Technical University of Munich
TUM School of Computation, Information and Technology

Understanding Human Actions: A Graph Convolutional Framework for Intelligent Systems in Human-Robot Interaction

Hao Xing

Complete reprint of the dissertation approved by the TUM School of Computation, Information and Technology of the Technical University of Munich for the award of the

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Chair: Prof. Dr. Hans Michael Gerndt

Examiners:

1. Prof. Dr.-Ing. Darius Burschka
2. Prof. Dr.-Ing. Klaus Diepold

The dissertation was submitted to the Technical University of Munich on 16 April 2024 and accepted by the TUM School of Computation, Information and Technology on 28 November 2024.

Abstract

Understanding human actions plays an important role in intelligent systems, particularly in the interdisciplinary domains of artificial intelligence (AI) and human-robot collaboration (HRC). Understanding human behavior is a key prerequisite for designing and developing systems that interact seamlessly with users. This work delves into the significance of abstracting the human action space through graph representations, employing several attention mechanisms to dynamically adapt and refine the relationships among graph nodes, which represent subjects and objects within the graph structure.

The proposed framework introduces a novel approach that utilizes graph convolutional networks (GCNs) to effectively parse spatial and temporal features embedded within graph representations. By encapsulating action elements (human and objects) within a graph framework, our approach enables seamless switching between classifiers and decoders, thereby facilitating the prediction of labels per clip and the segmentation of sequences into distinct sub-actions. This adaptability and versatility enables the framework to respond to the complexity of tasks with precision and agility. The fusion of graph representations with graph convolutional networks not only enhances the interpretability and robustness of the model but also contributes to a deeper understanding of the dynamics underlying human actions in diverse contexts.

Moreover, our study scrutinizes the uncertainty embedded within graph convolutional networks, trying to bridge the gap between the controlled environment of training datasets and the unpredictability of real-world scenarios. By illuminating and addressing the uncertainties in human actions, our approach aims to facilitate the development of real-time and robust models for human-robot interactions.

The experimental validation of our proposed model are conducted on challenging public human daily actions datasets, as well as real-world datasets. The experimental results underscore the effectiveness and generalizability of our framework, demonstrating its superior performance in accurately recognize human actions across varied contexts and scenarios. These findings not only validate the effectiveness of our approach but also demonstrate its potential applications across a spectrum of domains, ranging from healthcare supervision systems to collaborative human-robot environments.

Overall, this research provides insight into an important topic of understanding human actions within the context of intelligent systems. By leveraging graph representations and graph convolutional networks, our framework not only enhances the interpretability and robustness of human action recognition systems but also promotes a deeper understanding of human action primitives. The insights gained from our study of uncertainty quantification in graph neural networks have positive implications for the design and development of intelligent systems that integrate seam-

lessly with human activities.

Zusammenfassung

Das Verständnis menschlicher Handlungen spielt in intelligenten Systemen eine wichtige Rolle, insbesondere in den interdisziplinären Bereichen der künstlichen Intelligenz (KI) und der Mensch-Roboter-Kollaboration (MRK). Das Verständnis menschlichen Verhaltens ist eine wichtige Voraussetzung für den Entwurf und die Entwicklung von Systemen, die nahtlos mit Benutzern interagieren. Diese Arbeit befasst sich mit der Bedeutung der Abstraktion des menschlichen Handlungsraums durch Diagrammdarstellungen und nutzt mehrere Aufmerksamkeitsmechanismen, um die Beziehungen zwischen Diagrammknoten, die Subjekte und Objekte innerhalb der Diagrammstruktur darstellen, dynamisch anzupassen und zu verfeinern.

Das vorgeschlagene Framework führt einen neuartigen Ansatz ein, der Graph Convolutional Networks (GCNs) nutzt, um räumliche und zeitliche Merkmale, die in Graphdarstellungen eingebettet sind, effektiv zu analysieren. Durch die Kapselung von Aktionselementen (Menschen und Objekte) in einem Diagrammrahmen ermöglicht unser Ansatz einen nahtlosen Wechsel zwischen Klassifikatoren und Decodern und erleichtert so die Vorhersage von Beschriftungen pro Clip und die Segmentierung von Sequenzen in verschiedene Unteraktionen. Diese Anpassungsfähigkeit und Vielseitigkeit ermöglicht es dem Framework, präzise und agil auf die Komplexität von Aufgaben zu reagieren. Die Fusion von Graphdarstellungen mit Graph Faltungsnetzwerken verbessert nicht nur die Interpretierbarkeit und Robustheit des Modells, sondern trägt auch zu einem tieferen Verständnis der Dynamik bei, die menschlichen Handlungen in verschiedenen Kontexten zugrunde liegt.

Darüber hinaus untersucht unsere Studie die Unsicherheit, die in Graph Faltungsnetzwerken verankert ist, und versucht, die Lücke zwischen der kontrollierten Umgebung von Trainingsdatensätzen und der Unvorhersehbarkeit realer Szenarien zu schließen. Indem wir die Unsicherheiten menschlichen Handelns beleuchten und angehen, zielt unser Ansatz darauf ab, die Entwicklung robuster Echtzeitmodelle für Mensch-Roboter-Interaktionen zu erleichtern.

Insgesamt liefert diese Forschung Einblicke in ein wichtiges Thema des Verständnisses menschlicher Handlungen im Kontext intelligenter Systeme. Durch die Nutzung von Graphdarstellungen und Graph Faltungsnetzwerken verbessert unser Framework nicht nur die Interpretierbarkeit und Robustheit menschlicher Handlungserkennungssysteme, sondern fördert auch ein tieferes Verständnis der Grundprinzipien menschlicher Handlungen. Die Erkenntnisse aus unserer Untersuchung der Unsicherheitsquantifizierung in graphischen neuronalen Netzen haben positive Auswirkungen auf den Entwurf und die Entwicklung intelligenter Systeme, die sich nahtlos in menschliche Aktivitäten integrieren lassen.

Acknowledgement

I am very grateful to my supervisor, Professor Burschka, for his patient guidance and giving me this opportunity to make a small contribution in the field of artificial intelligence. Secondly, I would like to thank all sponsors of the project Geriatrnoics (Project X, grant no. 5140951 and Project Y, grant no. 5140953). With this fantastic project, We took a small step forward in the field of healthcare. Last but not least, I would like to thank my family, friends and colleagues for their encouragement when I faced difficulties and bottlenecks.

Munich, February 7, 2024 *Hao Xing*

Contents

1 Introduction **1**
- 1.1 Importance of Human Behavior Understanding . . . 1
- 1.2 Problems in Understanding Human Actions . . . 2
- 1.3 Proposed Solutions and Contributions . . . 7
- 1.4 Structure of the Thesis . . . 11
- 1.5 Publications . . . 11

2 Related Work **13**
- 2.1 Human Action Recognition . . . 13
  - 2.1.1 Action Recognition using Appearance Feature . . . 13
  - 2.1.2 Action Recognition using Skeletal Information . . . 14
  - 2.1.3 Human-Object Interaction Recognition . . . 15
- 2.2 Action Segmentation . . . 17
  - 2.2.1 Attention Mechanism . . . 17
  - 2.2.2 Graph Convolutional Network . . . 18
- 2.3 Exception Event Detection . . . 19
  - 2.3.1 Spatio-Temporal Latent Action Unit Extraction . . . 20
  - 2.3.2 Global Minimization with Robust Cost . . . 20
- 2.4 Uncertainty Quantification . . . 20
- 2.5 Human Tracking . . . 22

3 Approach **25**
- 3.1 Graph Representation of Actions . . . 25
  - 3.1.1 Graph Representation of Skeletal Information . . . 25
  - 3.1.2 Graph Representation of Human-Object Interactions . . . 26
- 3.2 Human Action Recognition using Graph Convolutional Network . . . 27
  - 3.2.1 Adaptively Update of Dynamic Relations between Nodes in Spatial Dimension . . . 28
  - 3.2.2 Temporal Graph Convolutional Layer . . . 33
  - 3.2.3 Hybrid Attention Graph Convolutional Network . . . 34
- 3.3 Human Activities Segmentation using Encoder-Decoder Structure . . . 35
  - 3.3.1 Pyramid Graph Convolutional Network . . . 36
  - 3.3.2 Temporal Fusion Graph Convolutional Network . . . 38
- 3.4 Event Detection using Sparse Coding and Dictionary Learning . . . 42
  - 3.4.1 Task Definition . . . 43
  - 3.4.2 Prepossessing of Data . . . 43
  - 3.4.3 Train Phase . . . 44

