolution can still gather information from the neighbourhood nodes for updating the hidden representation of each node.

EGTEA	**F1**@{10,25,50}			**Edit**	**Acc**
m-GRU	32.6	27.7	17.6	36.0	67.1
C-conv + R-conv	41.5	35.9	24.2	40.7	68.2
C-conv + R-GCN	41.4	35.7	23.2	41.2	68.2
C-GCN + R-conv	**41.7**	36.3	24.2	**42.4**	67.9
C-GCN + R-GCN	41.6	**37.5**	**25.9**	41.8	**69.5**

Table 2.5: Changing GCN operation to 1D convolution.

In this experiments, we replace either of R-CGN and C-GCN with 1D convolution and compare the performances on the EGTEA dataset. Results are shown in Table 2.5. **C-conv + R-conv** corresponds to the model where both C-GCN and R-GCN are replaced with 1D convolution. **C-conv + R-GCN** and **C-GCN + R-conv** denote the cases where either the C-GCN or R-GCN is replaced with 1D convolution, respectively. **C-GCN + R-GCN** corresponds to our originally reported result.

Despite the fact that 1D convolution increases the computational complexity especially with larger kernel sizes, the overall performance change remains small. This illustrates the benefit of using GCN rather than 1D convolution on nodes for reasoning the temporal relations of the action segments.

2.4.5 Comparison with Other Alternatives Tools for Modeling Relation

Similarly with the previous section, the GCNs in our GTRM could be also changed to other alternatives (*e.g.*LSTM). In this section we discuss the effect on result and computational cost when GCNs are replaced by other alternatives. These include bi-LSTM [SMJ+16], dilated convolution (dil.-conv) [FG19] and 1D-convolution (1D-conv) introduced above. We also compare the number of parameters, number of added parameters, and the FLOPs added for each alternative.

variants	**F1**@{10,25,50}			Params	Δ Params	Δ FLOPs
None	32.6	27.7	17.6	1.53M	0	0
LSTM	28.1	18.0	1.95M	0.42M	18%	
bi-LSTM [SMJ+16]	33.7	28.5	17.9	2.38M	0.85M	23%
dil.-conv [FG19]	39.8	34.0	22.6	2.50M	0.97M	20%
1D-conv	41.5	35.9	24.2	3.79M	2.26M	47%
GTRM	**41.6**	**37.5**	**25.9**	1.85M	0.32M	8%

Table 2.6: Number (and added number) of parameters, added FLOPs of different variants of GTRM on the EGTEA dataset.

Table 2.6 shows the number of parameters and its increase from the baseline (None), together with the increase in FLOPs. Overall, GCN not only achieves the best performance but is also significantly faster and requires much fewer parameters than other methods such as 1D convolution and recurrent network. As shown in the table, our GTRM uses much fewer parameters and is significantly faster than the best baseline (1D-conv). All the

other baselines mentioned in the previous paragraph also take up much more ($\sim 3\times$ more) computational time than our proposed model (except FCN).

2.4.6 Influence of Edge Weighting

In this section we discuss the influence of different edge weight designs in R-GCN and C-GCN. While in our GTRM edges are weighted according to the temporal distances, it is also possible to use uniform weights on each edge which still enables message passing between connected nodes.

We replace the edge weight of either C-GCN or R-GCN to uniform weighting and compare the performance on the EGTEA dataset. Experimental results are summarized in Table 2.7. **C-Uni+R-Uni** is the case where both GCNs use uniform weight. **C-Uni+R-GCN** and **C-GCN+R-Uni** indicates the cases where only edges in C-GCN or R-GCN is changed to uniform weight, respectively. **C-GCN+R-GCN** is our reported result.

Gain	**F1**@{10,25,50}			**Edit**	**Acc**
m-GRU	32.6	27.7	17.6	36.0	67.1
C-Uni + R-Uni	39.6	34.2	23.2	41.3	67.4
C-Uni + R-GCN	40.9	35.9	24.8	41.4	67.4
C-GCN + R-Uni	40.6	35.8	24.1	41.2	67.4
C-GCN + R-GCN	**41.6**	**37.5**	**25.9**	**41.8**	**69.5**

Table 2.7: Changing edge weight to uniform weight.

We can see that, even with the uniform edge weights, the performance is far better than the baseline m-GRU. This demonstrates the importance of modeling temporal relations among action segments for a better action segmentation. The performance is further improved with our proposed edge weighting scheme, and supports the effectiveness of the distance-based weight design.

2.4.7 Qualitative Results

In this section, we show more qualitative results on the Breakfast dataset (Fig 2.6) and the 50Salads dataset (Fig 2.7). In both figures, we show the ground truth segmentation results in the first row (a), results of MS-TCN in the second row (b), and the result of MS-TCN with our proposed GTRM on top in the third row (c). We can see that in these two datasets, while adding our proposed GTRM on top of MS-TCN improve the overall result, the influence is relatively small.

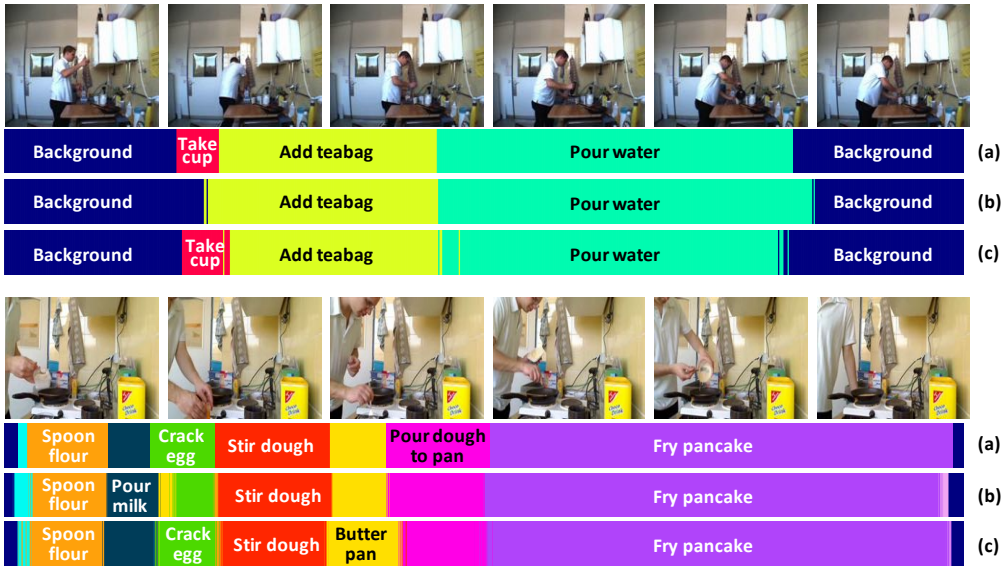

Figure 2.6: Qualitative results on the Breakfast dataset. Result is shown in full video length. Corresponding results are: (a) Ground truth, (b) MS-TCN, (c) MS-TCN + GTRM.

2.4.8 Limitations and Future Work

As discussed in Section 2.4.1, one of the limitations of our model is that it relies on the backbone model. If the backbone model output a poor result, our model can only slightly improve the segmentation performance.

Another limitation is that, if the backbone outputs are heavily frag-

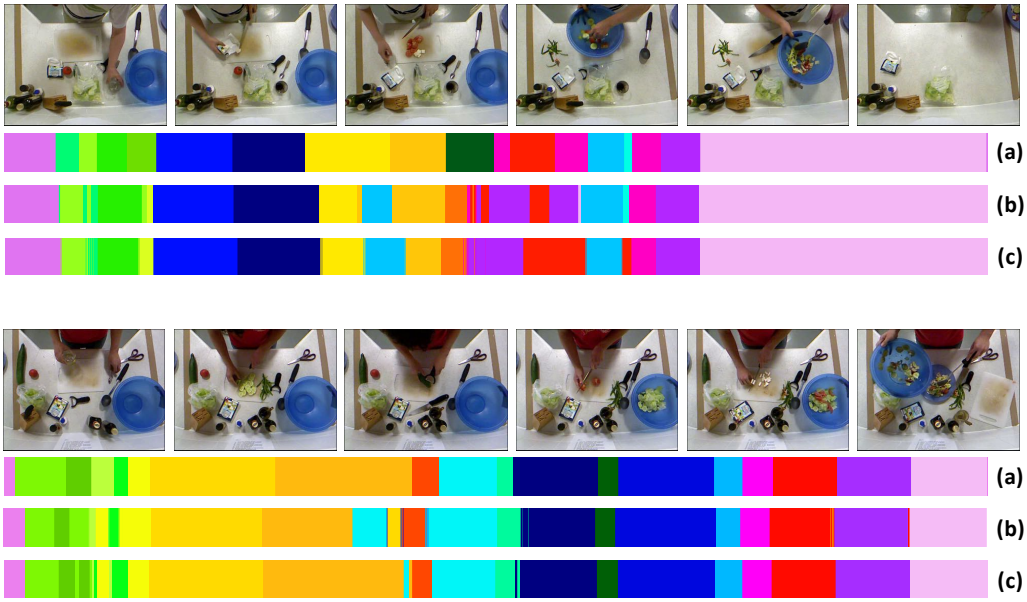

Figure 2.7: Qualitative results on the 50Salads dataset. Result is shown in full video length. Corresponding results are: (a) Ground truth, (b) MS-TCN, (c) MS-TCN + GTRM.

mented, the constructed graph would be large and the optimization becomes very inefficient. This also prevents us from building our model on top of the FC baseline. While it is possible to filter the action segments and ignore the small segments in the graph construction step, it is still an important future work to examine approaches to process the graph convolution in a more efficient way. Using additional information like eye-tracker or depth sensor, or using techniques such as adaptive sampling [HZRH18] or stochastic training [CZS17], will be promising candidates for future investigation.

2.5 Conclusion

This chapter aim to enhance the automatic modeling of human actions in long and complex first-person videos. In this chapter, we presented a novel approach for modeling action relations aiming at the task of action segmentation which can be built on top of most existing neural networks for action

segmentation. To model the temporal relations, we construct two graphs and use GCNs to perform reasoning on the graphs based on two different criteria. After updating the node representations, they are mapped back to individual frames as an updated representation for final action segmentation. Extensive experiments demonstrate that our model can effectively learn to use relations for better action segmentation, and demonstrated performance improvements brought by our model.

There still exists a lot of work to do for human behavior modeling. I will address the human attention modeling and explore the correlation of human gaze and actions in the next chapters.

Chapter 3

Human Attention Modeling from First-Person Perspective

In the previous chapter, I introduced the modeling of human action and human reasoning from the first-person perspective. Since humans take actions based on their perception of the surrounding environment, the way of humans perceive the world is also an important perspective of human behavior. If the knowledge of human perception can be transferred to the machines, it could enable a lot of applications such as efficient computation and medical robots. In this chapter, I focus on the modeling of human attention via the task of first-person gaze prediction. Since gaze is one of the primary forms of human attention and could be measured by eye-trackers, it is a good early step toward understanding the general human attention. In this chapter, I introduce my work that uses the cue of attention transition for the task of first-person gaze prediction.

3.1 Background

With the cameras and batteries becoming more and more portable, wearable cameras are receiving increasing popularity in the past few years. Accompanied by the increase of wearable cameras, the demand for automatic analysis of videos captured from a first-person perspective is also rising, which makes first-person (or egocentric) vision an emerging field in computer vision. One

specific characteristic of first-person vision is the modeling of the camera wearer's point-of-gaze, as gaze contains not only important information about the objects being interacted with but also the intent of the camera wearer. The automatic modeling of the human gaze can be used to infer important regions in images and videos for reducing the computational cost in the learning and inference procedures of various applications [FLR12, XML+15].

Given a first-person video, the goal of this chapter is to develop a computation model for predicting the camera wearer's point-of-gaze, *i.e.*, gaze prediction. Gaze prediction has been extensively studied during the past few decades, but most previous works formulated it as a saliency detection problem to find image or video regions that are likely to attract human attention. This saliency-based paradigm is based on the nature that the human visual system are naturally evolved to be sensitive to visually salient regions such as color, motion, or contrast, such that humans can easily find the predators or preys. However, it is hard for the saliency-based gaze prediction models to generalize to natural dynamic scenes, *e.g.* cooking in a kitchen, where high-level knowledge of the task has a strong influence on human attention.

Humans use series of gaze fixations to perceive the surrounding environment in a natural dynamic scene. Since the gaze naturally points to the objects/regions related to human interactions, the transition of human attention is profoundly related to the undergoing task. Especially in object manipulation scenarios, the task carried out by the camera wearer will determine a certain flow of objects or places being attended on so there will be a certain pattern of attention transition. For instance, to take a bottle of water out of the fridge, a person would first fixate at the door of the fridge, opens the fridge door, and then change the attention to the bottle in the fridge so as to grasp it. Therefore, we argue that to achieve a more accurate gaze prediction, it is necessary to explore the use of task-dependent patterns in attention transition.

In this chapter, we propose a hybrid deep-learning based model for the task of human gaze prediction to fuse the information of both bottom-up visual saliency and task-dependent attention transition learned from consecutively attended image regions. Three modules contribute to the proposed model: (1) The first module is a two-stream Convolutional Neural Network

(CNN) that generates saliency maps directly from video frames in a bottom-up fashion, and (2) the second module aims to generate gaze prediction in a top-down manner using the information from task-related attention transition. It includes a recurrent neural network and a fixation state predictor and generates an attention map for each frame based on the previously fixated regions and head motion. It is built based on two assumptions. The first assumption is that a person's gaze tends to be established on the same object during each fixation, and a large gaze shift is almost always accompanied with a large head motion [Lan04]. The second assumption is that the patterns in the temporal shift between regions of attention can be learned from training data since they are dependent on the performed task. (3) The last module is a small fully convolutional network to fuse the output of the previous two modules and generate a final gaze prediction map, from which the final prediction of 2D gaze position is made.

The main contributions of this chapter are: (1) To the best of our knowledge, this is the first model that leverages both bottom-up visual saliency and task-dependent attention transition for the task of gaze prediction from first-person videos. (2) We propose a novel recurrent network based module for learning the patterns of attention transition: the temporal shift of gaze fixations. This attention transition module can be used alone to predict the region of attention based on the video and the previous fixations. (3) The proposed approach achieves state-of-the-art gaze prediction performance on multiple public first-person activity datasets.

The remaining content of this chapter is organized as follows: Section 3.2 briefly reviews the related works about saliency prediction and gaze prediction. Section 3.3 describes the architecture and main components of our proposed hybrid model. We demonstrate the performance evaluation on two datasets in Section 3.4. Finally, the conclusions are given in Section 3.5.

3.2 Related Works

3.2.1 Visual Saliency Prediction.

Visual saliency is a quality of image regions that are more likely to attract human attention or gaze fixations than their neighbors [BI13]. One direction of visual saliency research used the feature integration theory [TG80] to distinguish the image region from its neighbors by some distinct visual features like brightness, contrast, and color. Itti *et al.* [IKN98] lead the research of computationally modeling visual saliency in bottom-up fashion, which is followed-up by various works such as a spectral clustering-based model [HHK12] and a graph-based saliency model [HKP07]. Recently Convolutional Neural Network (CNN) has made remarkable progress on modeling bottom-up visual saliency [LKWZ14, HSBZ15, PSGiN+16] and the CNN based saliency models greatly improved the performance. However, all of the pre-mentioned methods do not only used bottom-up visual saliency as the main cue and ignore the high-level task-related attention information such as the focus on a certain task-dependent object. Therefore, these models often fail to model natural human gaze in dynamic scenes.