3.4.4 Inference Phase 46
3.5 Understanding Human Activity with Uncertainty Measure for Novelty 47
3.5.1 Research Background of Uncertainty Quantification 48
3.5.2 Uncertainty Quantification by Ensemble and Dropout Methods 51
3.5.3 Distance-aware Feature Space by Spectral Normalized Residual Connection 52
3.5.4 Feature Space Distance Measurement using Gaussian Process . 53
3.6 Multiple Objects Tracking 53
3.6.1 Human Tracking 53
3.6.2 Objects Tracking 61
3.7 Real-Time System of Understanding Human-Object Interaction 62

4 Experiments **65**
4.1 Experimental Setup 65
4.1.1 Hardware 65
4.1.2 Datasets 66
4.1.3 Evaluation Metrics 70
4.2 Multiple Objects Tracking 72
4.2.1 Comparison of Detectors 72
4.2.2 Comparison of Feature Extractors 73
4.3 Action Recognition 74
4.3.1 Ablation Study 75
4.3.2 Comparison with State-of-the-Art 78
4.4 Action Segmentation 79
4.4.1 Ablation Study 80
4.4.2 Comparison with the State-of-the-Art 86
4.4.3 Qualitative Results 89
4.5 Event Detection 90
4.5.1 Validating the Effectiveness of GODL 90
4.5.2 Evaluation of Fall-Down using Action Unit and Temporal Structure 92
4.6 Uncertainty Quantification 95
4.6.1 Ablation Study 95
4.6.2 Comparison with State-of-the-Art 97
4.6.3 Qualitative Results 99
4.7 Real-Time System for Understanding of Human-Object Interaction . . . 100

5 Summary **103**
5.1 Conclusion of Proposed Methods 103
5.2 Future Work 105

A Appendix 1 **107**

Bibliography **109**

Chapter 1

Introduction

Advances in robotics have revolutionized various aspects of human life, from industry and healthcare to domestic and personal services. However, designing an intelligent system that can safely and effectively operate alongside humans in unstructured environments requires a major leap forward in the capabilities of existing systems. The ability to understand and predict human actions represents a critical aspect of this leap. It forms the foundation for informed decision-making, enabling robots to anticipate human actions, adapt to their preferences, and contribute constructively to shared tasks.

1.1 Importance of Human Behavior Understanding

Understanding human actions in the context of interactions with robots holds significant motivation and potential benefits across various domains. This motivation comes from the desire to create more intuitive, adaptable, and efficient *human-robot collaborations* and *learning from demonstrations*.

At the level of *human-robot collaboration*, human actions are a primary mode. When robots can accurately recognize and understand human actions, they can better interpret the intentions and needs of humans. This enables smoother and more efficient communication between humans and robots, fostering a natural and intuitive collaboration [KJM10]. In terms of coordinate quality, understanding human actions allows robots to anticipate and synchronize their actions with human actions. This coordination leads to more efficient and harmonious task execution, reducing errors and optimizing workflow. For safety in a collaborative task, recognizing event actions helps robots detect potential hazards, avoid collisions, and respond appropriately to unexpected actions. This adaptability is crucial for maintaining a safe working environment [NK10]. Different individuals perform actions in different ways. By understanding individual-specific action patterns, robots can tailor their responses to specific users, creating a more personalized and comfortable interaction experience [Ond+13]. Understanding human actions can also improve the efficiency of robots in completing tasks. For example, a robot designed to assist people with disabilities can be programmed to recognize and respond to specific movements or gestures made by its user, allowing for a more personalized and efficient assistance experience.

Besides Human-Robot collaboration tasks, understanding human actions helps robots *learn from human demonstrations* as well. By observing and analyzing human

behavior, robots can develop a better understanding of how humans interact with the environment and manipulate objects. Furthermore, robots can develop models that accurately simulate human movements, such as walking, running, or lifting objects. These models can be used to control the behavior of robots during collaboration tasks or training simulations [KD18]. In addition to modeling human behavior, gradual learning is another potential benefit of understanding human actions. Robots can learn new tasks or behaviors by observing human actions, which simplifies the training process and facilitates knowledge transfer. Humans often learn new skills through a process called gradual learning, where they gradually refine their movements and actions over time. By understanding this, robots can adapt their behavior more effectively during training and improve their ability to learn from less-than-ideal demonstrations [Kul+21]. Humans use a wide range of nonverbal cues, such as facial expressions, body language, and tone of voice, to communicate emotions and intent. By understanding these cues in human demonstrations, robots can develop a better understanding of human emotions and respond appropriately during interactions [Trö+21].

Furthermore, understanding human actions plays an important role in the construction of a higher-level semantic map, particularly when it comes to building a semantic map of a person's daily schedule. The identification of recurring patterns and routines in a person's daily schedule contributes to the creation of a semantic map. This map represents not only individual activities but also the broader structure of the daily routine. Armed with knowledge about a person's daily schedule, the robot can navigate proactively to support ongoing or upcoming activities. For instance, it can prepare a workspace before the individual starts working or offer assistance in the kitchen during cooking times. Finally, this ability contributes to the formation of user-centered robotic system. The robot can align its behavior with the user's preferences and rhythms, offering personalized assistance and creating a seamless integration into the user's lifestyle.

Ethical concerns are important issues that cannot be avoided when designing intelligent systems as well. By studying human behavior, developers can identify potential ethical issues related to the use of robots. For example, they may explore how robots should behave in situations where they must make decisions that could impact human lives, such as in autonomous vehicles or military applications [JAT20]. However, this topic will not be addressed in this thesis.

1.2 Problems in Understanding Human Actions

Human action is a rich and complex pattern, influenced by many factors such as speed, target objects, body conditions and environments. Recognizing and segmenting these activities are challenging due to their inherently dynamic, diverse, and context-dependent nature. Traditional robotic systems, with their limited perception and rigid programming, are ill-equipped to comprehend this complexity. In this section, the confronting challenges will be introduced.

The first step of an action recognition system is detecting the performer and related objects in different environments. Note that, for some specific actions, the scene has impacts on the actions label. In this work, we define the location and structures

in the environment as the background, and focus on the daily actions, which have less relation to the background. A complex background can lead to confusion between the background and the action-related subject and objects. Therefore, accurate recognition and segmentation require distinguishing the targets (subjects and objects) from the surroundings. Traditional approaches either manually select interesting areas and track feature points or focus on moving feature points [Sev+19; Lad+20]. Both are not suitable for large-scale data. Recently, with the advantage of Deep Learning, many promising object detection networks have been developed to solve this challenge [Red+16; Lin+20]. These approaches utilize a bounding box to cover the target, which diminishes the effect of the background on the result of action recognition but does not remove it completely. More recently, pixel classification (image segmentation) algorithms have been proposed for classifying pixels belonging to different objects. This technique can be used to completely remove the background, which requires high computational power. Since most of the human body can be viewed as an articulated system with rigid bones connected by joints, which are not sensitive to the background and the appearance of humans, action recognition using skeletal information has been widely investigated and attracted a lot of attention [Shi+19b; YXL18].

Besides distinguishing targets from the surroundings, capturing 3D information is another challenge. Compared to 2D information, 3D information has more extensive spatial features at the cost of higher time-consuming and manual labeling requirements. Most existing research methods still have an ill-posed , inverse problem that extracts 3D information from monocular images [Zhe+20]. The emergence of Microsoft Kinect [Pöh+16], and RealSense [Kes+17] cameras made multidimensional observation of human events feasible without high processing loads on the system. However, the noise of the depth measurement in these cameras has a significant influence on the action understanding.

After motion information is captured by a perception system, extracting motion features is the next step. In the current research work, various parameters such as shape, trajectory, velocity, optical flow, and skeleton have been extensively employed.

Yamato et al. [YT12] integrated silhouette contour features with Hidden Markov Models (HMM) for the purpose of action recognition. Carlsson et al. [LLS09] conducted motion recognition by establishing a shape match between the key frame extracted from the isochronous video and the stored action prototype. The shape information is conveyed through the detected Canny edge data. The advantage of using static contor and shape is intuitive, as numerous 2D image processing methods can be directly applied. However, a notable drawback lies in its dependency on stable segmentation, making it sensitive to factors such as color, light, contrast, and presenting challenges in handling occlusion issues.

Action trajectory is another import representation of motion. Usually the speed and direction of the abrupt point in the trajectory are considered key indicators reflecting distinct movements. In earlier studies [Rah+14; RM16], the KLT tracker was frequently employed for extracting trajectories, with descriptors such as HOF, HOG, and velocity history being utilized. The limitation of relying on trajectory for action identification lies in the challenge of obtaining an accurate trajectory. The difficulty in precisely determining the trajectory constitutes a drawback in utilizing it for action identification.

Optical flow contains a lot of sports-related information, distinguishing itself as one of the most crucial features distinct from static images. Efros [Ke+17] employed the optical flow feature to discern actions within the line of sight. The similarity between actions are measured by the histogram of optical flow (HOF), which is calculated based on the "half-wave rectification" in four directions. The benefit of utilizing optical flow lies in its exemption from requiring background extraction, but the drawback is its challenge in effectively managing background changes.

Recently, skeleton-based action recognition raised a lot attention, in which the action perfromer is depicted as skeleton, and objects are reprensted by the 3D center points [Xin+21]. Since most of human body can be viewed as an articulated system with rigid bones connected by joints, a skeleton representation of human body is an efficient way to simplify the scene. However, this method only focuses on skeleton-related motion and is less accurate when dealing with complex movements, especially interactions with the environment. In this study, we focus on recognizing daily actions, where background has less influence on action labels.

Analyzing the spatial and temporal information to obtain the action label is another challenge. The human action recognition can be regarded as a data classification problem. It demands the development of advanced algorithms that can accurately interpret human behaviors from sensory data, and predict action labels. Most existing methods can be clustered into following categories: key frame based approach, probabilistic based approach, and data driven approach categories.

In the realm of pattern classification, the most straightforward approach for comparing static templates with current samples is through the template matching method. Due to variations in the duration of identical actions, it is necessary to adjust samples in the time dimension. Dynamic Time Warping (DTW) stands out as the most typical method for addressing this requirement [WW17; Tan+18]. The advantages of template matching include simplicity in implementation and swift recognition speed. However, its reliance on the feature space of one or several fixed points to describe the dynamic system limits its ability to accurately reflect the distribution properties of dynamic systems within the feature space. On the other hand, approaches based on probability and statistical models align more consistently with the overall dynamic process of action change.

For the method using probability and statistics, at any moment, a system can be characterized as existing in several independent states. The system transitions to the next state at any given time based on the probability associated with the continuous state. The Hidden Markov Model (HMM) stands out as the most widely employed probabilistic model. The probability and statistics-based method exhibits exceptional robustness to slight variations in the temporal and spatial dimensions of action sequences [YT12]. However, it is sensitive to the quality and quantity of training data. This method faces challenges when encountering diverse and complex real-world scenarios, as they heavily depend on the assumptions made during the modeling process.

Most data driven methods use Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to model respectively human skeleton spatial structure and temporal dynamics. However, both cannot fully represent the spatial and temporal features of the human skeleton at the same time [YXL18]. In these networks, the skeleton input is usually processed as a pseudo-image or sequence of joint coordi-

nate vectors, ignoring the spatial connections between the skeleton joints [Shi+19b]. RNNs are additionally limited by the short memory in analyzing global temporal features. Moreover, it is hard to generalize the graph structure of skeleton data to any random form of skeleton using previous methods.

Recently, the advent of Graph Convolutional Networks (GCNs) has presented a potential solution to address pertinent challenges in spatial and temporal feature representation. Within the GCN framework, spatial characteristics are encapsulated by a spatial graph, comprising joints (vertices) and their inherent interconnections (edges). Likewise, temporal attributes are elucidated through a temporal graph, wherein each vertex is linked to its neighbors across consecutive frames by temporal edges, as articulated in the work by Yan et al. [YXL18].

Traditionally, spatial and temporal edges are delineated based on natural anatomical connections, such as the linkage between the elbow and wrist or the shoulder. However, this conventional approach proves inadequate when applied to activities involving interplay among distinct body parts, exemplified by actions like drinking and eating. These activities exhibit robust correlations between disparate body segments, such as the hands and head. To overcome this limitation and effectively capture the dynamic relations inherent in diverse actions, there arises a necessity for an adaptive mechanism within the computational model. Such an adaptive mechanism would facilitate the nuanced extraction of dynamic relations, especially in scenarios where conventional, static anatomical connections fall short, thereby enhancing the model's capacity to discern and represent intricate patterns of human activities.

Within the realm of human activities, the phenomenon of *human-object interactions* (HOIs) intricately intertwines with the surrounding environment and the objects present within the scene. The imperative task of recognizing HOIs in video sequences stands as a foundational pursuit in comprehending human activities. This involves the meticulous segmentation and recognition of sub-activities on a per-frame basis, achieved through a comprehensive analysis of the interactive relations manifesting between humans and objects [Mor+21]. When humans and objects are represented simplistically through skeletal configurations and center points, these representations inherently give rise to a relation graph that spans both spatial and temporal dimensions. This graph serves as a descriptive framework elucidating the relative positions of human and object entities, as well as the dynamic interplay unfolding throughout the duration of the activity. Capitalizing on the advancements in deep learning within the domain of computer vision, the establishment of a spatial relation graph becomes a tractable endeavor through the adept detection of humans and objects within the scene. Nevertheless, despite the strides made in spatial relation graph construction, unraveling the temporal structure governing sub-actions within a complex task remains a formidable challenge. The intricate temporal dynamics inherent in such scenarios pose a substantial hurdle, necessitating further advancements in computational methodologies to recognize and represent the temporal intricacies of human-object interactions during multifaceted activities.

Currently, the available graph convolutional networks (GCN) [YXL18; Shi+19b] primarily focus on the overall prevalent action being executed, in which only a single action is performed in one set of clips. Leveraging cascaded structures, these methodologies effectively extract and concentrate spatio-temporal features. However, their application is conventionally confined to the task of assigning action labels to spec-

ified segments [PD+20; XB22a]. The pertinent question arises: can the gleaned spatio-temporal information be harnessed to elucidate the temporal structure inherent in activities, specifically, the segmentation of actions?

The quality of an action recognition system relies on the identification of cues that define the label of an action, as well as the spatial-temporal relations demarcating the boundaries between consecutive actions in a task. The proposed approach represents activities in spatio-temporal graphs, where the joints of a human skeleton and the center points of bounding boxes enclosing the objects are graph nodes and graph edges represent the active relations between nodes. The delineation of sub-activities occurs through frame-wise analysis of the evolving connections among graph nodes, as elucidated by Xing et al. [XB22b]. An influential solution employed in processing dynamic Human-Object Interaction (HOI) graph relations is the Attention-based Graph Convolutional Network (GCN). This adaptive mechanism dynamically updates inter-node correlations through an attention mechanism, iteratively parsing features across spatial and temporal dimensions [DWA19; Mor+21]. Combining with a decoder, the processed graph features are further upsampled to the original time scale, and then the graph sequences are classified and segmented frame by frame [XB22b]. However, the persistent challenge lies in surmounting segmentation inaccuracies and mitigating over-segmentation within the temporal dimension, constituting a focal point for ongoing research efforts.