Other than the aforementioned bottom-up visual saliency models, exploiting high-level information using top-down mechanisms for visual saliency prediction is receiving an increasing amount of research attention. For integrating top-down knowledge, some methods tried to guide the bottom-up features using high-level context of the goal and objects [FBR05, WK06], and others [PI07, BSI12] constructed a separate top-down visual saliency model and fused the two outputs in the final step. For example, in [TOCH06], the high-level scene context is integrated into the low-level features for saliency modeling. Very recently, researchers begin to use high-level semantic information in saliency models based on deep neural networks. Since the CNNs are trained by calculating partial derivatives of the prediction and the labels, some works [SVZ13, CLY+15] trace the partial derivatives with respect to input image regions and use the derivatives to generate a saliency map of each class. Zhao *et al.* [ZOLW15] designed a model for detecting the salient object by combining the local context of image pixels and the global infor-

mation of the whole image. In [RDZS17], the authors used image captions as high-level information to construct a region-to-word mapping for learning the visual saliency.

However, none of the previous methods explored the temporal patterns of human attention transition inherent in a complex task. In this chapter, we propose to learn the task-dependent attention transition on how gaze shifts between different objects/regions to better model human attention in natural dynamic scenes.

3.2.2 First-Person Gaze Prediction

First-person vision (a.k.a, egocentric vision) focuses on automatic analysis of first-person videos recorded with wearable cameras and is a rapidly emerging research area because of the commercial success of wearable cameras. A unique and significant component of first-person vision is egocentric gaze (or first-person gaze), since it can play an important role in multiple applications such as first-person action recognition [FLR12] and video summarization [XML+15]. Although previous works revealed the correlation between bottom-up saliency and the spatial region of gaze fixation [PLN02], it has been found that only using bottom-up visual saliency based models can result in bad performance in modeling human gaze in first-person videos [YSO+10]. Thus, previous works on first-person gaze prediction focus on integrating different cues to best model human gaze. For example, Yamada *et al.* [YSO+11] presented exploited the correlation between gaze and head motion for predicting gaze. In their model, a bottom-up saliency map is combined with an attention map obtained based on camera rotation and translation to infer the final egocentric gaze position. Li *et al.* [LFR13] explored using different cues specific for first-person videos for modeling gaze in hand manipulation tasks. These cues include the position of hand and the motion of the head/hand. They built a graphical model for leveraging the cues and modeled the temporal behavior of gaze as latent variables to improve the gaze prediction. However, their model may not generalize well to other activities where hands are not always involved, because of the dependency on hand-crafted pre-defined first-person cues. Recently, Zhang *et al.* [ZTMHL+18]

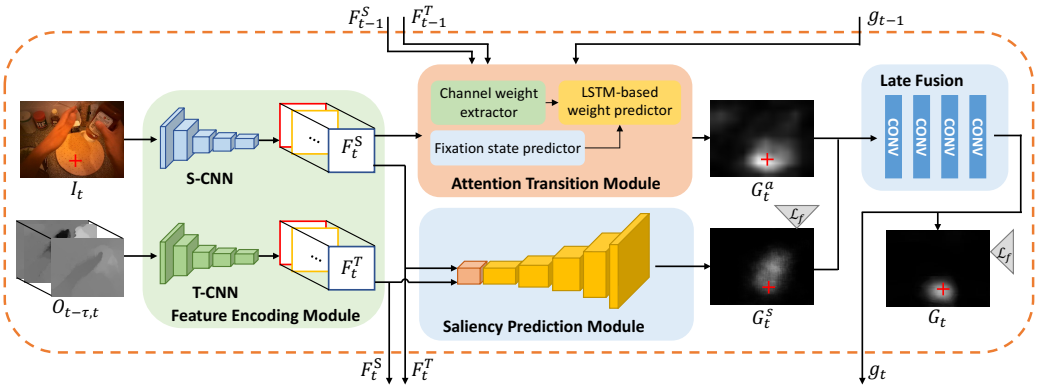

Figure 3.1: Our proposed gaze prediction model is mainly composed by these components: a feature encoding module, a saliency prediction module, an attention transition module and a late fusion module. We represent ground truth gaze positions as red cross.

proposed a Generative Adversarial Network (GAN) based model for modeling current human gaze and forecasting future gaze positions in first-person videos. They first generate future frames using the current video frame via GAN, and then predict gaze positions on the generated future frames using a saliency-based CNN with 3D convolution.

In this chapter, we propose a new hybrid model to predict gaze in first-person videos, which combines bottom-up visual saliency with task-dependent attention transition. To the best of our knowledge, this is the first work to explore the patterns in attention transition for first-person gaze prediction. By learning task-dependent attention transition, our model can exploit the temporal context of gaze fixations which greatly improves gaze prediction accuracy.

3.3 Proposed method

In this section, we first present the overview of the proposed gaze prediction model and then explain in detail about each component. We offer the details of training the model at the end of this section.

3.3.1 Model Architecture

Our model takes as input consecutive video frames and output a gaze position in each frame. To leverage both bottom-up visual saliency and task-dependent attention transition, we propose a hybrid model that 1) predicts a saliency map from each video frame, 2) predicts an attention map by exploiting temporal context of gaze fixations, and 3) fuses the saliency map and the attention map to output a final gaze map.

The model architecture is depicted in Figure 3.1. The feature encoding module is a two-stream encoder [SZ14a] assembled from a spatial Convolutional Neural Network (S-CNN) and a temporal Convolutional Neural Network (T-CNN). The S-CNN and T-CNN extract latent representations from a single RGB image and a stack of optical flow images respectively. The saliency prediction module (SP) generates a saliency map based on the extracted latent representation. The attention transition module generates an attention map based on previous gaze fixations and head motion. The late fusion module combines the results of saliency prediction and attention transition to generate a final gaze map. The details of each module will be described in the following part.

3.3.2 Feature Encoding

At time t, the current video frame I_t and stacked optical flow $O_{t-\tau,t}$ are fed into S-CNN and T-CNN to extract latent representations $F_t^S = h^S(I_t)$ from the current RGB frame, and $F_t^T = h^T(O_{t-\tau,t})$from the stacked optical flow images for later use. Here τ is fixed to be 10 following [SZ14a].

The feature encoding network of S-CNN and T-CNN follows the architecture of the first five convolutional blocks in Two Stream CNN [SZ14a], while omitting the final max pooling layer. We choose to use the output feature map of the last convolution layer from the 5-th convolutional group, i.e., *conv5_3*. Further analysis of different choices of deep feature maps from other layers is described in Section 3.4.4.

3.3.3 Saliency Prediction Module

Biologically, human tends to gaze at an image region with high saliency, i.e., a region containing unique and distinctive visual features [SMS13]. In the saliency prediction module of our gaze prediction model, we learn to generate a visual saliency map which reflects image regions that are likely to attract human gaze. We fuse the latent representations F_t^S and F_t^T as an input to a saliency prediction decoder (denoted as S) to obtain the initial gaze prediction map G_t^s (Eq. 3.1). We use the "3dconv + pooling" method of [FPZ16] to fuse the two input feature streams. Since our task is different from [FPZ16], we modify the kernel sizes of the fusion part, which can be seen in detail in Section 3.3.7. The decoder outputs a visual saliency map with each pixel value within the range of $[0, 1]$. Details of the architecture of the decoder is described in Section 3.3.7. The equation for generating the visual saliency map is:

$$G_t^s = S(F_t^S, F_t^T) \tag{3.1}$$

However, a saliency map alone does not predict accurately where people actually look [YSO+10], especially in first-person videos of natural dynamic scenes where the knowledge of a task has a strong influence on human gaze. To achieve better gaze prediction, high-level knowledge about a task, such as which object is to be looked at and manipulated next, has to be considered.

3.3.4 Attention Transition Module

During the procedure of performing a task, the task knowledge strongly influences the temporal transition of human gaze fixations on a series of objects. Therefore, given previous gaze fixations, it is possible to anticipate the image region where next attention occurs. However, direct modeling the object transition explicitly such as using object categories is problematic since a reliable and generic object detector is needed. Motivated by the fact that different channels of a feature map in top convolutional layers correspond well to spatial responses of different high-level semantics such as different object categories [CZX+17, ZKL+16], we represent the region that is likely to attract human attention by weighting each channel of the feature map differently. We train a Long Short Term Memory (LSTM) model [HS97] to predict a

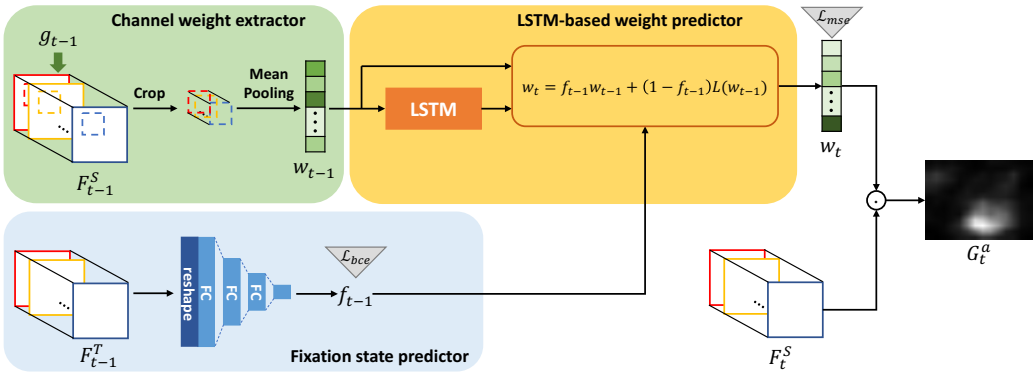

Figure 3.2: The attention transition module is composed of a channel-weight extractor, an LSTM and a fixation predictor.

vector of channel weights which is used to predict the region of attention at next fixation. Figure 3.2 depicts the framework of the proposed attention transition module. The module is composed of a channel weight extractor (C), a fixation state predictor (P), and a LSTM-based weight predictor (L).

The channel weight extractor takes as input the latent representation F^S_{t-1} and the predicted gaze point g_{t-1} from the previous frame. F^S_{t-1} is in fact a stack of feature maps with spatial resolution 14×14 and 512 channels. From each channel, we project the predicted gaze position g_{t-1} onto the 14×14 feature map, and crop a fixed size area with height H_c and width W_c centered at the projected gaze position. We then average the value of the cropped feature map at each channel, obtaining a 512-dimensional vector of channel weight w_{t-1}:

$$w_{t-1} = C(F^S_{t-1}, g_{t-1}) \tag{3.2}$$

where $C(\cdot)$ indicates the cropping and averaging operation, w_{t-1} is used as feature representation of the region of attention around the gaze point at frame $t-1$.

The fixation state predictor takes the latent representation of F^T_{t-1} as input and outputs a probabilistic score of fixation state $f^p_{t-1} = P(F^T_{t-1}) \in [0, 1]$. Basically, the score tells how likely fixation is occurring in the frame $t-1$. The fixation state predictor is composed by three fully connected layers followed by a final softmax layer to output a probabilistic score for

gaze fixation state.

We use a LSTM to learn the attention transition by learning the transition of channel weights. The LSTM is trained based on a sequence of channel weight vectors extracted from images *at the boundaries of* all gaze fixation periods with ground-truth gaze points, *i.e.* we only extract one channel weight vector for each fixation to learn its transition between fixations. During testing, given a channel weight vector w_{t-1}, the trained LSTM outputs a channel weight vector $L(w_{t-1})$ that represents the region of attention at next gaze fixation. We also consider the dynamic behavior of gaze and its influence on attention transition. Intuitively speaking, during a period of fixation, the region of attention tends to remain unchanged, and the attended region changes only when saccade happens. Therefore, we compute the region of attention at current frame w_t as a linear combination of previous region of attention w_{t-1} and the anticipated region of attention at next fixation $L(w_{t-1})$, weighted by the predicted fixation probability f^p_{t-1}:

$$w_t = f^p_{t-1} \cdot w_{t-1} + (1 - f^p_{t-1}) \cdot L(w_{t-1}) \tag{3.3}$$

Finally, an attention map G^a_t is computed as the weighted sum of the latent representation F^S_t at frame t by using the resulting channel weight vector w_t:

$$G^a_t = \sum_{c=1}^{n} w_t[c] \cdot F^S_t[c] \tag{3.4}$$

where $[c]$ denotes the c-th dimension/channel of w_t/F^S_t respectively.

3.3.5 Late Fusion

We build the late fusion module (LF) on top of the saliency prediction module and the attention transition module, which takes G^s_t and G^a_t as input and outputs the predicted gaze map G_t.

$$G_t = LF(G^s_t, G^a_t) \tag{3.5}$$

Finally, a predicted 2D gaze position g_t is given as the spatial coordinate of maximum value of G_t.

3.3.6 Training

For training gaze prediction in saliency prediction module and late fusion module, the ground truth gaze map $\hat{G}$ is given by convolving an isotropic Gaussian over the measured gaze position in the image. Previous work used either Binary Cross-Entropy loss [KWW16], or KL divergence loss [ZTMHL+18] between the predicted gaze map and the ground truth gaze map for training neural networks. However, these loss functions do not work well with noisy gaze measurement. A measured gaze position is not static but continuously quivers in a small spatial range, even during fixation, and conventional loss functions are sensitive to small fluctuations of gaze. This observation motivates us to propose a new loss function, where the loss of pixels within small distance from the measured gaze position is down-weighted. More concretely, we modify the Binary Cross-Entropy loss function ($\mathcal{L}_{bce}$) across all the N pixels with the weighting term $1 + d_i$ as:

$$\mathcal{L}_f(G, \hat{G}) = -\frac{1}{N}\sum_{i=1}^{N}(1+d_i)\big\{\hat{G}[i]\cdot log(G[i]) + (1-\hat{G}[i])\cdot log(1-G[i])\big\} \quad (3.6)$$

where d_i is the euclidean distance between ground truth gaze position and the pixel i, normalized by the image width.

For training the fixation state predictor in the attention transition module, we treat the fixation prediction of each frame as a binary classification problem. Thus, we use the Binary Cross-Entropy loss function for training the fixation state predictor. For training the LSTM-based weight predictor in the attention transition module, we use the mean squared error loss function across all the n channels:

$$\mathcal{L}_{mse}(w_t, \hat{w}_t) = \frac{1}{n}\sum_{i=1}^{n}(w_t[i] - \hat{w}_t[i])^2 \quad (3.7)$$

where $w_t[i]$ denotes the i-th element of w_t.

3.3.7 Implementation Details

We describe the network structure and training details in this section. Our implementation is based on the PyTorch [PGC+17] library. The feature encoding module follows the base architecture of the first five convolutional

blocks (*conv1* $\sim$ *conv5*) of VGG16 [SZ14b] network. We remove the last max-pooling layer in the 5-th convolutional block. We initialize these convolutional layers using pre-trained weights on ImageNet [DDS+09]. Following [SZ14a], since the input channels of T-CNN is changed to 20, we average the weights of the first convolution layer of T-CNN part. The saliency prediction module is a set of 5 convolution layer groups following the inverse order of VGG16 while changing all max pooling layers into upsampling layers. We change the last layer to output 1 channel and add sigmoid activation on top. Since the input of the saliency prediction module contains latent representations from both S-CNN and T-CNN, we use a 3d convolution layer (with a kernel size of $1 \times 3 \times 3$) and a 3d pooling layer (with a kernel size of $2 \times 1 \times 1$) to fuse the inputs. Thus, the input and output sizes are all 224 $\times$ 224. The fixation state predictor is a set of fully connected (FC) layers, whose output sizes are 4096,1024,2 sequentially. The LSTM is a 3-layer LSTM whose input and output sizes are both 512. The late fusion module consists of 4 convolution layers followed by sigmoid activation. The first three layers have a kernel size of 3 $\times$ 3, 1 zero padding, and output channels 32,32,8 respectively, and the last convolution layer has a kernel size of 1 with a single output channel. We empirically set both the height H_c and width W_c for cropping the latent representations to be 3.