Understanding Human-Object Interactions requires not only the precise recognition and segmentation of interaction relations on a per-frame basis but also an assessment of prediction uncertainty. In critical applications such as robots or autonomous vehicles, safety is essential. Understanding and quantifying for uncertainty helps develop systems that can operate safely, minimizing the risk of accidents or errors. The existing methods [XB22b; Mor+21] show promising performance in terms of recognition accuracy and preventing boundary shifts and over-segmentation. Nevertheless, conventional learning-based models often manifest overconfidence in wrong predictions, a characteristic that proves disadvantageous in real-world scenarios characterized by unforeseen circumstances, such as noise and unknown data. Decision making systems that take network predictions as input need to make choices based on incomplete or noisy information. Uncertainty quantification provides a framework for making decisions that take into account the confidence or uncertainty associated with different pieces of information. For robots operating in human environments, understanding and managing uncertainty is crucial for effective collaboration. It helps in developing systems that can communicate their intentions and make decisions that align with human expectations. The presence of these unpredictable factors increases both the risk and complexity of model deployment. Therefore, the imperative arises for the integration of mechanisms capable of detecting novel human actions, thereby enhancing the adaptability and robustness of our model in the face of diverse and unexpected real-world scenarios.

Another pivotal facet of human-robot interaction pertains to the identification of trigger events necessitating a response from the robot [LP07; Tur+08; Fan+09]. Typically, these events manifest as unforeseen actions or motions executed by the human subject, potentially prompting additional learning of new motions or invoking emergency responses in the event of accidents. In this work, we specifically focus on the prevalent event of detecting instances where people experience fall, arising from

conditions such as tripping or compromised health.

Beyond the detection of event actions, the endeavor to acquire and establish a structured representation of these actions is both essential and intricate. Notably, different actions may share common start and end positions, as well as exhibit similar pose transformations and rotations, exemplified by actions such as lying down and falling. However, their latent temporal feature diverge significantly. Modeling latent spatio-temporal structures of actions is one of the most widely-used techniques for action recognition, and representation [Rab89; WM10; TFK12]. A latent spatio-temporal structure comprises two integral components: an action unit with spatial information and a temporal model. The action unit encapsulates the sequential and constituent elements of the action, while the temporal feature delineates the magnitude of the transition from the previous state to the subsequent state [Qi+18]. In the specific context of the fall-down event, the temporal feature manifests as a discernible, abrupt change in the skeletal height [Ma+14].

The challenges of human action understanding can be briefly summarized as follows:

- Efficiently extract motion targets (subjects and objects) from the image.
- Adaptively update the dynamic relations between human body parts, and between human and objects.
- Accurately recognize human actions and human-object interactions in a clip or framewise.
- Reduce segmentation errors and over-segmentation in the temporal dimension.
- Measure the novelty of predictions and distinguish in-distribution and unknown data.
- Robustly detect event action under the noise in the depth measurement.
- Establish structure representation of the trigger event that can lead to an emergency situation.

By tackling these challenges, this thesis aims to make a significant contribution to the field of robotic vision. The objective is to empower robots with the ability to recognize, segment, and respond appropriately to human activities, thereby facilitating more effective collaboration between humans and robots. Beyond the scope of robotics, the methodologies and insights gleaned from this research could have far-reaching implications, potentially catalyzing advancements in other domains such as intelligent systems, healthcare technology, and autonomous vehicles. In essence, the motivation for this work is to bring us closer to the vision of intuitive, adaptive, and intelligent robotic companions seamlessly integrated into our daily lives.

1.3 Proposed Solutions and Contributions

To effectively extract targets from images, we employ a human pose estimator and an object detector for object bounding box detection. Subsequently, the identified

2D information, such as skeleton nodes and center points, is projected into 3D space along the depth direction.

To dynamically update the evolving relations, we propose an adaptive solution: attention mechanism. Originating from the domain of Natural Language Processing (NLP), the attention mechanism has demonstrated success in discerning potential relationships between words situated at varying positions [Vas+17]. For a similar purpose, we introduce two distinct spatial attention mechanisms, namely simple attention and hybrid attention mechanisms. These mechanisms have the capability to generate novel edges between highly correlated vertices during the training process, thereby autonomously adapting to diverse graph representations of actions and distinct input streams.

In conjunction with the innovative attention mechanism, we introduce a novel graph convolutional network known as the Hybrid Attention-based Graph Convolutional Network (HAGCN). This network incorporates two distinct attention mechanisms tailored for diverse input data. Specifically, the Relative Distance (RD) attention mechanism is particularly beneficial for the bone stream, while the Relative Angle (RA) attention mechanism proves advantageous for action classification related to the joint stream. The resultant model undergoes evaluation on two widely recognized public datasets for human action recognition: NTU-RGBD [Sha+16b] and Kinetics Skeleton [Kay+17]. Impressively, our model demonstrates robust performance on both datasets, affirming its efficacy in the realm of human action recognition.

Regarding the question of action segmentation, we find that it is similar to the difference between image classification and segmentation. In image classification, a cascade structure is commonly employed, facilitating the extraction of global high-level features to classify the entire image as a whole [KSH12]. Conversely, in image segmentation, the emphasis lies in discerning distinctions between pixels by upsampling the cascaded features back to the original scale [Wu+19a].

In this thesis, we introduce a Pyramid Graph Convolution Network (PGCN) designed to enhance HOI recognition and segmentation. This is achieved by integrating the cascaded graph convolutional network with a novel temporal upsampling module, referred to as temporal pyramid pooling (TPP). To address the dynamic interactive relations between humans and objects, we incorporate a spatial attention mechanism within the Graph Convolution Network (GCN). This mechanism dynamically generates new edges between strongly correlated vertices throughout the course of the activity. The efficacy of PGCN is substantiated through its demonstrated framewise recognition and segmentation capabilities, showcasing superior quantitative and qualitative performance on two challenging human-object interaction datasets.

To enhance Human-Object Interaction (HOI) recognition and segmentation, we present the Temporal Fusion Graph Convolutional Network (TFGCN). Comprising an attention-based graph convolutional encoder and a newly devised Temporal Fusion (TF) decoder, this novel architecture aims to improve overall performance. The TF decoder leverages multiple parallel temporal-pyramid-pooling blocks to extract global features and enrich temporal characteristics by fusing high-dimensional features from the encoder with processed low-dimensional features. Experimental results on public datasets demonstrate superior performance in terms of recognition accuracy and the mitigation of boundary shifts and over-segmentation.

Despite these advancements, conventional learning-based models exhibit a ten-

dency to be overly confident in erroneous predictions. Real-world scenarios often involve unexpected situations, such as noise and unknown data, heightening the risks and complexities of application. Consequently, the detection of novel human actions becomes imperative for the effective implementation of our model.

Multi-object tracking algorithms provide valuable insights into addressing the problem, commonly assigning IDs by evaluating the distance between representation features and the existing feature space [WB18]. In essence, this approach necessitates the model to be distance-aware within the representation space, as articulated below:

$$\alpha \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel_X < \parallel g(\boldsymbol{x}) - g(\boldsymbol{x}') \parallel_G < \beta \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel_X \tag{1.1}$$

where g means the graph convolutional layer and maps the input data from manifold X (input space) to the representation space G (feature space), $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two different inputs. The parameters α and β are the lower and upper bounds with a constraint of $0 < \alpha < \beta$. In this *bi-Lipschitz* condition, the upper bound affects the sensitivity of hidden representations to the novel observations (out-of-distribution, OOD), and the lower bound guarantees the distance in hidden representation space for meaningful changes in the input manifold [Liu+20a].

Traditional cascaded convolutional networks provide an upper bound for the hidden representation space distance through normalization and activation functions [RHK18]. However, they suffer from the problem of exploding and vanishing gradients.

Residual connections demonstrate the capability to address gradient issues [VWB16], but they exhibit an expanded bound range and result in indistinguishable features within the representation space for Out-of-Distribution (OOD) detection. To maintain the meaningful isometric property in our deterministic model, we introduce a Spectral Normalized Residual (SN-Res) connection. This connection imposes an upper Lipschitz constraint on the residual flow. We construct an Uncertainty Quantified Temporal Fusion Graph Convolution Network (UQ-TFGCN) with this innovative design, wherein the hidden representation space is confined within a reasonable region. Subsequently, the final label and similarity of unknown data are predicted through maximum likelihood in a Gaussian Process (GP) kernel.

To robustly detect event action under the noise in the depth measurement, we employed a gradual filtering processing on skeleton sequences extracted from RGB images using a lightweight Deep Learning toolbox with aligned depth information.

For the latent action unit extraction, Sparse Coding Dictionary (SCD) is a well-known approach [Chi+13; BDB18; Mai+10]. This method approximates a given video sequence $\mathbf{Y}$ by the manipulation of a low-rank dictionary $\mathbf{D}$ and its coefficient matrix $\mathbf{X}$. Online Dictionary Learning is one of the most successful SCD methods and is widely used in the field of action recognition. As fall event detection represents an extreme case within action recognition, we consider the ODL algorithm as a baseline method in this study. Its cost can be expressed in the least squares problem with a regularizer as follows:

$$\min_{\forall \mathbf{D}_i \in \mathcal{D}, \forall \mathbf{X}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{1}{2} \parallel \mathbf{Y}_i - \mathbf{D}_i \mathbf{X}_i \parallel_F^2 + \lambda \parallel \mathbf{X}_i \parallel \tag{1.2}$$

where F means Frobenius norm, N is the number of action unit and λ is the reg-

ularization parameter. Unfortunately, in the presence of outliers, Eq (1.2) provides a poor estimation for **D** and **X** [Yan+20]. The performance is worse for the 3D skeleton-based human fall event detection because the 3D skeleton has more outlier sources, such as skeleton estimation and depth measurement.

In this study, an attempt to improve event detection latency and temporal resolution is presented and performed in the example of fall detection. We separate the fall event into five latent action atoms "*standing*", "*bending knee*", "*opening arm*", "*knee landing*" and "*arm supporting*".

Overall, all technical contributions are listed as follows:

- We introduce a novel adaptive mechanism that integrates a spatial **Hybrid Attention** (HA) layer, incorporating a mixture of relative distance and relative angle information. Within the framework, the relative distance attention contributes more to the bone stream-related action recognition and the relative angle attention provides more beneficial for the classification of the joint stream related action.

- We present a novel **Pyramid Graph Convolution Network** (PGCN) that leverages a unique temporal pyramid pooling module, thereby extending the capabilities of Graph Convolutional Networks (GCNs) for action segmentation tasks.

- We propose a **Temporal Fusion Graph Convolution Network** (TFGCN), which incorporates a novel temporal feature fusion module to enhance the capabilities of Graph Convolutional Networks (GCNs) to understand human-object interactions.

- We extend the capabilities of the model by introducing a novel **Spectral Normalized Residual connection** (SN-Res), which helps to preserve input distance in the representation space and enables the model to estimate prediction uncertainty.

- We propose a novel **Gradual Online Dictionary Learning** method that uses Graduated Non-convexity (GNC) with Geman McClure (GM) cost function to decrease outlier weight during training.

- Additionally, we design and implement a **real-time system** for understanding human actions, thereby contributing to the safety in decision-making systems and facilitating human-robot collaboration tasks.

To evaluate the proposed contributions, we conduct experiments on several challenging, public human daily action datasets, including two pure human action datasets: NTU-RGBD [Sha+16b] and Kinetics Skeleton [Kay+17], two human-object interaction datasets: Bimanual Actions dataset [DWA19], IKEA Assembly dataset [Ben+20]. Compared to other current action recognition and segmentation approaches, our models achieve the best performance on all datasets in terms of accuracy, robustness, and novelty estimation.

1.4 Structure of the Thesis

The subsequent sections of this thesis are meticulously structured to provide a comprehensive exploration of the proposed framework. In Chapter 2, a brief review of existing approaches in human action recognition, action segmentation, fall event detection, uncertainty quantification, and human tracking techniques is presented. This foundational overview establishes the context for our novel contributions by illuminating the state-of-the-art methodologies in the relevant domains.

Chapter 3 serves as the center of our contributions, revealing the intricacies of the proposed framework. It describes the utilization of graph representations for abstracting the elements of human actions, the employment of graph convolutional networks for human action recognition, the architecture for human activities segmentation involving encoder-decoder structures, fall event detection through sparse coding and dictionary learning, the incorporation of uncertainty measures for discerning novel human activities, human tracking facilitated by a multi-object tracking approach with the Kalman-filter algorithm, and the realization of a real-time system for comprehending human-object interactions.

The subsequent exploration in Chapter 4 provides a detailed account of the experimental validations performed to assess the effectiveness of the proposed model. This chapter lists the experimental results, offering an in-depth analysis of the impact of attention mechanisms, graph representations, encoder-decoder setups, and uncertainty quantification techniques on the overall performance of the framework. Rigorous experimentation and thorough discussions are presented, revealing the strengths and limitations of each component, thus contributing valuable insights to the scientific community.

Finally, in Section 5, a comprehensive summary is provided, outlining the substantive contributions of this study. Additionally, this section provides a discussion of potential future research directions, building upon the identified shortcomings and challenges encountered during this investigation. This forward-looking discussion provides researchers and practitioners with a roadmap for future work in the dynamic and evolving field of human action recognition in intelligent systems.

1.5 Publications

Journals

1 **H. Xing**, Darius Burschka. "Understanding human activity with uncertainty measure for novelty in graph convolutional networks." The International Journal of Robotics Research (2024): 02783649241287800.

Conferences

1 **H. Xing**, D. Burschka "Skeletal Human Action Recognition using Hybrid Attention based Graph Convolutional Network". 2022 26th International Conference on Pattern Recognition (ICPR), IEEE. 2022, pp. 3333-3340.

2 **H. Xing**, D. Burschka. "Understanding Spatio-Temporal Relations in Human-Object Interaction using Pyramid Graph Convolutional Network". 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2022, pp. 5195-5201.

3 **H. Xing**, Y Xue, M Zhou, D Burschka. "Robust Event Detection based on Spatio-Temporal Latent Action Unit using Skeletal Information". 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2021, pp. 2941-2948.

4 M. Tröbinger, A. Costinescu, **H. Xing**, J. Elsner, T. Hu, A. Naceri, L. Figueredo, E. Jensen, D. Burschka, S. Haddadin. "A Dual Doctor-Patient Twin Paradigm for Transparent Remote Examination, Diagnosis, and Rehabilitation". 2021 International Conference on Robotics and Automation (ICRA), IEEE. 2021, pp. 2933-2940.

Publications not Included in this Thesis

1 **H. Xing**, Y. Cao, M. Biber, M. Zhou, D. Burschka. "Joint Prediction of Monocular Depth and Structure using Planar and Parallax Geometry". Pattern Recognition (PR) 130 (2022): 108806.

2 K. Wu, EH. Chen, **H. Xing**, F. Wirth, K. Vitanova, R. Lange, D. Burschka "Adaptable Action-Aware Vital Models for Personalized Intelligent Patient Monitoring". 2022 International Conference on Robotics and Automation (ICRA), IEEE. 2022, pp.826-832.

3 Q. Wang, **H. Xing**, Y. Ying, M. Zhou. "CGFNet: 3D Convolution Guided and Multi-scale Volume Fusion Network for fast and robust stereo matching". Pattern Recognition Letters 173 (2023): 38-44.

4 Y. Wu, X. Su, D. Salihu, **H. Xing**, M. Zakour, C. Patsch. "Modeling Action Spatiotemporal Relationships using Graph-Based Class-Level Attention Network for Long-Term Action Detection". 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2023, pp. 6719-6726.

Chapter 2

Related Work

In this chapter, we briefly review the existing works on human action recognition, human activity segmentation, event detection and uncertainty quantification technique.

2.1 Human Action Recognition

Human action recognition can be categorized into two clusters: using appearance feature and using skeletal information.