The whole model is trained using the Adam optimizer [KB14] with its default settings. We fix the learning rate as 1e-7 and first train the saliency prediction module for 5 epochs for the module to converge. We then fix the saliency prediction module and train the LSTM-based weight predictor and the fixation state predictor in the attention transition module. Learning rates for other modules in our framework are all fixed as 1e-4. After training the attention transition module, we fix the saliency prediction and the attention transition module to train the late fusion module in the end.

3.4 Evaluation

We first evaluate our gaze prediction model on two public first-person activity datasets namely **GTEA Gaze** and **GTEA Gaze Plus**. We compare the proposed model with other state-of-the-art methods and provide

detailed analysis of our model through ablation study and visualization of outputs from different modules. Furthermore, to examine our model's ability in learning attention transition, we visualize output of the attention transition module on a newly collected test set from GTEA Gaze Plus dataset (denoted as **GTEA-sub**).

3.4.1 Datasets

We introduce the datasets used for gaze prediction and attention transition.

GTEA Gaze contains 17 video sequences of kitchen tasks performed by 14 subjects. Each video clip lasts for about 4 minutes with the frame rate of 15 fps and an image resolution of 480 × 640. We use videos 1, 4, 6-22 as a training set and the rest as a test set as in Yin *et al.* [LFR13].

GTEA Gaze Plus contains 37 videos with the frame rate of 24 fps and an image resolution of 960 × 1280. In this dataset each of the 5 subjects performs 7 meal preparation activities in a more natural environment. Each video clip is 10 to 15 minute long on average. Similarly to [LFR13], gaze prediction accuracy is evaluated with 5-fold cross validation across all 5 subjects.

GTEA-sub contains 227 video frames selected from the sampled frames of GTEA Gaze Plus dataset. Each selected frame is not only under a gaze fixation, but also contains the object (or region) that is to be attended at the next fixation. We manually draw bounding boxes on those regions by inspecting future frames. The dataset is used to examine whether or not our model trained on GTEA Gaze Plus (excluding GTEA-sub) has successfully learned the task-dependent attention transition.

3.4.2 Evaluation Metrics

We use two standard evaluation metrics for gaze prediction in first-person videos: Area Under the Curve (AUC) [BTSI13a] and Average Angular Error (AAE) [RDM$^+$13a]. **AUC** is the area under a curve of true positive rate versus false positive rate for different thresholds on the predicted gaze map. It is a commonly used evaluation metric in saliency prediction. **AAE** is the

average angular distance between the predicted and the ground truth gaze positions.

3.4.3 Results on Gaze Prediction

Baselines

We use the following baselines for gaze prediction:

- *Saliency prediction algorithms*: We compare our method with several representative saliency prediction methods. More specifically, we used Itti's model [IK00], Graph Based Visual Saliency (GBVS [HKP07]), and a deep neural network based saliency model as the current state of the art (SALICON [HSBZ15]).

- *Center bias*: Since first-person gaze data is observed to have a strong center bias, we use the image center as the predicted gaze position as in [LFR13].

- *Gaze prediction algorithms*: We also compare our method with two state-of-the-art gaze prediction methods: the first-person cue-based method (Yin *et al.* [LFR13]), and the GAN-based method (DFG [ZTMHL+18]). Note that although the goal of [ZTMHL+18] is gaze anticipation in future frames, it also reported gaze prediction in the current frame.

Performance Comparison

We first give the quantitative comparison of the results of different methods on two datasets in Table 3.1. Clearly, our method significantly outperforms all baselines on both datasets, particularly on the AAE score. Although there is only a small improvement on the AUC score, it can be seen that previous method of DFG [ZTMHL+18] has already achieved quite high score thus the space of improvement is limited. Besides, we have observed from experiments that high AUC score does not necessarily mean high performance of gaze prediction partly because the computation of AUC score is based on the similarity of two probability maps and our goal is to predict a 2D gaze

Table 3.1: Performance comparison of different methods for gaze prediction on two public datasets. Higher AUC (or lower AAE) means higher performance.

Metrics	GTEA Gaze Plus		GTEA Gaze	
	AAE (deg)	AUC	AAE (deg)	AUC
Itti *et al.* [IK00]	19.9	0.753	18.4	0.747
GBVS [HKP07]	14.7	0.803	15.3	0.769
SALICON [HSBZ15]	15.6	0.818	16.5	0.761
Center bias	8.6	0.819	10.2	0.789
Yin *et al.* [LFR13]	7.9	0.867	8.4	0.878
DFG [ZTMHL+18]	6.6	0.952	10.5	0.883
Our full model	**4.0**	**0.957**	**7.6**	**0.898**

position. Since our goal is to predict a 2D gaze position on each frame and AAE measures the average angular distance between the predicted and the ground truth gaze positions, we believe that the AAE score is a more appropriate metric for gaze prediction. The overall performance on GTEA Gaze is lower than that on GTEA Gaze Plus. The reason might be that the number of training samples in GTEA Gaze is smaller and over 25% of ground truth gaze measurements are missing. It is also interesting to see that the center bias outperforms all saliency-based methods and works only slightly worse than Yin *et al.* [LFR13] on GTEA Gaze Plus, which demonstrates the strong spatial bias of gaze in first-person videos.

Ablation Study

To study the effect of each module of our model, and the effectiveness of our modified binary cross entropy loss (Equation 3.6), we conduct an ablation study and test each component on both GTEA Gaze Plus and GTEA Gaze datasets. Our baselines include: 1) single-stream saliency prediction with binary cross entropy loss (**S-CNN bce** and **T-CNN bce**), 2) single-stream saliency prediction with our modified bce loss (**S-CNN** and **T-CNN**), 3)

Table 3.2: Results of ablation study

Metrics	GTEA Gaze plus		GTEA Gaze	
	AAE (deg)	AUC	AAE (deg)	AUC
S-CNN (bce)	5.61	0.893	9.90	0.854
T-CNN (bce)	6.15	0.906	10.08	0.854
S-CNN	5.57	0.905	9.72	0.857
T-CNN	6.07	0.906	9.6	0.859
SP (bce)	5.63	0.918	9.53	0.860
SP	5.52	0.928	9.43	0.861
AT	5.02	0.940	9.51	0.857
Our full model	**4.05**	**0.957**	**7.58**	**0.898**

two-stream saliency prediction with bce loss (**SP bce**), 4) two-stream input saliency prediction with our modified bce loss (**SP**), 5) the attention transition module (**AT**), and our full model.

Table 3.2 shows the results of the ablation study. The comparison of the same framework with different loss functions shows that our modified bce loss function is more suitable for the training of gaze prediction in first-person video. The SP module performs better than either of the single-stream saliency prediction (S-CNN and T-CNN), indicating that both spatial and temporal information are needed for accurate gaze prediction. It is important to see that the AT module performs competitively or better than the SP module. This validates our claim that learning task-dependent attention transition is important in first-person gaze prediction. More importantly, our full model outperforms all separate components by a large margin, which confirms that the bottom-up visual saliency and high-level task-dependent attention are complementary cues to each other and should be considered together in modeling human attention.

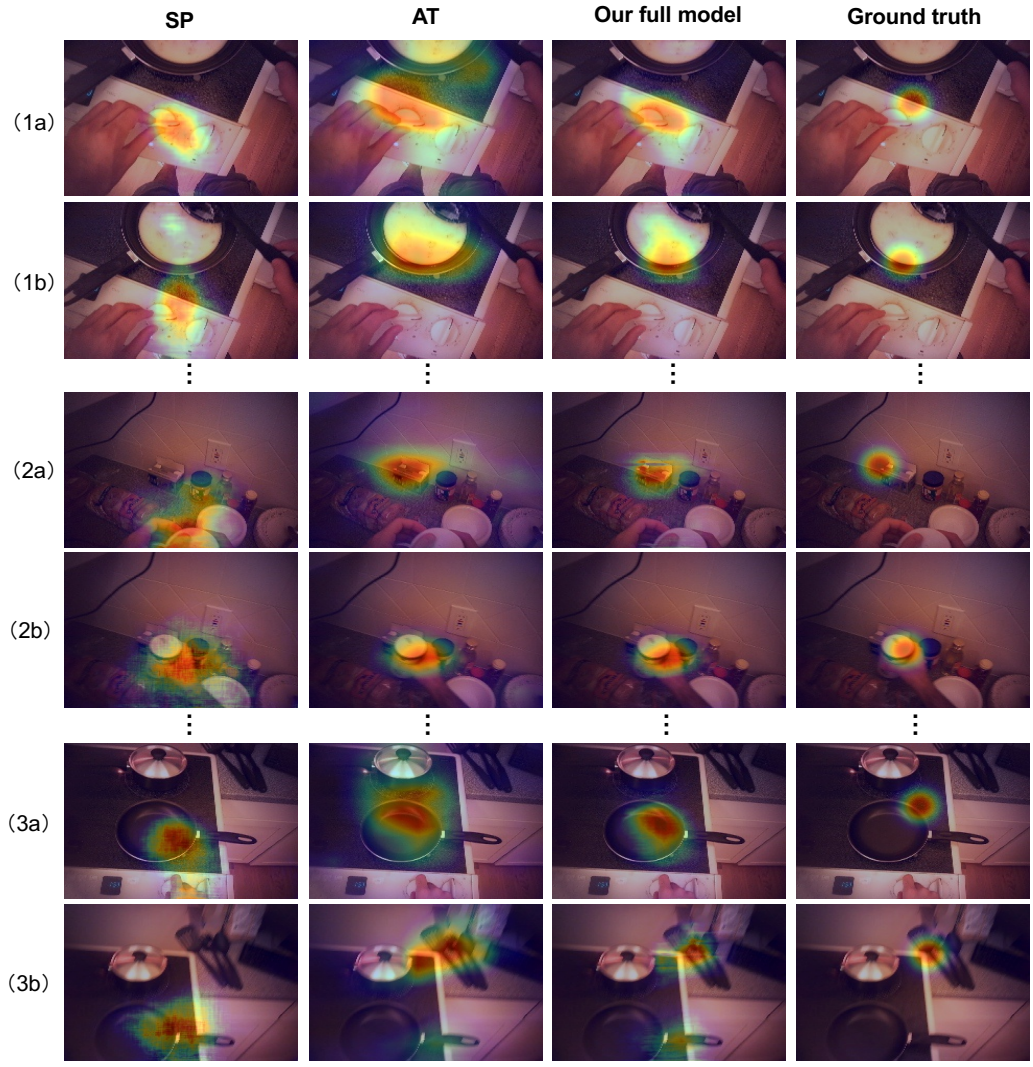

Figure 3.3: Visualization of predicted gaze maps from our model. Each group contains two images from two consecutive fixations, where a happens before b. We show the output heatmap from the saliency prediction module (**SP**) and the attention transition module (**AT**) as well as our full model. The ground truth gaze map (the rightmost column) is obtained by convolving an isotropic Gaussian on the measured gaze point.

Visualization

Figure 3.3 shows qualitative results of our model. Group (1a, 1b) shows a typical gaze shift: the camera wearer shifts his attention to the pan after turning on the oven. SP fails to find the correct gaze position in (1b) only from visual features of the current frame. Since AT exploits the high-level temporal context of gaze fixations, it successfully predicts the region to be on the pan. Group (2a, 2b) demonstrates a "put" action: the camera wearer first looks at the target location, then puts the can to that location. It is interesting that AT has learned the camera wearer's intention, and predicts the region at the target location rather than the more salient hand region in (2a). In group (3a, 3b), the camera wearer searches for a spatula after looking at the pan. Again, AT has learned this context which leads to more accurate gaze prediction than SP. Finally, group (4a, 4b) shows that SP and AT are complementary to each other. While AT performs better in (4a), and SP performs better in (4b), the full model combines the merits of both AT and SP to make better prediction. Overall, these results demonstrate that the attention transition plays an important role in improving gaze prediction accuracy.

Cross Task Validation

To examine how the task-dependent attention transition learned in our model can generalize to different tasks under same (kitchen) scene, we perform a cross validation across the 7 different meal preparation tasks on GTEA Gaze Plus dataset. We consider the following experiment settings:

- **SP**: The saliency prediction module is treated as a generic component and trained on a separate subset of the dataset. We also use it as a baseline for studying the performance variation of different settings.
- **AT_d**: The attention transition module is trained and validated under different tasks. Average performance of 7-fold cross validation is reported.
- **AT_s**: The attention transition module is trained and validated on two splits of the same task. Average performance of 7 tasks is reported.

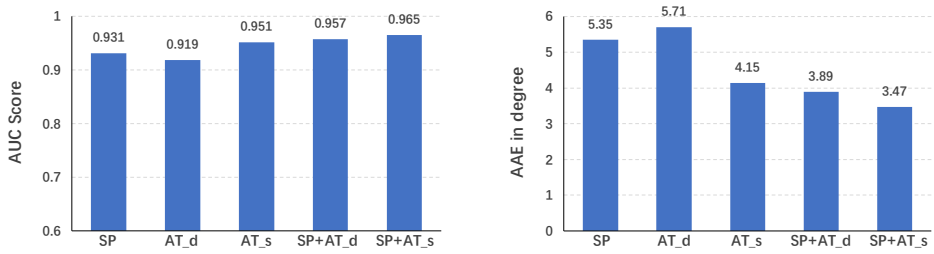

Figure 3.4: AUC and AAE scores of cross task validation. Five different experiment settings (explained in the text below) are compared to study the differences of attention transition in different tasks.

- **SP+AT_d**: The late fusion on top of **SP** and **AT_d**.
- **SP+AT_s**: The late fusion on top of **SP** and **AT_s**.

Quantitative results of different settings are shown in Figure 3.4. Both AUC and AAE scores show the same performance trend with different settings. AT_d works worse than SP, while AT_s outperforms SP. This is probably due to the differences of gaze behavior contained in different tasks. However, SP+AT_d with the late fusion module can still improve the performance compared with SP and AT_s, even with the context learned from different tasks.

3.4.4 Examination of the Attention Transition Module

We further demonstrate that our attention transition module is able to learn meaningful transition between adjacent gaze fixations. This ability has important applications in computer-aided AR system, such as implying a person where to look next in performing a complex task. We conduct a new experiment on the GTEA-sub dataset (as introduced in Section 3.4.1) to test the attention transition module of our model. Since here we focus on the module's ability of attention transition, we omit the fixation state predictor in the module and assume the output of the fixation state predictor as $f_t = 0$ in the test frame. The module takes w_t calculated from the region of current fixation as input and outputs an attention map on the same frame which represents the predicted region of the next fixation. We extract a 2D position

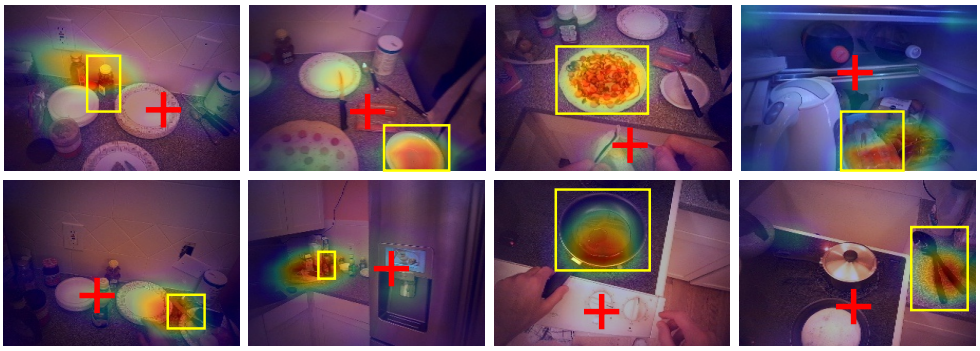

Figure 3.5: Qualitative results of attention transition. We visualize the predicted heatmap on the current frame, together with the current gaze position (red cross) and ground truth bounding box of the object/region of the next fixation (yellow box).

from the maximum value of the predicted heatmap and calculate its rate of falling within the annotated bounding box as the transition accuracy.

We conduct experiments based on different latent representations extracted from the convolutional layer: *conv5_1*, *conv5_2*, and *conv5_3* of S-CNN. The accuracy based on the above three convolutional layers are 71.7%, 83.0%, and 86.8% respectively, while the accuracy based on random position is 10.7%. We also tried using random channel weight as the output of channel weight predictor to compute attention map based on the latent representation of *conv5_3*, and the accuracy is 9.4%. This verifies that our model can learn meaningful attention transition of the performed task. Figure 3.5 shows some qualitative results of the attention transition module learned based on layer *conv5_3*. It can be seen that the attention transition module can successfully predict the image region of next fixation.

3.5 Conclusion

This chapter presents a hybrid model for gaze prediction in first-person videos. Task-dependent attention transition is learned to predict human attention from previous fixations by exploiting the temporal context of gaze fixations. The task-dependent attention transition is further integrated with

a CNN-based saliency model to leverage the cues from both bottom-up visual saliency and high-level attention transition. The proposed model achieves state-of-the-art performance in two public first-person datasets. As for the future improvements, we plan to explore the task-dependent gaze behavior in a broader scale, *i.e.* tasks in an office or in a manufacturing factory, and to study the generalizability of our model in different task domains.

This chapter and Chapter 2 depict methods for automatic modeling of human gaze and human actions. However the human as a whole coordinates his behavior globally and dynamically. The separate knowledge of each individual type of behavior does not help to get a comprehensive understanding of human behavior. The correlation between multiple types of human behavior should be considered.

Chapter 4

Joint Modeling of Human Attention and Actions

4.1 Background

Building on the prior work of human gaze prediction and human action segmentation introduced in previous chapters, this work takes a further step to study the mutual influence of gaze and actions. In particular, I consider both the task of *first-person action recognition* and *first-person gaze prediction*. Recently, many works have been done on the two tasks separately and remarkable progress has been made. However, to the best of our knowledge, although these tasks are profoundly correlated with each other, hardly any attention has been paid to leveraging their relationships to enhance the two tasks simultaneously.

In this chapter, our goal is to jointly model first-person gaze and first-person action since they are deeply correlated. Previous studies have revealed that the knowledge of the human gaze could be incorporated in the task of first-person action recognition since gaze can highlight the important action related regions. Thus, these methods design models that use machine attention mechanism to use gaze to suppress the irrelevant background information for more accurate action recognition. While it is proved that the human gaze can provide critical information on determining the human action, there is no previous effort on investigating the other side of the story, *i.e.*, "does the

human action contain information for the human gaze?"

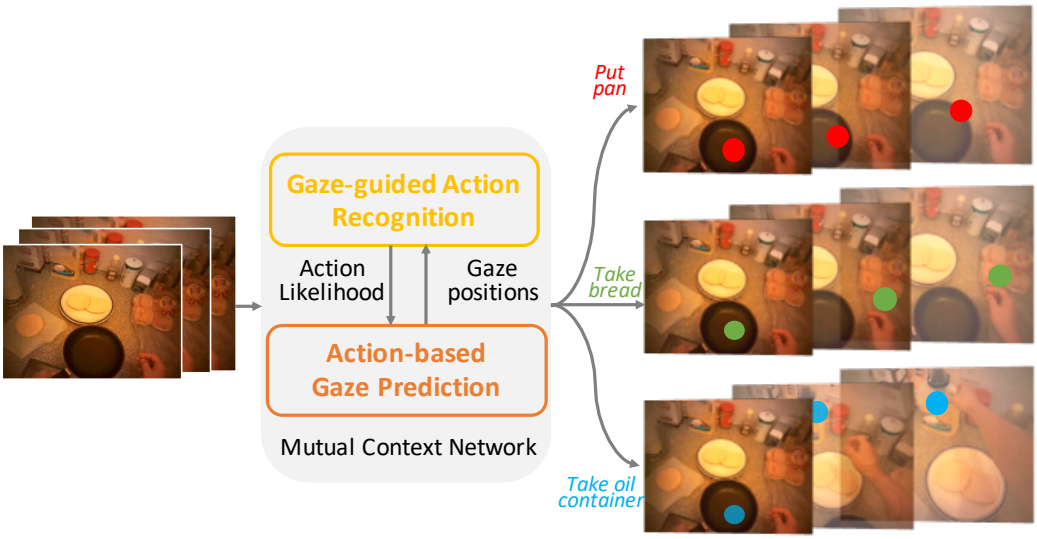

Figure 4.1: The network structure of our proposed mutual context network (MCN). The MCN takes first-person video frames as input to estimate both what action is happening and where the person is looking at. Motivated by the assumption that the human gaze and actions can provide useful information for guiding the modeling of the other, we design the MCN to take advantage of the mutual context between first-person gaze and action. For instance. we show a concept example in this figure. The desired action class will influence the position of the gaze. The positions of gaze will be more likely to be on the table if the camera wearer is going to put the pan on the table (first row). In another case, the gaze positions would be more likely on the bread if the undergoing action is "take bread" (second row).

When performing a task (especially a hand manipulation task), human gaze and actions of hand-object interaction are mutually correlated. While the image regions around a person's gaze point explicitly reveal important and discriminative information about the undergoing action, the action performed by a person implicitly affects where the person is looking [TLB92, Vic09, SKS15]. For example, to wipe the kitchen table, a person will first move his/her focus to the table cloth and then keeps gaze fixation on the

table when wiping it. Besides, similar gaze patterns happen when different persons performing the same daily action (*e.g.*"put cup"). Therefore, we argue that to take a further step in increasing the performance of modeling human gaze and first-person actions, we should jointly harness the gaze context for actions and the action context for gaze.

In this chapter, to jointly predict human gaze predictions and recognize human actions, we propose a novel framework to model the mutual context between human gaze and actions. We name our framework Mutual Context Network and will refer to it as MCN in the following of this chapter for simplicity. An illustration of the overview of our MCN is depicted in Figure 4.1. The proposed MCN uses two core modules to take video frames as input and output the likelihood of action classes of the video clip and the gaze positions on each frame. The two core modules are respectively responsible for using gaze for action recognition and integrating action information for enhancing gaze prediction. The action based gaze prediction module utilizes the estimated action likelihood as context and the gaze-guided action recognition module leverages the estimated gaze for improving action recognition. We do not discuss the motivation in detail here but offer detailed motivation of the two modules in Section 4.3.

The first module called the action-based gaze prediction module takes the predicted action likelihood as input and produces a set of convolutional kernels that are relevant to the action being performed. The generated action kernels are then used to convolve the input feature maps for locating action-related regions. It is built based on the assumption that the intent of performing an action determines which object/place to look at during the action, and therefore the information about the undergoing action should be taken into consideration for predicting human gaze. The second module called the gaze-guided action recognition module uses the estimated gaze point as a guideline to spatially aggregate the input features for action recognition. Rather than only using the region around the gaze point as in previous work, the features are aggregated both in the gaze region and the non-gaze region separately and then used as input to the gaze-guided action recognition module, while the relative importance of the two regions is learned automatically during training. Our motivation is to treat the gaze region and the non-gaze

region separately in action recognition, as the non-gaze region also provides supplementary evidence for some actions like "put" and "take". We use two widely used public datasets of first-person videos for evaluation: the EGTEA dataset [LLR18] and the GTEA Gaze plus dataset [LFR13]. We demonstrate via experiments that our method can achieve state-of-the-art performance on both first-person gaze prediction and first-person action recognition.

Here is the summarization of the major contribution of this chapter: Firstly, in this chapter, we propose a new framework based on deep neural networks for leveraging the mutual context between human gaze and actions. Secondly, a novel module for gaze prediction using action context is developed in this chapter that explicitly utilizes the estimated action likelihood. To the best of our knowledge, this is the first work that considers action context for first-person gaze prediction. Thirdly, our proposed framework achieves state-of-the-art performance on two public datasets GTEA Gaze plus and EGTEA on both first-person gaze prediction and first-person action recognition tasks.

4.2 Related Works

4.2.1 First-Person Gaze Prediction

Gaze prediction from first-person video is a well-established research topic [LFR13] and can benefit a diverse range of applications such as joint attention discovery [HCK+17, KYHS16, PJS12], human computer interaction [FTT17, KAH+16, KFJ+16], action recognition [FLR12] and video summarization [XML+15]. Also, the analysis of first-person gaze can provide significant cues in the research of cognitive science [LAL+17] and developmental psychology [VSC+18, EGCSB17]. Previous works tried to leverage different kinds of cues for gaze prediction, and perhaps the most famous cue for gaze prediction is visual saliency [PLN02]. Since only by using the visual saliency cannot predict accurate gaze position in natural dynamic scenes, the following works tried to use various additional cues for gaze prediction in first-person videos [LFR13, TRKB19, YSO+10, ZTMHL+18, ZBYC18]. For example, Li *et al.*[LFR13] hand-crafted multiple first-person features such

as head motion and hand position and use a graphical model for predicting gaze prediction on first-person videos of cooking dishes in a kitchen. However, the hand-crafted first-person cues may harm the performance of their model applied in other scenarios. With the development of the recent deep learning techniques, Zhang *et al.*[ZTMHL+18] first designed deep convolutional neural networks for predicting and forecasting gaze. Their method is based on 3D convolution and the model is trained in a bottom-up manner for directly construct a mapping between RGB values of image pixels and pixelwise likelihood of gaze position. However, these works only rely on the image appearance for gaze prediction, which turns to be challenging without high-level information especially when multiple salient objects appear simultaneously in the field of view. Our previous chapter is the first work to assimilate high-level information with bottom-up appearance based information. The high-level information we used is the temporal task-dependent gaze shift patterns. However, in the previous chapter we considered the general action transition which is action agnostic. Here in this chapter we consider the more detailed gaze patterns with respect to the ongoing actions.

In this chapter, we propose the first work for the gaze prediction task that explicitly associates the contextual influence from the undertaking actions. We use the action likelihood as the contextual information for gaze prediction.

4.2.2 First-Person Action Recognition

Within the field of first-person vision, action recognition is almost the most focused research direction with a reasonable number of studies in the past few years [FBF18, LLWP19, PEPA16, PR12, SPL+07, YYJ+17, YKS16, CTRD18]. Separating by the target, the action recognition of first-person videos can be divided into two categories: the first category aims to recognize the motion of the camera wearer, *e.g.*, "take" or "wash" [KOSS11, SAJ16], while the second category target recognizing actions in a finer level, *i.e.*, recognizing both the motion and the object related with the motion such as "take pencil" or "wash dishes". Earlier works mostly fall into the first group. For example, Kitani *et al.*[KOSS11] recognized the first-person motion by grouping global camera motion in an unsupervised manner. In the

work of Singh *et al.*[SAJ16], a deep neural network with image, optical flow and hand segmentation mask as input is designed and proved to be effective in action recognition. Poleg *et al.*[PEPA16] aim to recognize the long-term actions and designed a 3D CNN with optical flow images as input. The work on the second category can also trace back to a decade before, where Fathi *et al.*[FFR11] adopted a graphical model to jointly model the objects, hand and head motion for inferring the first-person actions. More recently together with the rise of deep learning, Ryoo *et al.*[RRM15] proposed a novel pooling method for improving the performance of action recognition. In [MFK16] the authors described a two-stream model for jointly recognizing action and the action-object. Sudhakaran *et al.*[SL18] proposed a novel attention block equipped on LSTM networks for spatially localizing the discriminative regions for action recognition. Recently, researchers [WFF+19] further leveraged supportive information extracted over a longer video span named long-term feature bank to augment the receptive field of the models. In this chapter, we focus on the second category of fine-grained action recognition. Different from previous works, our method recognizes actions with the contextual information from gaze by modeling actions and gaze in a unified framework.

4.2.3 Gaze and Actions

Only a few works connected the tasks of first-person gaze prediction and action recognition, and most of which only considered using gaze for action recognition [FLR12, SNLZ18, ZYP+18] but not counter-wisely, despite that human gaze and actions are deeply correlated in first-person videos. For example, Li *et al.*[LYR15] used hand-crafted features extracted from only the gaze region and found these features are more discriminative for the task of action recognition. Shen *et al.*[SNLZ18] encoded gaze positions using bounding boxes obtained by an extra object detection model. They define an "event" to be the case that gazes temporally go inside and then outside of a bounding box, and designed an event-based recurrent neural network for action recognition and achieved promising performance. However, very few works have jointly modeled the human gaze and human actions in a unified

framework. Li *et al*.extended from their previous work of [FLR12] to a deep model [LLR18] for jointly modeling gaze and actions. They integrated the probabilistic nature of gaze into the convolutional layers for the prediction of gaze and used the probabilistic distribution of gaze for improving action recognition result. However, their work only implicitly modeled action and gaze in a neural network like a black box. We in this chapter explicitly leverage the mutual contextual information of gaze and actions and found in the experiment that gaze prediction could be greatly improved and influenced by the contextual information of the likelihood of actions.

In this chapter, we demonstrate our model that leverages the mutual contextual information between gaze and actions for jointly modeling gaze and actions and improving the gaze prediction and action recognition performance. We use the action likelihood as a conditional input for guiding the gaze prediction, and in the meantime use the gaze as a threshold factor for supporting the action recognition. We show using experiments that our proposed framework for explicitly exploring such mutual context can obtain state-of-the-art performance in both first-person action recognition and gaze prediction.

4.3 Motivation

4.3.1 Action Context for Gaze Prediction

The key to accurate prediction of gaze in a first-person video is to locate the regions of human attention. While this step is usually done by estimating the visual saliency based on the image appearance, this often fail especially in first-person videos of daily activities where multiple salient objects or regions may exist simultaneously. We believe that top-down information should be amalgamated in the gaze prediction model for determining the real gaze position when the salient regions are ambiguous. By observing first-person videos of natural daily living, we discovered that a semantic connection exists between gaze region and the action of the camera wearer. It is possible for this semantic information to be used in the gaze prediction framework so as to improve its performance. We find that the fine-grained

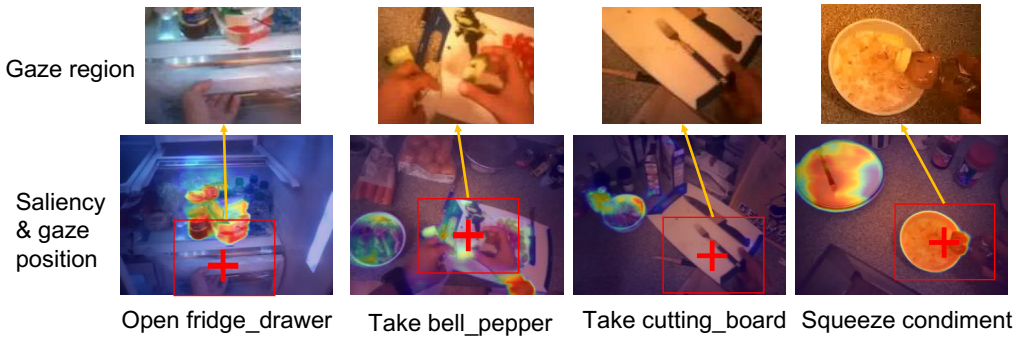

Figure 4.2: The difference between saliency maps (overlayed on images) and gaze regions (red cross, also enlarged above). The saliency maps are obtained using PiCANet [LHY18] pretrained on the DUTS dataset [WLW+17]. We can see that the gaze region is more action-dependent and can be significantly different from the visually salient regions.

first-person actions comprised of a verb and one or several nouns, encodes semantic information that is critical for locating the region of attention. For example, the nouns encode the region which is highly likely to be attended. An example is shown in Figure 4.2, where the gaze regions at the top row match well with the semantic information involved in the performed actions, but do not necessarily match the visually salient regions. For example, in the action of *Open fridge_drawer*, the object of *fridge_drawer*, as well as the hand, are contained in the real gaze region, while visual saliency is mainly distributed on other salient regions irrelevant to the performed action.