2.1.1 Action Recognition using Appearance Feature

Human action recognition using appearance feature relying on extracting motion features from images. One of the most important feature is the change of silhouette. Venkatesha and Turk [VT10] introduced a local shape descriptor to represent turning change and distance cross the shape of the adjacent contour, as shown in bottom of Fig 2.1. Ma *et al.* [Ma+14] extracted curvature scale space features from the silhouettes and mapped them into bag of words space and analyze the word pattern in frequency domain. Horimoto *et al.* [HAT03] project hand shape to an eigenspace with predefined eigen bases. With development of hardware, the contour is easily obtained by depth or event images. More recently, Antonik [Ant+19] utilized the histograms of oriented gradients (HOG) algorithm to extract spatial and shape information from motions and further classify these feature by a recurrent neural network. Jalal *et al.* [Jal+12] generate descriptor from depth silhouettes using R transformation, principle component analysis and linear discriminant analysis. Plizzari *et al.* [Pli+22] introduced a channel attention network to extract motion features from event image sequence.

Another important feature of motion is optical flow from adjacent images as it is invariant to appearance, as demonstrated in top of Fig 2.1. Sevilla-Lara *et al.* [Sev+19] has analyzed the influence of optical flow on the performance human action recognition, and found that jointly learning optical flow and action minimizes the action recognition error. Ladjailia *et al.* [Lad+20] proposed an optical flow descriptor by deriving features from the motion. De *et al.* [De +17] introduced a fall down detection system which combines optical flow with depth contour.

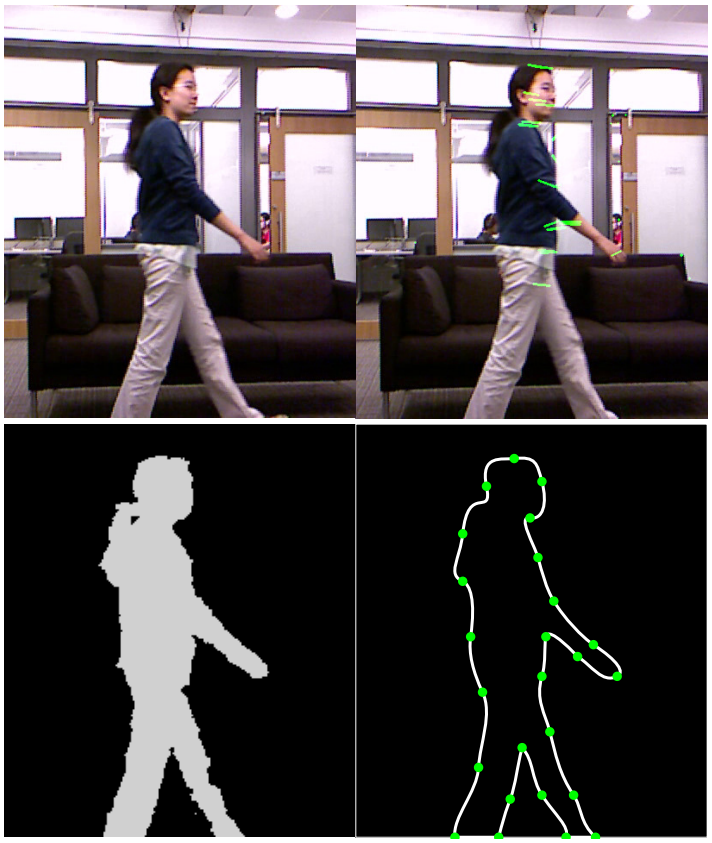


Figure 2.1: The optical flow (top) and the depth contour (bottom) of a *walking* example from MSRDailyActivity3D dataset [Wan+12].

2.1.2 Action Recognition using Skeletal Information

Since most of human body can be viewed as an articulated system with rigid bones connected by joints, a skeleton representation of human body is an efficient way to simplify the scene and action recognition, as shown in Fig 2.2.

Human action recognition using skeletal information can be categorized into two clusters: handcrafted feature based methods and learning based methods. The first method manually designs several features to model human body. Vemulapalli *et al.* [VAC14] represented the action sequence as a curve in Lie group $SE(3) \times \cdots \times SE(3)$, which can be mapped into its Lie algebra and form a feature vector. Hussein *et al.* [Hus+13] constructed a matrix descriptor according to covariance of each joints. Fernando *et al.* [Fer+15] adopted a ranking machine to extract the appearance feature changing of frames evolves with time. In our previous work [Xin+21], we modeled the fall down event with spatial action unit and temporal height change of skeleton, which has a good performance on single event detection. However, these methods are barely satisfied for large-scale multi-class action recognition, because of the complexity of action space.

Recently, with the success of data driven methods, Deep Learning methods have been widely applied in the field of human action recognition. These approaches are mostly using Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNN-based methods convert each skeleton frame to a pseudo image using designed transformation strategy [Li+21a]. Baradel *et al.* [BWM17] combined hu-

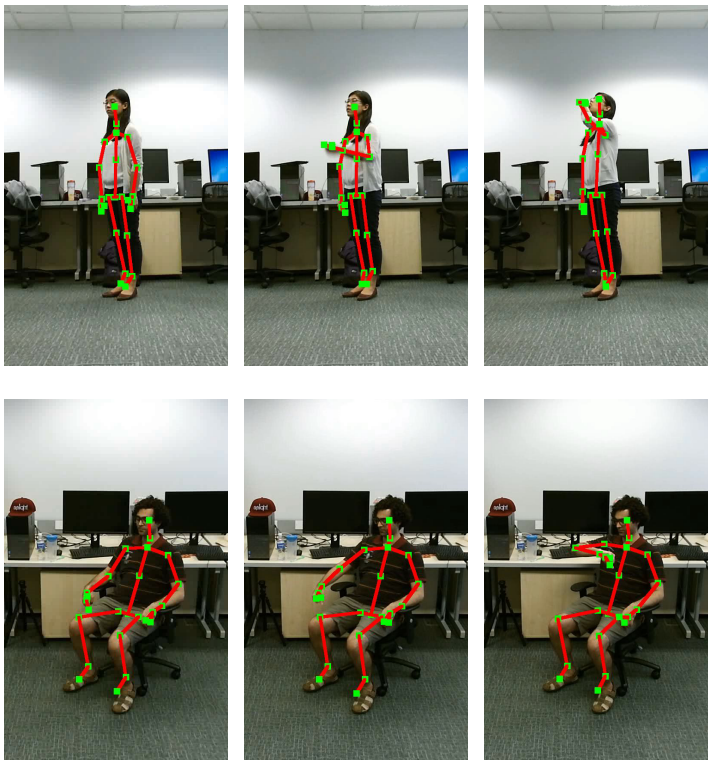


Figure 2.2: Skeleton representation of action examples *drinking* (top) and *watching time* (bottom) from NTU-RGBD dataset [Sha+16b]

man skeletal information with RGB images, which can offer richer contextual cues for action recognition. The authors introduced also a spatial hands attention mechanism, which crops the image around hands. RNN-based methods emphasize the temporal dynamics of skeleton joints [WW17]. Zhang *et al.* [Zha+17] proposed an adaptive RNN model, which can adjust to the most suitable observation viewpoints for cross-view action recognition. Si *et al.* [Si+19] reformed the input skeleton information into the graph-structured data through a graph convolutional layer within the Long Short-Term Memory (LSTM) network.

However, both CNN-based and RNN-based methods cannot fully model human action spatial features and temporal features [YXL18]. Both methods ignore the spatial connections between joints, and RNN-based methods suffer from short-memory in analyzing global temporal features.