Motivated by this semantic connection between the gaze region and the performed action, we propose a framework that could incorporate contextual information from action for gaze prediction. In the proposed framework, information about the performed action (*e.g.*, represented by the softmax vector of action recognition) is used to produce intermediate information which is semantically meaningful and could be directly utilized for gaze prediction.

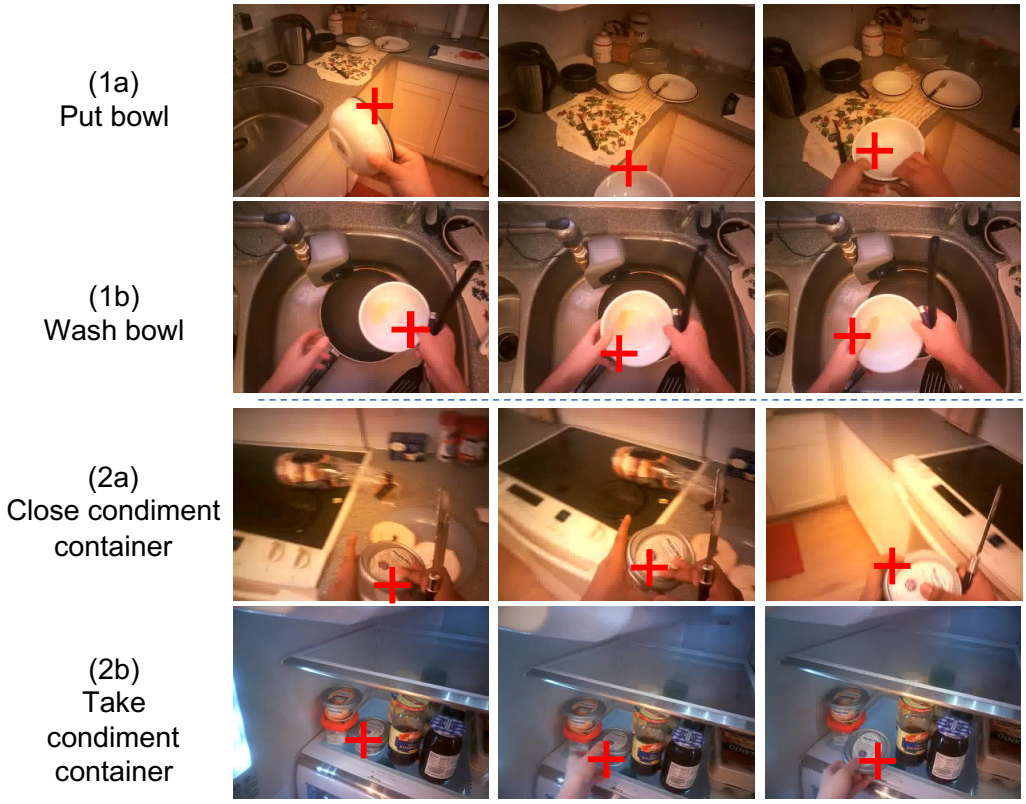

Figure 4.3: Gaze context for different actions. In (1a) and (1b), gaze focuses on the regions of bowl which help to recognize *Put bowl* and *Wash bowl* from other actions. With additional features from surrounding background, it is able to further differentiate the two actions. Similarly, in (2a) and (2b), it is easier to recognize *Close condiment_container* and *Take condiment_container* by extracting features from both gaze regions and background.

4.3.2 Gaze Context for Action Recognition

Human attention mechanism functions via gaze focus on the action-related object when executing an action [HB05]. Thus, one important characteristic of human gaze is that it highlights critical regions or indicates important information about the objects being manipulated. Based on such motivation, previous works [FFR11, LYR15, LLR18] have shown that giving more weight to the visual features from gaze regions, can lead to better performance of

action recognition. While information from the gaze regions is useful for recognizing many actions, the information from the surrounding background is also needed for differentiating some fine-grained actions with similar objects. As shown in (1a) and (1b) of Figure 4.3, it is hard to distinguish the actions only with information from gaze regions. The information from the surrounding background, in other words, the "non-gaze" regions, helps to distinguish these two actions, since the existence of the sink in (1b) strongly indicates the action to be *wash bowl* rather than *put bowl*. Similarly in (2a) and (2b), the fridge and the containers around the gaze region help for the recognition of *take condiment_container* rather than *close condiment_container*.

Thus, motivated by the usefulness of the regions guided by gaze in action recognition, we propose to make use of information from both the gaze regions and the non-gaze regions in a complementary way for better action recognition.

Overall, in this chapter we propose an MCN to model the mutual context of action and gaze. The proposed MCN jointly solves the two coupled tasks of first-person gaze prediction and first-person action recognition by using the context from one task to help the other. We will then describe the details of the proposed framework.

4.4 Approach

In this section, we first introduce an overview of the proposed MCN followed by details of each of its modules. We then provide the training strategy of MCN and finally the details of the model architecture at the end of this section.

4.4.1 Overview

In this work, we propose a mutual context network (MCN) that leverages the mutual context of action gaze for joint gaze prediction and action recognition. The MCN uses the estimated action to predict the gaze point while in the mean time uses gaze as a guidance for action recognition.

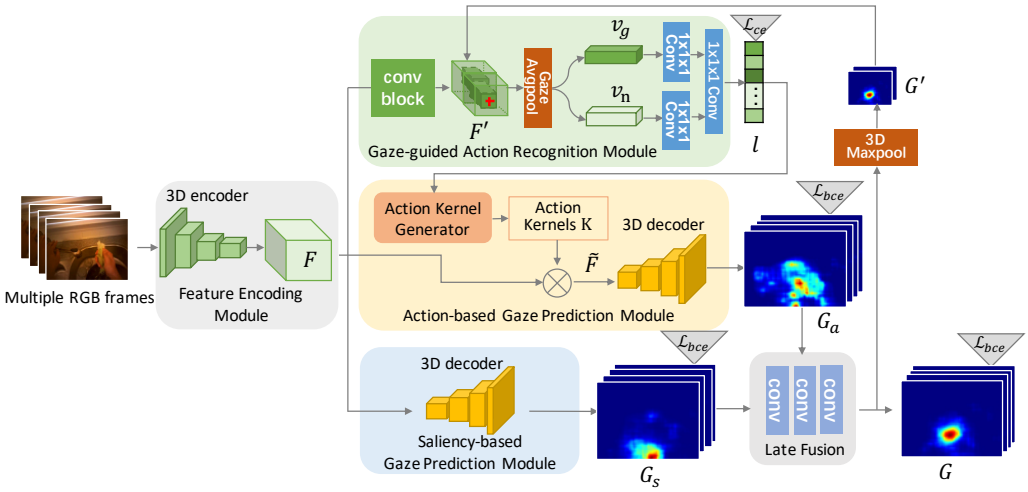

Figure 4.4: Architecture of our proposed mutual context network (MCN). MCN consists of 5 sub-modules: the feature encoding module which encodes input video frames into feature maps F, the gaze-guided action recognition module which uses gaze as a guideline to recognize actions, the action-based gaze prediction module which takes predicted action likelihood l as input and outputs an action-dependent gaze probability map G_a, the saliency-based gaze prediction module which outputs a saliency map G_s, and finally the late fusion module to get the final gaze probability map G.

Figure 4.4 depicts the architecture of our MCN. A feature encoding module consist of 3D convolution blocks first encode the RGB frames and optical flow images of a trimmed video snippet into a feature map F. F is then used as input to the following modules. One of the key components in our model is the action-based gaze prediction module that learns to predict gaze G_a using the predicted action likelihood l as a conditional input. As complementary information for gaze prediction, we also obtain a saliency map G_s with the saliency-based gaze prediction module. The outputs from the two modules are then fused by the late fusion module to get the final gaze probability map $G = \{g_1, g_2, \cdots, g_N\}$. Another component in our MCN is the gaze-guided action recognition module which takes the predicted gaze G as guidance to selectively filter the input features for action recognition. The output of ac-

tion likelihood l is then used as conditional input to the action-based gaze prediction module, thus a loop of mutual context is constructed.

4.4.2 Feature Encoding Module

We adopt the first four convolutional blocks of the resnet50 version I3D network I3D-resnet [WGGH18] for feature encoding. Following [LLR18], we fuse the RGB stream and optical flow stream at the end of the 4th convolutional block by element-wise summation. With this 3D encoder, the output feature map F is of size (c, t, h, w), where c is the number of channels, t is the temporal dimension, and (h, w) are the spatial height and width.

4.4.3 Saliency-Based Gaze Prediction Module

Image regions with high saliency tend to attract human attention. For instance, regions with unique and distinguishing features such as a moving object or high contrast of brightness are more likely to be looked at than other regions. Therefore, we use a saliency-based gaze prediction module to learn the image regions that are more likely to draw human attention. For this, we use a 3D decoder that takes the encoded feature map F as input and outputs a series of gaze probability maps G_s with each pixel value within the range of [0, 1]. While this bottom-up approach provides information about salient regions in the image, it is not sufficient to reliably identify the attended region when multiple salient regions exist, which is common in first-person video.

4.4.4 Action-Based Gaze Prediction Module

As different actions are associated with different objects and motion, people's gaze patterns when performing different actions are different. As stated in Section 4.3.1, motivated by the connection between the region of attention and the performed action, we propose an action-based gaze prediction module to leverage action information for more reliable gaze prediction. The proposed module is expected to be able to extract semantically meaningful

information that could be used for locating gaze regions. To this end, inspired by [XWBF18, CYL+17], we use the estimated action likelihood from the action recognition module to generate a group of convolutional kernels (called as "action kernels") which encode the semantic information of the performed action. The generated action kernels are then used to convolve with the input features in order to locate the action-related regions. Finally, gaze probability maps that have the same size with input frames are generated by a decoder consisting of deconvolutional layers.

More formally, given action likelihood $l \in \mathbb{R}^n$ estimated by the action recognition module and the input feature maps $F \in \mathbb{R}^{c \times t \times h \times w}$ with c channels (t and h, w are temporal and spatial dimension), the gaze probability map G_a is generated through the following procedure:

$$K = A(l) \tag{4.1}$$

$$\tilde{F} = K \otimes F \tag{4.2}$$

$$G_a = Decoder(\tilde{F}) \tag{4.3}$$

where A is the action kernel generator, $K \in \mathbb{R}^{k \times c \times k_t \times k_h \times k_w}$ is a group of k kernels, and $\tilde{F} \in \mathbb{R}^{k \times t \times h \times w}$ is the filtered feature maps. $\otimes$ denotes the operator of convolution. The kernel generator contains one fully connected layer and two convolutional layers. The output of the first fully connected layer is first reshaped into size (k, k_t, k_h, k_w) and then forwarded to the following convolution layers.

We also adopt the saliency-based gaze prediction module which can be seen as a complementary to the action-based gaze prediction module. Finally, we use a late fusion module to combine the outputs G_s and G_a from the previous modules:

$$G = LF(G_s, G_a) \tag{4.4}$$

Late fusion technique has been proved to be effective in previous work of gaze prediction [HCLS18]. Following previous works [ZTMHL+18, LFR13], we take the spatial location with maximum likelihood on G as the predicted gaze point.

4.4.5 Gaze-Guided Action Recognition Module

Here we describe the gaze-guided action recognition module in our MCN that uses the predicted gaze point as a guide to exploit discriminative features for action recognition. Previous works [FLR12, LLR18] mostly used gaze as a filter to remove features of image regions far from the gaze point. However, focusing only on the region around the gaze point might lose important information about the action. We observed that when performing certain actions such as "put an object", the person may fixate on the table on which to place the object instead of looking at the object in hand which contains critical information about the action. Therefore, we think that while the gaze region is important, the region outside the gaze (non-gaze region) might also contain complementary information about the action. In this work, we develop a two-way pooling structure to aggregate features in the gaze and non-gaze regions separately and use both as input for action recognition.

As shown in Figure 4.4, we first forward F to the fifth convolutional block of I3D to encode more compact features $F' \in \mathbb{R}^{c' \times t' \times h' \times w'}$. On each temporal dimension of F', we locate the corresponding spatial gaze point $(x_{\bar{t}}, y_{\bar{t}})$ on the feature map by selecting the spatial location of the maximum value in the 3d max-pooled gaze map G'. Then we split spatial dimensions of the feature map into two parts: gaze region and non-gaze region. Gaze region on a feature map (dark green region of F' in the figure) is the locations whose spatial positions are within range $([x_{\bar{t}}-r, x_{\bar{t}}+r], [y_{\bar{t}}-r, y_{\bar{t}}+r])$, and non-gaze region is the left-out region (light green region of F' in the figure). We pool the two regions separately on the spatial dimensions, generating two feature tensors v_g and v_n:

$$v_g[c,t] = \frac{\sum_{i=x_{\bar{t}}-r}^{x_{\bar{t}}+r} \sum_{j=y_{\bar{t}}-r}^{y_{\bar{t}}+r} \bar{F}'[c,t,i,j]}{4r^2} \tag{4.5}$$

$$v_n[c,t] = \frac{\sum_i \sum_j \bar{F}'[c,t,i,j] - 4r^2 v_g[c,t]}{h' \times w' - 4r^2}, \tag{4.6}$$

where $\bar{F}'_x[c,t,i,j]$ denotes the c-th channel and position (t,i,j) of the feature map F'_x, similarly for $v[c,t]$.

The pooled feature tensors v_g and v_n are fed into two 1x1x1 convolution layers (denoted as $\mathcal{F}_g, \mathcal{F}_n$), and the outputs are channel-wise concatenated

and forwarded into the final 1x1x1 convolution layer (denoted as $\mathcal{F}_{logit}$) for predictions. We average the predictions on temporal dimension to get the action likelihood l:

$$v'_g \in \mathbb{R}^{s \times t} = \mathcal{F}_g(v_g) \tag{4.7}$$

$$v'_n \in \mathbb{R}^{s/2 \times t} = \mathcal{F}_n(v_n) \tag{4.8}$$

$$l = Softmax(Average(\mathcal{F}_{logit}(\{v'_g; v'_n\}))) \tag{4.9}$$

Here $\{;\}$ denotes channel-wise concatenation. We set the output channel of v'_g to be s and v'_n to be $\frac{s}{2}$ since the modeling of non-gaze region is empirically simpler than that of the gaze region, so we limit its channel size to prevent over-fitting.

4.4.6 Implementation and Training Details

The whole framework is implemented using Pytorch framework [PGC+17]. The feature encoding module is identical to the first 4 convolutional blocks of the I3D-resnet [WGGH18] network without the last pooling layer. With our input of 24 stacked images of size 320×240, the output of feature encoding module is of size $c = 1024, t = 6, h = 14, w = 14$. The 3D decoder contains a set of 4 transposed convolution layers, with kernel sizes $4, 4, (3, 4, 4), (3, 4, 4)$, and stride $2, 2, (1, 2, 2), (1, 2, 2)$ respectively. Padding 1 is added on all layers. Each layer is followed by batch normalization and ReLU activation. We add another convolution layer with kernel size 1 and a sigmoid layer on top of the 3D decoder for outputting values within $[0, 1]$. The action kernel generator takes the input vector $l \in \mathbb{R}^n$ where n is the number of action categories, and firstly encoded to a latent size of $\mathbb{R}^{4800}$ and reshaped into $(64, 3, 5, 5)$. The two convolutional layers output channels 256 and 1024, with kernel size 3, stride 1 and padding 1. The output size of the action kernel generator is $(k, c, k_t, k_w, k_h) = (64, 1024, 3, 5, 5)$. For the gaze guided action recognition module, the convolution block is identical to the 5-th convolution block of the I3D-resnet network. Thus the output size of F'_x is $(c', t', h', w') = (1024, 3, 7, 7)$. The 3d max-pooling layer therefore has kernel size (8,32,32). We set $r = 1$ and $s = 256$. The late fusion module is composed of 4 convolutional layers with output channels 32,32,8,1, in which the

first 3 layers have a kernel size of 3 with 1 zero padding and the last layer has a kernel size of 1 with no padding.

For training the whole network, we first train the gaze-guided action recognition module and the saliency-based gaze prediction module using ground truth action labels and gaze positions. We use Adam optimizer [KB14] in all experiments. The base I3D weights are initialized from weights pretrained on kinetics dataset [KCS+17]. We then use the result of action recognition to train the action-based gaze prediction module and then the late fusion module. We use cross entropy loss for action recognition and binary cross entropy loss for gaze prediction. We apply a Gaussian with $\sigma = 18$ on the gaze point for generating ground truth images for gaze prediction. The learning rates for action recognition module and all gaze prediction modules are fixed as 10^{-4} and 10^{-7} respectively. We first resize the images to 256×256 and then random crop images into 224×224, random flip with probability 0.5 for data augmentation during training. Ground truth gaze images perform the same data augmentation. When testing, we resize the image and send both the images and their flipped version and report the averaged performance.

Algorithm 1 Alternative inference procedure

1: Using the saliency-based gaze prediction module to initialize gaze prediction G:
   $G \leftarrow G_s$;
2: Denote action likelihood vectors as l.
3: **while** $e > 0.1$ and $\#iteration \leq max\_iter$ **do**
4: Update l from gaze-guided action recognition module based on G;
5: Get G_a from action-based gaze prediction module using l;
6: Update G using the previous G and G_a:
   $G_{new} \leftarrow LF(G, G_a)$;
7: Compute the AAE of G and G_{new}:
   $e \leftarrow AAE(G, G_{new})$;
8: $G \leftarrow G_{new}$
9: **end while**

We iteratively infer gaze positions and action likelihood vectors in an alternative fashion as described in Algorithm 1. The iteration terminates

when the variation (measured by average angular error AAE) of current gaze prediction from the previous prediction is below a threshold or the number of iteration surpasses an upper bound. We empirically set this upper bound max_iter to be 10.

4.5 Evaluation

4.5.1 Dataset and Evaluation Metric

Our experiments are conducted on two public datasets: EGTEA [LLR18] and GTEA Gaze+ [LFR13]. The GTEA Gaze+ dataset consists of 7 activities performed by 5 subjects. Each video clip is 10 to 15 minutes with resolution 1280×960. We do a 5-fold cross validation across all 5 subjects and take their average for evaluation as [LFR13]. The EGTEA dataset is an extension of GTEA Gaze+ which contains 29 hours of first-person videos with the resolution of 1280×960 and 24 fps, taken from 86 unique sessions with 32 subjects performing meal preparation tasks in a kitchen environment. Fine-grained annotations of 106 action classes are provided together with measured ground truth gaze points on all frames. Following [LLR18], we use the first split (8299 training and 2022 testing instances) of the dataset to evaluate the performance of gaze prediction and action recognition. We use the trimmed action clips of both datasets for training and testing unless otherwise noted.

We compare different methods on both tasks of gaze prediction and action recognition. For gaze prediction, we adopt two commonly used evaluation metrics: AAE (Average Angular Error in degrees) [RDM+13b] and AUC (Area Under Curve) [BTSI13b]. For action recognition, we use classification accuracy as the evaluation metric.

4.5.2 Results of Gaze Prediction

We compare our method with the following baselines:

- Saliency prediction methods: we use two representative traditional methods **GBVS** [HKP07], **Itti's model** [IK00] as our baseline. We also re-implement the deep FCN based model **SALICON** [HSBZ15]

as another baseline and train on the same dataset with gaze as ground truth saliency map.

- First-person gaze prediction methods: We also compare with three first-person gaze prediction methods closely to our work: coarse gaze prediction method (**Li *et al.***[LLR18]), the GAN-based method (**DFG** [ZTMHL+18]), and the attention transition-based method (**Huang *et al.*** [HCLS18]). Since [LLR18] only outputs a coarse gaze prediction map (of resolution 7×7), we resize their output using bilinear interpolation. For **Li *et al.*** and **DFG** we report the results based on our implementation as no code is publicly available. For **Huang *et al.*** we use the author's original implementation.

- Subsets of our full MCN: We also conduct ablation study using subsets of our full model. These include the saliency-based gaze prediction module (**Saliency-based**), the action-based gaze prediction module (**Action-based**). In addition, we also test the action-based gaze prediction module with ground truth action labels (**Action-based***). To further validate that the action-based gaze prediction module can provide useful information, we change the action-based gaze prediction module to center-bias and feed them into the late fusion module, which forms the ablation baseline **Saliency-based + center bias**.

Table 4.1 shows the quantitative comparison of different methods on gaze prediction performance. We first analyze the performance comparison with previous methods shown on the top part of the table. Our method outperforms state-of-the-art first-person gaze prediction methods ([ZTMHL+18] and [HCLS18]) on both datasets with the same experimental setting. It is important to notice that even our action-based gaze prediction module alone could achieve comparable performance with [HCLS18], which verifies the effectiveness of action context on gaze prediction.

We also conduct ablation study by comparing different subsets of our MCN. As shown in the lower part of Table 4.1, the action-based module performs better than the saliency-based module, verifying the effectiveness of action context in gaze prediction. When feeding the action-based module with ground-truth action labels, the performance is further improved.

Method	EGTEA		GTEA Gaze+	
	AAE	AUC	AAE	AUC
GBVS [HKP07]	12.81	0.707	12.68	0.829
Itti *et al.*[IK00]	12.50	0.717	12.73	0.801
SALICON [HSBZ15]	11.17	0.881	12.34	0.867
Li *et al.*[LLR18]	8.58	0.870	8.97	0.889
DFG [ZTMHL+18]	6.30	0.923	6.39	0.910
Huang *et al.*[HCLS18]	6.25	0.925	6.23	0.924
Saliency-based	6.36	0.922	6.57	0.929
Saliency-based + center bias	6.30	0.924	6.51	0.930
Action-based	6.20	0.928	6.35	0.923
Action-based*	6.04	0.927	6.20	0.933
Our full MCN	**5.79**	**0.932**	**5.74**	**0.945**

Table 4.1: Comparison of gaze prediction performance on two datasets. Results of previous methods are placed on top. Results of our full MCN and the subsets of MCN are placed on the bottom. Lower AAE and higher AUC indicate better performance. * denotes using ground truth action label as input.

To examine the effectiveness of late fusion, we first tried the the fusion of saliency-based module with center bias and found that it only slightly improves the performance of saliency-based module alone. However, the fusion of saliency-based module with action-based module (our full MCN) greatly improves the performance of two individual modules, as demonstrated by the decrease of AAE score from 6.36/6.20 to 5.79 on EGTEA dataset and from 6.57/6.35 to 5.74 on GTEA Gaze+ dataset. This indicates that an ideal gaze prediction method should consider information from both low-level visual saliency and high-level action context.

Qualitative results are shown in Figure 4.5. It can be seen that with the help of the action-based gaze prediction module, our full MCN can better locate the action, thus giving better gaze prediction results. For example,

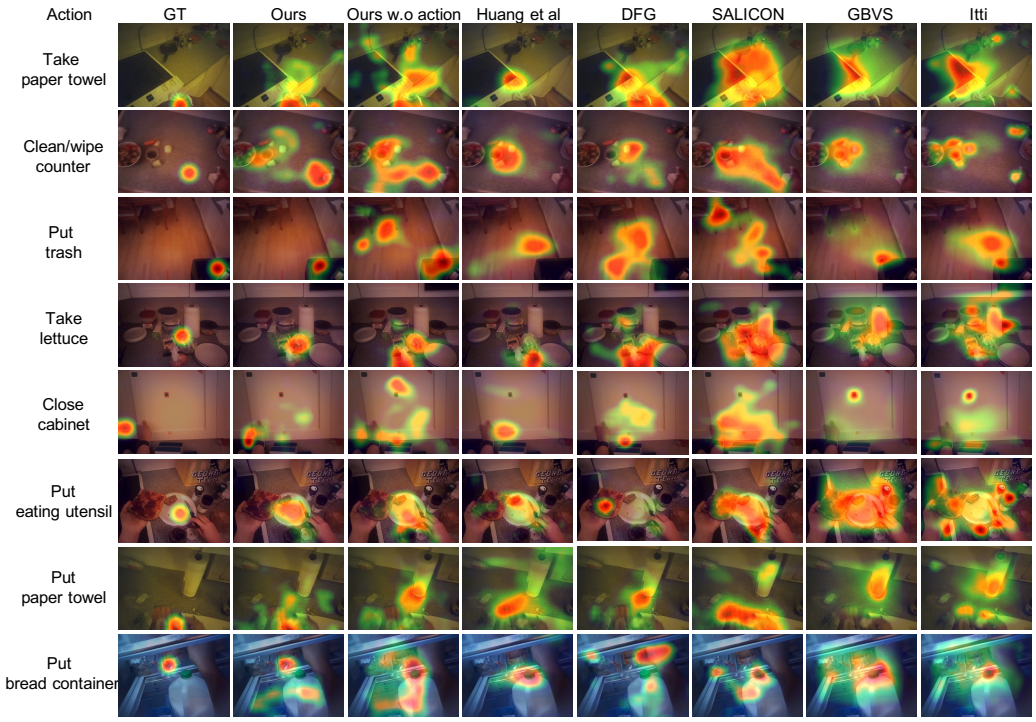

Figure 4.5: Qualitative visualizations of gaze prediction results on EGTEA dataset. We show the output heatmap from our full MCN and several baselines. Ground truth action labels and gaze points (**GT**) are placed on the leftmost columns.

in the first row, our MCN successfully recognizes the action as "take paper towel", thus finds the paper towel in the hand. Other baseline methods mostly focus on the stove or other salient regions. In the second row, while other methods are distracted by the plates and food on the counter, our MCN successfully locates the hand with dishrag on the bottom right corner and a part of the counter which will be cleaned in the next few frames. More interestingly as shown in the fourth row, the lettuce of ground-truth gaze fixation is placed on a cluttered kitchen table, which is challenging for other methods to locate. Still, our full MCN correctly predicts gaze to be on the lettuce with the help of context from the action "take lettuce". Similar situations can be found in other rows of the figure.

4.5.3 Examination of Action-based Gaze Prediction Module

We further demonstrate that our action-based gaze prediction module is able to learn meaningful gaze patterns relevant to different actions. Intuitively, the gaze patterns for similar actions should also be similar: for example, for the action "take bowl", the gaze prediction performance should not decrease obviously if we use a label of "take plate" as input to the action-based gaze prediction module, but should decline sharply if it is given the label of "cut tomato" as input. Thus we conduct a new experiment on the top 20 frequent actions in test set of EGTEA dataset to examine our action-based gaze prediction module. We feed the module with action label representing each of the 20 action classes and examine how gaze prediction performance (AAE score) varies when the module is tested on each of these actions. For example, we feed the action-based gaze prediction module with the action label of "take plate" and test the AAE scores on the videos of all 20 actions. As a result, we obtain a matrix of AAE scores with the size of 20×20, denoted by M, in which $M_{i,j}$ is the AAE score of the action-based gaze module fed with the action label of the i-th action and applied to the videos of the j-th action.

We found that the average AAE on the diagonal of M is 6.21, while the average AAE of M without diagonal is 6.87. This indicates that correct action label can benefit the predictions of action-based gaze prediction module. To better understand the effect of different action labels on the action-based gaze prediction module, we visualize an "affinity matrix" A with the following equation:

$$A_{i,j} = 1 - \frac{M_{i,j} - min(M_{i,*})}{max(M_{i,*}) - min(M_{i,*})}, \tag{4.10}$$

where $A_{i,j}$ can be seen as the "affinity score" between the measured ground truth gaze pattern of the i-th action and the learned gaze pattern of the j-th action. We normalize each number to have numeric range of [0,1].

We visualize the affinity matrix in Figure 4.6. We can see from several dark blocks along the diagonal (marked by boxes) that there exist several groups of actions of which the learned gaze patterns are similar to each other, for example, the action group of "put" in the middle and the action group

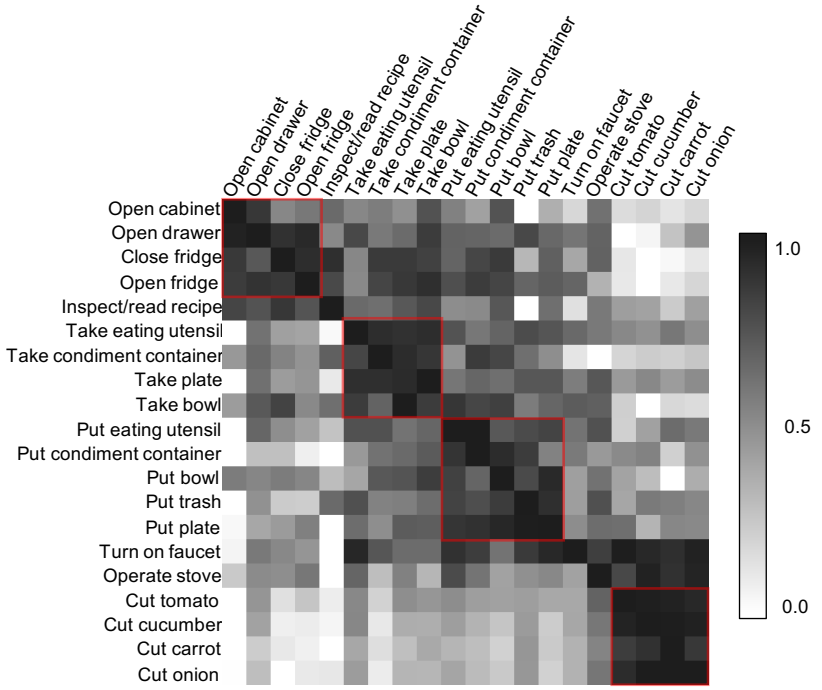

Figure 4.6: Affinity matrix of the top 20 frequent actions in EGTEA dataset. Actions are re-ordered for the ease of viewing. Each row of the matrix represents the "affinity score" of one action against all the 20 actions. Darker indicates higher "affinity" between corresponding actions. We mark several darker groups of similar action with high "affinity" for the ease of reading.

of "cut" on the bottom-right of the figure. The obtained affinity matrix is actually consistent with our common sense of these actions. For the action group of "put", persons tend to fixate on a table which is often the destination of placement. For the action group of "cut", the gaze is often fixated on a knife. More importantly, the results show that our action-based gaze prediction module has learned meaningful action-based gaze patterns. We think these patterns might be used to study the similarity between different actions from the perspective of human attention in future works.