2.1.3 Human-Object Interaction Recognition

As part of action recognition, the human-object interaction recognition task aims at detecting the interaction label between human and object for a whole trimmed action clip. Feichtenhofer et al. [FPZ16] introduced a two-stream 2D CNN that utilizes features from both appearance in still images and stacks of optical flow. In a more recent work [CZ17], authors proposed a two-stream inflated 3D CNN (I3D) that improves the ability of 2D CNNs in extracting spatial-temporal features. Dreher et al. [DWA19] presented a graph network that uses three multilayer perceptron (MLP) blocks to update nodes, edges and aggregation features from graph represen-

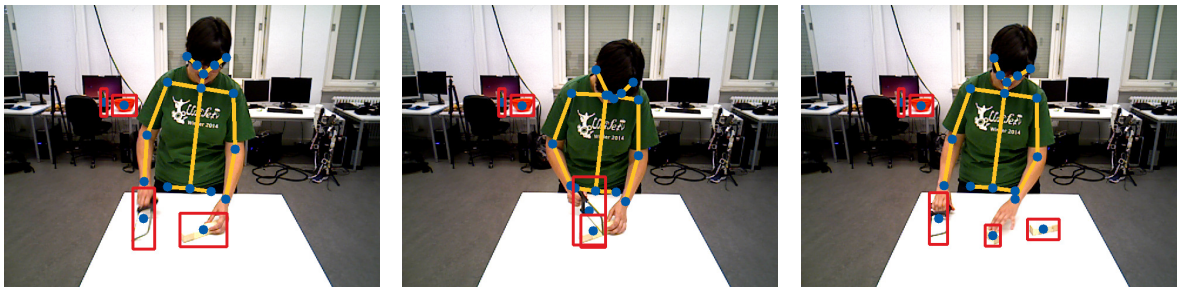


Figure 2.3: A *sawing* example of human-object interaction from the Bimanual Actions dataset [Kre+21]

tation of HOI. Authors also published their HOI dataset, namely Bimanual Actions dataset. Asynchronous-Sparse Interaction Graph Networks (ASSIGN) [Mor+21] is a recent attempt on the HOI recognition task. It used a recurrent graph network that automatically detect the structure of interaction events associated with entities of a sequence of interaction, which are defined as human and objects in a scene. However, the short-term memory of recurrent networks limits their performance in analyzing global temporal structures. In order to expand the receptive field, we adopt dilated convolution layers [Wan+16] in the head of our temporal pyramid pooling module, which constrains the implementation in real-time scenarios as it requires relations from future. [Lag+23] exploited spatial and temporal hand-object relations by leveraging an encoder-decoder framework with graph neural networks. The network can recognize the hand action label and forecast the next motion by a multilayer perceptron module. However, the authors represented human appearance features by a single graph node, which weekends the performance of action recognition. [Tra+23] modeled human-object interactions through a long-term activity route (persistent process) and short-term sub-actions (transient processes). Instead of recognizing action labels, authors focus on the 2D/3D trajectory prediction of the whole activity. Besides single-person action recognition, multi-person involved HOI understanding is another important task. To address the occlusion issue in multi-person actions, [Qia+22] combined both visual and geometry HOI features together and processed features through a Two-level Geometric feature-informed Graph Convolutional Network (2G-GCN). [Rei+22] represented individual actions as graph nodes, interactions between people as graph edges, and finally extracted team intentions from graph features.

Many existing recurrent networks showcase commendable real-time performance but are constrained by limited short-term memory. To address this challenge, the encoder-decoder structure emerges as a promising solution, offering a comprehensive field of view. In contrast to multi-person intent recognition, single-person action recognition represents a more fundamental and challenging task.

Inspired by the effective implementation of the temporal pooling decoder, this study embraces the encoder-decoder structure to improve the performance of single-performer action understanding. The approach entails extracting global features through the temporal pooling module and subsequently fusing condensed features into the temporally pooled features.

2.2 Action Segmentation

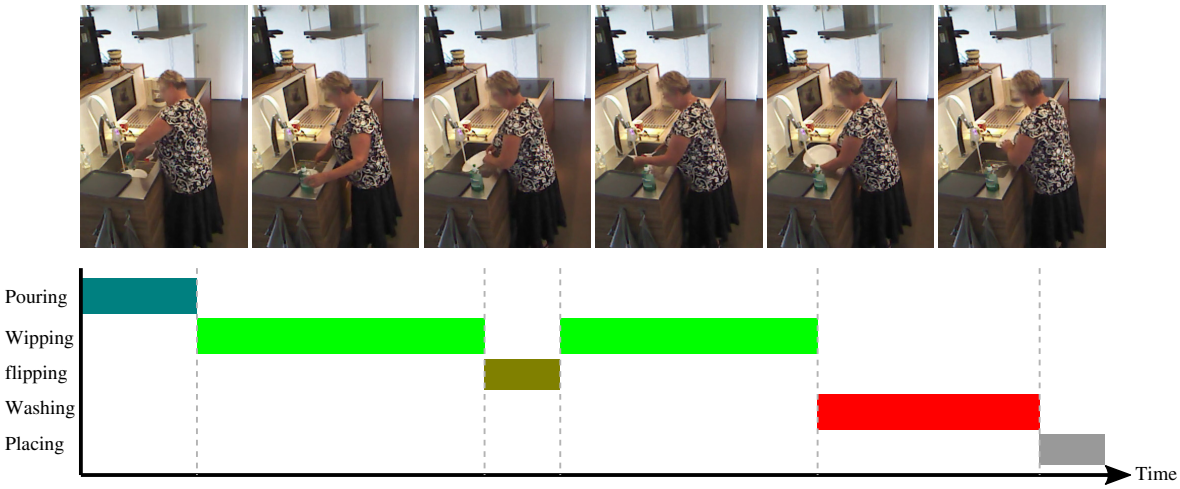


Figure 2.4: Action segmentation of an example of *cleaning dishes* from the Toyota Smart Home dataset [Dai+22]

Different from action recognition with trimmed clip, action segmentation aims to segment activity into sub actions by exploring the temporal structure [SWC16], as shown in Fig 2.4.

As part of earlier works, the hidden Markov model (HMM) is often used to find activity temporal structure. Yamato et al. [YOI92] applies a HMM framework to segment human actions using binary silhouettes of human. Pantic et al.[PP06] introduced a facial profile recognition scheme combining with HMM to segment facial actions. Some other approaches [ZI06; Xin+21] segment action using a sliding window and comparing the similarity between multiple temporal scales. More recently, convolutional neural networks (CNNS) and recurrent neural networks (RNNs) were main streams for action segmentation. For instance, Shou et al. [SWC16] proposed a multi-stage CNN model to classify and localize sub-actions in untrimmed long sequence. Fathi et al. [FR13] segment human activities by identifying state changes of objects and materials in the environment using a RNN model. Motivated by the success of temporal convolution in Nature Language Process (NLP) area, many works applied various temporal convolution networks for action segmentation task, such as dilated temporal convolution [HGS19], encoder-decoder temporal convolution [Lea+17]. Very recently, attention mechanism from transformer has been successfully applied to action segmentation [Zhe+21], due to its strong ability of extracting global information. However, the attention mechanism requires known number of involved objects and subjects to define the size of adjacent matrix.

2.2.1 Attention Mechanism

Attention based neural networks have been successfully applied in NLP and image description. In the field of NLP, the multi head self-attention layer generates the representation of a sequence by aligning words in the sequence with other words

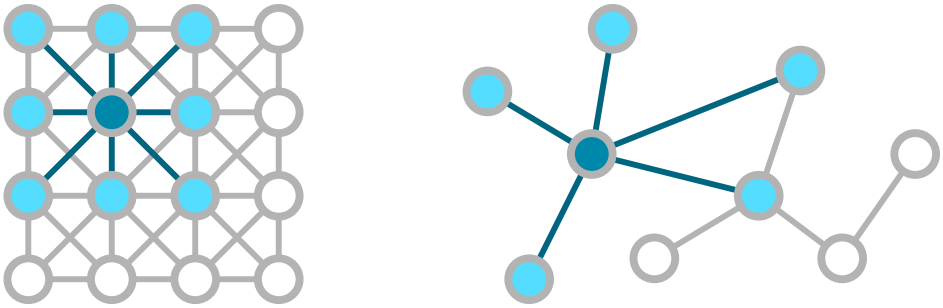


Figure 2.5: Comparison of traditional convolutional kernel (left) with graph kernel (right). In a traditional kernel, the center node (dark blue) has a fix number of connected nodes (light blue), while while the number of connected nodes in the graph kernel changes according to the defined adjacency matrix.