4.5.4 Results of Action Recognition

As for the task of action recognition, we compare our method with the following methods:

- **I3D** [CZ17] is one of the state of the art models for action recognition. We refer to [LLR18] for the accuracy of this baseline method.

- Methods using measured gaze: **I3D+Gaze** is to use a ground truth gaze point as a guideline to pool feature maps from the last convolution layer of the fifth convolutional block. **EgoIDT+Gaze** [LYR15] is a traditional method which uses dense trajectories [WKSL11] selected by a ground truth gaze point for action recognition.

- State-of-the-art first-person action recognition methods: **Li *et al.*** [LLR18] uses a estimated gaze probability map as soft attention to perform a weighted average on top I3D features. **Sudhakaran *et al.*** [SL18] adopts attention mechanism in a recurrent neural network to recognize actions. **LSTA** [SEL19] is a recent RNN based first-person action recognition method that models action related attention for better action recognition. We also compare our method with **Ma *et al.*** [MFK16] and **Shen *et al.*** [SNLZ18] that use additional annotations of object locations and hand masks during training. [SNLZ18] even uses ground-truth gaze positions as input during testing. We compare the performance as reported in their original papers.

- Baselines of our model: **MCN (w/o gaze)** is the baseline that does not use gaze information and is constructed to validate the effectiveness of the gaze-guided action recognition module. It performs a direct average pooling as in [CZ17, WGGH18]. The **MCN (center bias)** is the baseline that uses the image center as the predicted gaze position. We construct this baseline to validate the usefulness of better gaze prediction on action recognition. **MCN (gaze region)** is a baseline of our MCN that uses only the gaze-centered region for pooling. We use this baseline to validate the usefulness of information from the non-gaze regions. **MCN (soft gaze)** is a baseline that uses the predicted gaze

probability map as a soft attention map on the features of the final convolutional block as in [LLR18].

Method	EGTEA	GTEA Gaze+
EgoIDT + Gaze [LYR15]	46.50	60.50
I3D [CZ17]	49.79	57.64
I3D [CZ17] + Gaze	51.21	59.72
Li *et al.*[LLR18]	53.30	N/A
Sudhakaran *et al.*[SL18]	N/A	60.13
Ma *et al.*[MFK16]	N/A	66.40
Shen *et al.*[SNLZ18]	N/A	(67.10)
LSTA [SEL19]	61.86	N/A
MCN (w/o gaze)	55.59	59.88
MCN (center bias)	53.22	59.59
MCN (gaze region)	56.43	61.12
MCN (soft gaze)	60.83	65.52
Our full MCN	**62.58**	**67.36**

Table 4.2: Quantitative comparison of action recognition. We report recognition accuracy in %. Values in brackets indicate the methods that rely on ground truth gaze.

Quantitative comparison of different methods on two datasets is shown in Table 4.2. The deep learning method I3D [CZ17] outperforms EgoIDT+Gaze [LYR15] that uses handcrafted features on EGTEA dataset but not on GTEA Gaze+ dataset. This is possibly due to the smaller number of training samples in GTEA Gaze+ dataset. With the use of measured gaze, the performance of I3D+Gaze is improved compared with I3D. On both datasets, our MCN outperforms state-of-the-art methods ([MFK16, SNLZ18, SEL19]), including [SNLZ18] that relies on ground-truth gaze positions during testing.

We also conduct ablation study to examine the effectiveness of different components of our model. The baseline of (MCN w/o gaze) takes whole images as input without considering distinct information from gaze or non-

gaze regions. It performs better than the similar method of I3D [CZ17] and shows the advantage of more advanced base network (I3D-resnet [WGGH18]) adopted in our model. The comparison between MCN (center bias) and (gaze region) indicates the usefulness of predicted gaze for action recognition. The superiority of our full model over MCN (gaze region) indicates the usefulness of the non-gaze regions, and validates our thought that the non-gaze regions should be considered together with gaze regions in action recognition. Although MCN (soft gaze) partly considers regions distant from gaze with less weight, our full model outperforms MCN (soft gaze) by explicitly incorporating information from gaze and non-gaze regions.

4.6 Discussion

4.6.1 Model Convergence

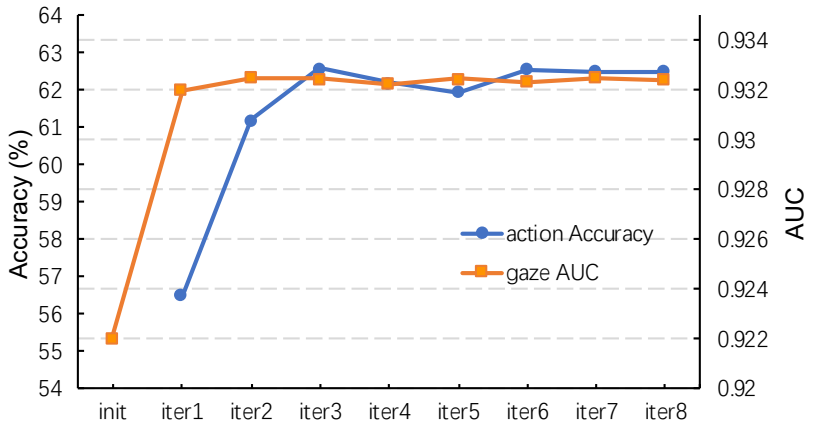

Figure 4.7: Gaze prediction AUC and action recognition accuracy with respect to inference iteration on the EGTEA dataset. Blue curve with circle markers correspond to action recognition accuracy on the left axis, and orange curve with square markers correspond to gaze prediction AUC on the right axis.

In the proposed method, the network inference is conducted in an alternative manner. Here, we show the performance of two tasks on EGTEA dataset

with our method at different iteration of inference in Figure 4.7. Note that at the stage of initialization, only saliency-based module is used. Then gaze-guided action recognition and action-based gaze prediction are conducted alternatively from the first iteration. It can be seen that the performance of both gaze prediction and action recognition increases dramatically at first and converges after about two iterations. This strongly supports our hypothesis that the mutual context of gaze and action can be beneficial for both tasks. In addition, the performance of gaze prediction converges faster than that of action recognition. We think the reason might be that even coarse information of actions (*e.g.*, the object or verb of an action) is sufficient as context for gaze prediction. Actually this is also demonstrated as in Figure 4.6 that several groups of actions have learned similar gaze patterns.

4.6.2 Failure Cases

Figure 4.8: Failure cases of our MCN on gaze prediction. In the first row, failed action recognition misleads gaze prediction. In the second row, although the action recognition is correct, the camera wearer shifts the gaze fixation onto the region of future destination when he/she has already finished the action of grabbing the bread.

Here we discuss several failure cases of gaze prediction with our method. One failure case happens when critical information of an action is incorrectly

predicted. As shown in the first row of Figure 4.8, the wrong prediction of "take cup" as "take plate" causes our model to focus on the region of the plate while true gaze is on the region of the cup. Still, the impact of failed action recognition is limited in our model. We have analyzed gaze prediction results in two opposite cases and found that: among all the testing data of the EGTEA dataset, our model achieves an AAE score of 6.01 when action recognition fails, and an AAE score of 5.68 when action recognition is correct.

Another failure case comes from the circumstances when a person begins to shift the gaze fixation between consecutive actions. An example is shown in the second row of Figure 4.8. After grabbing the bread, instead of keeping fixation on the bread, the person's attention goes to the plate on which he's planning to put the bread. Our current method is trained based on trimmed action sequences and could not identify such circumstances, thus fails to predict the true gaze positions at the boundaries of consecutive actions. This reveals the necessity of taking attention transition into consideration for our current gaze prediction model.

To further analyze the effect of learning attention transition with untrimmed video, we also compared the performance of the original version of [HCLS18] (denoted as Huang *et al.*†) which is trained based on untrimmed videos on the GTEA Gaze+ dataset. By learning attention transition, Huang *et al.*† achieves performance of 4.83 in the AAE metric and 0.939 in the AUC metric, and outperforms our method by the metric of AAE (4.83 versus 5.74). Meanwhile, when trained with the same trimmed videos, Huang *et al.* [HCLS18] clearly performs worse than our method (as shown in Table 4.1), possibly due to the lack of consideration for action context. Overall, the comparison between our method and the two variants of [HCLS18] shows that while our method can benefit from action context and achieves state-of-the-art performance on the trimmed dataset, its current version could not fully explore the useful information from additional data in untrimmed videos. We think the combination of action-based gaze prediction and attention transition between consecutive actions could be a good research direction to explore and would leave it as our future work.

4.7 Conclusion

This chapter aims to explore the mutual influence between human gaze and human actions. In this chapter, we proposed a novel deep model for both first-person gaze prediction and action recognition. Our model explicitly leverages the mutual context between the two tasks. Within our model, the action-based gaze prediction module predicts gaze positions using the a set of convolutional kernels generated based on the action likelihood. The gaze-guided action recognition module selectively aggregates the features of gaze region and non-gaze region for better action recognition. Experiments show that our model achieves state-of-the-art performance for both tasks on two public first-person video datasets.

Although our model outperforms previous methods in trimmed action sequences, gaze prediction performance still needs further improvement, especially for the transition periods between consecutive actions in untrimmed videos. As for the future work, We think it would be an interesting direction to explore the gaze transition patterns in a broader activity scope which involves multiple consecutive fine-grained actions. Another possible direction of future work is to study the ways of mitigating the negative influence of failed action recognition on gaze prediction. We think considering the likelihood of the top-5 action and their relations might be a promising direction to explore. The biggest obstacle for deploying the system into real-world applications is the high computational cost of the proposed framework. In the future it would be promising to improve the efficiency of the proposed model to enable wider applications such as skill assessment. Also, since eye-trackers on wearable cameras are often costly and hard to use, replacing the eye-trackers with build-in algorithms is another interesting direction to improve this work.

Chapter 5

Conclusion and Future Directions

In this thesis, I present methods for automatically modeling human behavior from two aspects: modeling the human gaze and modeling human actions, under the first-person paradigm. Chapter 1 explains the motivation of this work, describing the importance of the first-person human behavior modeling and the different types of human behaviors to model. For modeling each type of human behavior, we propose methods that leverage various novel cues that are introduced in the following chapters. Chapter 2 introduces a new model for the task of action segmentation in first-person videos. It is targeted to solve the problem of observation limitation in first-person videos. A graph-based temporal reasoning module is proposed to model the temporal relation among multiple action segments. The proposed module can be applied on top of most existing action segmentation models and is proved by experiments to be effective on both first-person and third-person videos. In Chapter 3, a first-person vision model is proposed to predict where people look (gaze prediction) in everyday manipulation tasks. The model takes as input the videos recorded by a wearable camera and outputs the likelihood map indicating the possibility of the point (pixel) to be looked at by the camera wearer. Advances of deep learning techniques are incorporated in the model to combine the saliency cue and task-related attention transition cue together. The model achieves state-of-the-art performance and is ahead

of other methods by a large margin. In Chapter 4, a further step is taken for human behavior modeling by jointly considering the human behaviors of gaze and action. Based on the hypothesis that the knowledge of action can improve the performance of human gaze prediction and vice versa, a mutual context network is designed to simultaneously predict gaze and recognize the observed action. Experiments on first-person video datasets strongly prove the hypothesis. As a whole, the methods presented in this thesis offer a computational way of studying the behavior of humans in natural daily manipulation tasks.

The main contributions of this work are summarized as follow:

- This thesis proposes a lightweight module for action segmentation in long and complex first-person videos. Temporal relations among multiple neighboring actions are leveraged to overcome the challenge of limited observations in the first-person perspective. Furthermore, the proposed module is shown to be able to cooperate with multiple backbone models for action segmentation and can perform well on both first-person and third-person videos.

- This thesis proposes a first-person vision model for human gaze prediction. The model is capable of predicting where humans look during everyday manipulation tasks with a single wearable monocular camera with state-of-the-art performance. The work shows the potential for using computer vision techniques to replace the use of eye-trackers for enabling more real-life settings.

- This thesis proposes a method for jointly estimating the human gaze and recognizing the observed action given first-person videos. The task of gaze prediction and action recognition mutually improves each other, and consequently lead to better performance of both tasks. The experiments demonstrate that different human behaviors should be jointly considered for more thorough and accurate modeling of human behavior.

Future Work

The majority of this thesis has focused on advancing the frontier of human behavior modeling using deep neural networks. I would like to conclude this thesis with the future direction. My future work involves developing generalized and robust systems for human behavior modeling, and the applications of human behavior modeling in various scenarios. My research will progress along the following paths:

Generalized Gaze Prediction

In Chapter 2, the proposed model leverages high-level attention transition cues for gaze prediction. However, the model is only trained and validated in indoor kitchen scenes. Since the model relies on a saccade prediction module, its performance could be affected by the scene changes. I will work on gaze prediction in more generalized settings, including both indoor and outdoor videos. I will leverage the commonalities of human dynamics to link human gaze behavior in different environments. I also consider to use other sensors such as stereo camera, RGB-D camera, accelerometer, or EMG sensor for improving the gaze prediction performance and enabling the gaze prediction to be in 3D space.

Detecting Overlapping Actions

In Chapter 3, a graph-based temporal reasoning module is proposed for segmenting first-person perspective actions from long and complex videos. One drawback of the proposed module is that it cannot detect the action when it overlaps. Overlapping actions can frequently happen especially in the first-person perspective, where people sometimes do two actions simultaneously (*e.g.*drinking water while typing). I recently started investigating the use of action detection in solving the problem of overlapping actions. I will explore the solution to model the large diversity of first-person videos captured in different domains for better detecting the subtle or short actions. I will also investigate the network architecture design so that the model can effectively diversify the action snippets and the non-action snippets.

Other Future Directions

As discussed in the previous chapters, the modeling of human behavior can facilitate a wide range of applications such as autism diagnosis and human-computer interaction. However, there still exist obstacles for directly applying the methods in the current form. Here I list several future directions to be explored for the deployment of my previous works into real-world applications.

Chapter 2 and Chapter 4 both introduce methods for first-person gaze prediction. However, the efficiency of the proposed modules is still sub-optimal. The encoder-decoder style of neural networks used in these two chapters is computationally costly and not real-time, which could harm the real-world deployment especially when we want to replace the burdensome eye-tracker with built-in software. Making the models more lightweight or using parallel computing techniques might be a good starting point. Other directions include but not limit to applying neural architecture search, using compression techniques to directly process the videos, and take advantage of quantized neural networks.

Since the video data of human behavior often involves privacy concerns, an important question for us is how to learn the modeling of human behavior without violating privacy. For example, when we want to develop a system based on gaze prediction for assisting the autism diagnosis [BHL+10], it may be difficult to acquire enough data for centralized training. This could be solved by using a line of rapidly emerging research called federated learning [LSTS20]. I am fascinated by investigating the use of federated learning in first-person human behavior modeling.

The modeling of human attention can benefit an interesting application towards the next-generation intelligent robot. By utilizing imitation learning [LMM07], robots can learn to behave like humans in the real world without direct instruction from human by just analyzing first-person videos. I will investigate developing new methods to transfer the knowledge of human behavior to robots, especially focusing on relieving the burden of human labeling during the robot training. Through these efforts, I hope to contribute to the development of intelligent robots that could quickly understand the

environment and complete tasks by imitating the learned human behavior.

Bibliography

[BHL+10] Elgiz Bal, Emily Harden, Damon Lamb, Amy Vaughan Van Hecke, John W Denver, and Stephen W Porges. Emotion recognition in children with autism spectrum disorders: Relations to eye gaze and autonomic state. *Journal of autism and developmental disorders*, 2010.

[BI13] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[BKSS14] Subhabrata Bhattacharya, Mahdi M Kalayeh, Rahul Sukthankar, and Mubarak Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014.

[BNW+18] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[BSI12] Ali Borji, Dicky N Sihite, and Laurent Itti. Probabilistic learning of task-specific visual attention. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012.

[BTSI13a] Ali Borji, Hamed R Tavakoli, Dicky N Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013.

[BTSI13b] Ali Borji, Hamed R Tavakoli, Dicky N Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009.

[CFPC14] Yu Cheng, Quanfu Fan, Sharath Pankanti, and Alok Choudhary. Temporal sequence modeling for video event detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014.

[CHEGCN15] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.

[CLY+15] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2015.

[CRY+19] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[CTRD18] Alejandro Cartas, Estefania Talavera, Petia Radeva, and Mariella Dimiccoli. On the role of event boundaries in egocentric activity recognition from photostreams. *arXiv preprint arXiv:1809.00402*, 2018.

[CYL+17] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[CZ17] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[CZS17] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.

[CZX+17] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. 2017.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2016.

[DDMF+18] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2009.

[DPSCRDS17] Carolina Diaz-Piedra, Jose M Sanchez-Carrion, Héctor Rieiro, and Leandro L Di Stasi. Gaze-based technology as a tool for surgical skills assessment and training in urology. *Urology*, 2017.

[DX17] Li Ding and Chenliang Xu. Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. *arXiv preprint arXiv:1705.07818*, 2017.

[DX18] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[EGCSB17] Maria K Eckstein, Belén Guerra-Carrillo, Alison T Miller Singley, and Silvia A Bunge. Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental cognitive neuroscience*, 2017.

[etm] *Argus Science ETMobile Eye Tracking Glasses*. https://imotions.com/hardware/argus-science-eye-tracking-glasses/.

[FBF18] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2018.

[FBR05] Simone Frintrop, Gerriet Backer, and Erich Rome. Goal-directed search with a top-down modulated computational attention system. In *Joint Pattern Recog. Symp.*, 2005.

[FFR11] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2011.

[FG19] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[FLR12] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *Eur. Conf. Comput. Vis. (ECCV)*, 2012.

[FPZ16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[FR13] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013.

[FRR11] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2011.

[FTT17] Minatsu Fujisaki, Hiroshi Takenouchi, and Masataka Tokumaru. Interactive evolutionary computation using multiple users' gaze information. In *Int. Conf. on HCI*, 2017.

[goo] *Google Glasses*. https://www.google.com/glass/start/.

[gop] *Gopro Hero 7 Black*. https://gopro.com/en/us/shop/cameras/hero7-black/CHDHX-701-master.html.

[GYDD18] Pallabi Ghosh, Yi Yao, Larry S Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018.

[GYN17] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *Brit. Mach. Vis. Conf. (BMVC)*, 2017.

[HB05] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 2005.

[HCK+17] Yifei Huang, Minjie Cai, Hiroshi Kera, Ryo Yonetani, Keita Higuchi, and Yoichi Sato. Temporal localization and spatial segmentation of joint attention in multiple first-person videos. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2017.

[HCLS18] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[HFFN16] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[HGS19] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019.

[HHK12] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

[HKP07] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2007.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[HSBZ15] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2015.

[HUB+19] R Austin Hicklin, Bradford T Ulery, Thomas A Busey, Maria Antonia Roberts, and JoAnn Buscaglia. Gaze behavior and cognitive states during fingerprint target group localization. *Cognitive research: principles and implications*, 2019.

[HZRH18] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2018.

[IK00] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 2000.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998.

[IZJ+17] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017.

[KAH+16] Mohamed Khamis, Florian Alt, Mariam Hassib, Emanuel von Zezschwitz, Regina Hasholzner, and Andreas Bulling. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In *2016 CHI Conf. Extended Abstracts on Human Factors in Comput. Sys.*, 2016.

[KAS14] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014.

[KASB17] Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. Gaze embeddings for zero-shot image classification. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KCS+17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[KFJ+16] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. Eyeswipe: Dwell-free text entry using gaze paths. In *2016 CHI Conf. on Human Factors in Computing Sys.*, 2016.

[KGS16] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *Win. Conf. Applic. Conput. Vis. (WACV)*, 2016.

[KKUG07] Volker Krüger, Danica Kragic, Aleš Ude, and Christopher Geib. The meaning of action: A review on action recognition and mapping. *Advanced robotics*, 2007.

[KOSS11] Kris M Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2011.

[KRG17] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017.

[KSDB14] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, 2014.

[KW17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Int. Conf. on Learning Repres. (ICLR)*, 2017.

[KWW16] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[KYHS16] Hiroshi Kera, Ryo Yonetani, Keita Higuchi, and Yoichi Sato. Discovering objects of joint attention via first-person sensing. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2016.

[LAL+17] Hannah Lawrence, Lucy Akehurst, Amy-May Leach, Julie Cherryman, Aldert Vrij, Megan Arathoon, and Zarah Vernham. 'look this way': using gaze maintenance to facilitate the detection of children's false reports. *Applied Cognitive Psychology*, 2017.

[Lan04] Michael F Land. The coordination of rotations of the eyes, head and trunk in saccadic turns produced in natural situations. *Experimental brain research*, 2004.

[LFR13] Yin Li, Alireza Fathi, and James M Rehg. Learning to predict gaze in egocentric video. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013.

[LFV+17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[LG18] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2018.

[LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[LHY18] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[LHZ+18] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2018.

[LKWZ14] Yuetan Lin, Shu Kong, Donghui Wang, and Yueting Zhuang. Saliency detection within a deep convolutional architecture. In *AAAI Workshops*, 2014.

[LLL+19] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action pro-

posal generation. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[LLR18] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[LLWP19] Minlong Lu, Ze-Nian Li, Yueming Wang, and Gang Pan. Deep attention network for egocentric action recognition. *IEEE Trans. Image Processing*, 2019.

[LMM07] Manuel Lopes, Francisco S Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *IEEE/RSJ Conf. Intel. Rob. Sys.*, 2007.

[LRVH16] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[LSTS20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Sig. Proc. Mag.*, 2020.

[LT18] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[LYR15] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.

[MFK16] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[MJY+19] Khoi-Nguyen C. Mac, Dhiraj Joshi, Raymond A. Yeh, Jinjun Xiong, Rogerio S. Feris, and Minh N. Do. Learning motion in feature space: Locally-consistent deformable convolution

networks for fine-grained action detection. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[MMH+08] Emily Moxley, Tao Mei, Xian-Sheng Hua, Wei-Ying Ma, and BS Manjunath. Automatic video annotation through search and mining. In *IEEE Int. Conf. Mult. Expo (ICME)*, 2008.

[NCA+17] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *IEEE Conf. on Robot. and Auto. (ICRA)*, 2017.

[PCF19] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[PEPA16] Yair Poleg, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. Compact cnn for indexing egocentric videos. In *Win. Conf. Applic. Conput. Vis. (WACV)*, 2016.

[PGC+17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[PGZ19] Jia-Hui Pan, Jibin Gao, and Wei-Shi Zheng. Action assessment by joint relation graphs. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[PI07] Robert J Peters and Laurent Itti. Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2007.

[PJS12] Hyun S Park, Eakta Jain, and Yaser Sheikh. 3d social saliency from head-mounted cameras. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2012.

[PLN02] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 2002.

[PR12] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012.

[PR14] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014.

[PSGiN+16] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[QWJ+18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[RAAS12] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012.

[RDM+13a] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: state-of-the-art and study of comparison metrics. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013.

[RDM+13b] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations:

state-of-the-art and study of comparison metrics. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013.

[RDZS17] Vasili Ramanishka, Abir Das, Jianming Zhang, and Kate Saenko. Top-down visual saliency guided by captions. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2015.

[RKG17] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[RRM15] Michael S Ryoo, Brandon Rothrock, and Larry Matthies. Pooled motion features for first-person videos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.

[SAJ16] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[SEL19] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[SKS15] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.

[SL18] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. *Brit. Mach. Vis. Conf.*, 2018.

[SM13] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013.

[SMJ+16] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[SMS13] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Appearance-based gaze estimation using visual saliency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[SNLZ18] Yang Shen, Bingbing Ni, Zefan Li, and Ning Zhuang. Egocentric activity prediction via event modulated attention. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[SPL+07] Dipak Surie, Thomas Pederson, Fabien Lagriffoul, Lars-Erik Janlert, and Daniel Sjölie. Activity recognition using an egocentric perspective of everyday objects. In *Int. Conf. on Ubiquitous Intel. and Computing*, 2007.

[SR17] Matthias Schröder and Helge Ritter. Deep learning for action recognition in augmented reality assistance systems. In *ACM SIGGRAPH 2017 Posters*. 2017.

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[SY18] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[SZ14a] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2014.

[SZ14b] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TCSU08] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, 2008.

[TFFK12] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012.

[TG80] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 1980.

[TG20] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[TLB92] Steven P Tipper, Cathy Lortie, and Gordon C Baylis. Selective reaching: evidence for action-centered attention. *Journal of Experimental Psychology: Human Perception and Performance*, 1992.

[tob] *Tobii Pro Glasses 2*. https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/.

[TOCH06] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 2006.

[TRKB19] Hamed Rezazadegan Tavakoli, Esa Rahtu, Juho Kannala, and Ali Borji. Digging deeper into egocentric gaze prediction. In *Win. Conf. Applic. Conput. Vis. (WACV)*, 2019.

[VB14] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2014.

[Vic09] Joan N Vickers. Advances in coupling perception and action: the quiet eye as a bidirectional link between gaze, attention, and action. *Progress in brain research*, 2009.

[VSC+18] Angelina Vernetti, Atsushi Senju, Tony Charman, Mark H Johnson, Teodora Gliga, BASIS Team, et al. Simulating interaction: Using gaze-contingent eye-tracking to measure the reward value of social signals in toddlers with and without autism. *Developmental cognitive neuroscience*, 2018.

[VW87] Robin R Vallacher and Daniel M Wegner. What do people think they're doing? action identification and human behavior. *Psychological review*, 1987.

[WFF+19] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[WG18] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[WGGH18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[WK06] Dirk Walther and Christof Koch. Modeling attention to salient proto-objects. *Neural networks*, 2006.

[WKSL11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2011.

[WLS+19] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J. Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[WLW+17] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[WXW+16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[WZKX18] Shaojie Wang, Wentian Zhao, Ziyi Kou, and Chenliang Xu. How to make a blt sandwich? learning to reason towards understanding web instructional videos. *arXiv preprint arXiv:1812.00344*, 2018.

[XGC+19] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S. Davis, and David J. Crandall. Temporal recurrent networks for online action detection. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[XML+15] Jia Xu, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M Rehg, and Vikas Singh. Gaze-enabled egocentric video summarization via constrained submodular maximization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.

[XWBF18] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Stochastic future gen-

eration via layered cross convolutional networks. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[YKS16] Ryo Yonetani, Kris M Kitani, and Yoichi Sato. Recognizing micro-actions and reactions from paired egocentric videos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[YMR16] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[YPLM18] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[YSO+10] Kentaro Yamada, Yusuke Sugano, Takahiro Okabe, Yoichi Sato, Akihiro Sugimoto, and Kazuo Hiraki. Can saliency map models predict human egocentric visual attention? In *Asian Conf. Comput. Vis. (ACCV)*, 2010.

[YSO+11] Kentaro Yamada, Yusuke Sugano, Takahiro Okabe, Yoichi Sato, Akihiro Sugimoto, and Kazuo Hiraki. Attention prediction in egocentric video using motion and visual saliency. In *Pacific-Rim Symposium on Image and Video Technology*, 2011.

[YXL18] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[YYJ+17] Tang Yansong, Tian Yi, Lu Jiwen, Feng Jianjiang, and Zhou Jie. Action recognition in rgb-d egocentric videos. In *IEEE Int. Conf. on Image Processing*, 2017.

[ZAOT18] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[ZBYC18] Zehua Zhang, Sven Bambach, Chen Yu, and David J Crandall. From coarse attention to fine-grained gaze: A two-stage 3d fully convolutional network for predicting eye gaze in first person video. 2018.

[ZHT+19] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2019.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[ZOLW15] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.

[ZSXS19] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *arXiv preprint arXiv:1908.09995*, 2019.

[ZTHS19] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[ZTMHL+18] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Qi Zhao, and Jiashi Feng. Anticipating where people will look using adversarial networks. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[ZXW+17] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[ZYP+18] Zheming Zuo, Longzhi Yang, Yonghong Peng, Fei Chao, and Yanpeng Qu. Gaze-informed egocentric action recognition for memory aid systems. *IEEE Access*, 2018.

Publications

Publications Related to the Thesis

[1] **Yifei Huang**, Minjie Cai, Zhenqiang Li and Yoichi Sato. "Predicting Gaze in Egocentric Video by Learning Task-dependent Attention Transition". In proc. *European Conference on Computer Vision (ECCV)*, October 2018, oral presentation.

[2] **Yifei Huang**, Yusuke Sugano and Yoichi Sato. "Improving Action Segmentation via Graph Based Temporal Reasoning". In proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] **Yifei Huang**, Minjie Cai, Zhenqiang Li, Feng Lu and Yoichi Sato. "Mutual Context Network for Jointly Estimating Egocentric Gaze and Actions". In proc. *IEEE Transactions on Image Processing (TIP)*, July 2020.

Other Publications

[1] **Yifei Huang**, Minjie Cai, Ryo Yonetani, and Yoichi Sato. "Temporal localization and spatial segmentation of joint attention in multiple first-person videos". In proc. *IEEE International Conference on Computer Vision Workshops (ICCVW)*, September 2017.

[2] **Yifei Huang**, Minjie Cai and Yoichi Sato. "An Ego-Vision System for Discovering Human Joint Attention". In proc. *IEEE Transactions on Human-Machine Systems (THMS)*, April 2020.

[3] Hong Chen, **Yifei Huang**, and Hedeki Nakayama. "Semantic aware attention based deep object co-segmention". In proc. *Asian Conference on Computer Vision (ACCV)*, December 2018.

[4] Zhenqiang Li, **Yifei Huang**, Minjie Cai and Yoichi Sato. "Manipulation-skill assessment from videos with spatial attention network". In proc. *IEEE International Conference on Computer Vision Workshops (ICCVW)*, September 2019.

[5] Zhenqiang Li, Weimin Wang, Zuoyue Li, **Yifei Huang** and Yoichi Sato. "A Comprehensive Study on Visual Explanations for Spatio-temporal Networks". In proc. *Winter Conference on Applications of Computer Vision (WACV)*, January, 2021.

[6] Hong Chen, **Yifei Huang** and Hedeki Nakayama. "Commonsense knowledge-aware Keyword Planning For Diverse and Informative Visual Storytelling ". To appear. *AAAI Conference on Artificial Intelligenct (AAAI)*, Feburary, 2021.