[Vel+18]. Vaswani *et al.* [Vas+17] employed a local attention mechanism on each node and its neighbor pairs in parallel, so that the spatial feature from each neighbor node is weighted by the relative relationship. Devlin *et al.* [Dev+19] extended a self-attention layer bidirectionally, which can model many downstream tasks in text processing.

In the field of image description, the attention mechanism is adopted to generate a learnable weight mask in spatial domain, which demonstrates the importance of a region [Xu+15]. Liu *et al.* [Liu+17] adopted an attention correctness mechanism to generate the attention mask for a corresponding image area. Anderson *et al.* [And+18] combined a top-down attention based CNN with a bottom-up Fast R-CNN to determine feature weightings for each detected region.

As part of natural language, human actions also have strong attention relations between different body parts, such as relative distances and relative angles. Inspired by aforementioned great previous works, we attempt to improve the performance of the graph convolutional networks on Human Action Recognition by designing a novel attention mechanism.

2.2.2 Graph Convolutional Network

Recently, Graph Convolution Networks (GCN) designed for structured data representation raise the attention. Compared to the traditional convolutional networks, the GCN efficiently processes graph features according to a given adjacency matrix. As shown in Fig 2.5, the convolutional kernel process feature in a fixed size, such 3×3, while the graph kernel process feature in dynamic sizes. A challenge for graph convolutional networks is the manual definition of adjacency matrices. In the case of human action recognition, the adjacency matrix can be easily obtained by natural bone connections in human skeleton.

The Graph Convolution Networks (GCNs) can also be categorized into two clusters: spatial and spectral. The spatial GCNs operate the graph convolutional kernels directly on spatial graph nodes and their neighborhoods [Shi+19a]. Yan *et al.* [YXL18] proposed a Spatial-Temporal Graph Convolutional Network (ST-GCN), which extract spatial feature from the skeleton joints and their naturally connected neighbors and temporal feature from the same joints in consecutive frames. Shi *et al.* [Shi+19b] introduced a two stream Adaptive Graph Convolutional Network (2s-

AGCN) based on ST-GCN, which not only extracts features from skeleton joints but also considers the direction of each joint pair (bone information).

The spectral GCNs consider the graph convolution in form of spectral analysis [Li+16]. Henaff *et al.* [HBL15] developed a spectral network incorporating with graph neural network for the general classification task. Kipf and Welling [KW17] extends the spectral convolutional network further in the field of semi-supervised learning on graph structured data. [Lin+23] introduced a bi-stream (joint and bone) spatial graph convolutional network to detect eye contact for conveying information and intent in wild environments.

This work follows the spatial GCNs that apply the graph convolutional kernels on spatial domain.

2.3 Exception Event Detection

Figure 2.6: An example of *fall down* from the NTU-RGBD dataset [Sha+16b].

With the rapid development of motion capture technologies, e.g., single RGB camera systems [Rou+11; De +17; Hua+18; Mir+12; TP13], exception event detection has recently received growing attention because of its importance in the health-care area. An example of *fall down* is demonstrated in Fig 2.6.

For 3D event detection, RGBD cameras, e.g., Microsoft Kinect and Intel RealSense, provide a significant advantage over standard cameras[WSZ19]. Nghiem et al. [NAM12] proposed a method to detect falling down, based on the speed of head and body centroid and their distance to the ground. Stone et al. [SS14] used Microsoft Kinect to obtain person's vertical state from depth image frames based on ground segmentation. Fall is detected by analyzing the velocity from the initial state until the human is on the ground. In contrast with using depth images directly, Volkhardt et al. [VSG13] segmented and classified the point cloud from depth images to detect fall events.

Since depth-based methods are sensitive to the error of shape and depth [WSZ19], many researchers prefer 3D skeleton-based methods. Tran [LM+14] computed three states (distance, angle, velocity) from Kinect's 3D skeleton and applied support vector machine (SVM) to classify falling down action. Kong et al. [Kon+18] applied Fast Fourier Transform (FFT) to classify the 3D fall event skeleton dataset. However, the 3D skeleton estimation using a monocular camera is an ill-posed and inverse problem [Zhe+20].

2.3.1 Spatio-Temporal Latent Action Unit Extraction

Based on sparse coding and dictionary learning method, falling down action can be represented as a linear combination of dictionary elements (latent action units). After Mairal et al. [Mai+10] proposed an Online Dictionary Learning algorithm. It has attracted a lot of attention because of its robustness [Chi+13; Fer+17; Qi+18; WSM14]. Ramirez et al. [RSS10] proposed a classic Dictionary Learning method with Structured Incoherence (DLSI) considering the incoherence between different dictionaries as part of the cost, which could have shared atoms between dictionary. In against sharing dictionary, Yang et al. [Yan+11] presented Fisher Discrimination Dictionary Learning (FDDL) using both the discriminative information in the reconstruction error and sparse coding coefficients to maximize the distance between dictionary. In other words, one training data should only be approximated by the dictionary generated from its cluster. Kong et al. [KW12] separated the dictionary into Particularity and Commonality and proposed a novel dictionary learning method COPAR. With the similar idea, Tiep et al. [VM16] developed Low-Rank Shared Dictionary Learning (LRSDL) that extract a bias matrix for all dictionary based on FDDL. However, its performance is limited for action recognition because each action unit should have a different action space. The results are discussed in the evaluation chapter.

Recently, spatio-temporal deep convolutional networks [PCM20; Wen+19; YXL18; CZZ20; Li+20a] have been widely applied for action recognition. The common principle of these works is that using several continuous frames generate temporal information around feature joints. However, the size of temporal block is a tricky problem among different actions. Besides that, some events have a strict sequence, such as fall down starts from standing (sitting) and ends on the ground. Most of the deep learning networks cannot identify the sequence by summing all temporal blocks note.

2.3.2 Global Minimization with Robust Cost

Global minimization of ODL is NP-hard with respect to both outliers and chosen of regularization parameters. RANSAC [FB81] is a widely used approach but does not guarantee optimality and its calculation time increases exponentially with the outlier rate [Yan+20]. The Graduated Non-convexity has also been successfully applied in Computer Vision tasks to optimize robust costs [Nie95][RC90]. However, with a lack of non-minimal solvers, GNC is limited to be used for spatial perception. Zhou et al. [ZPK16] proposed a fast global registration method, which combines the least square cost with weight function by Black-Rangarajan duality. Yang et al. [Yan+20] applied this method to 3D point cloud registration and pose graph estimation.

2.4 Uncertainty Quantification

Most existing learning based classification methods have high accuracy in in-distribution datasets, but suffer from overconfidence on out-of-distribution (OOD) data and barely detect out-of-distribution samples, as demonstrated in Fig 2.7.

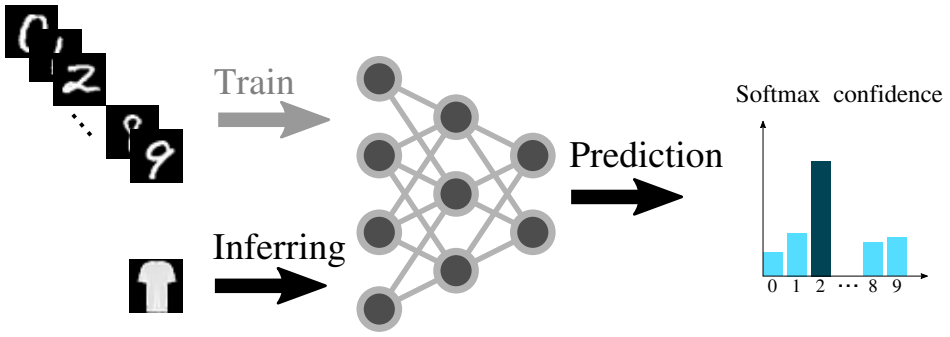


Figure 2.7: A traditional network is trained on MNIST dataset [LeC+98] and predicts a *T-shirt* example from the Fashion MNIST dataset [XRV17] as number 2.

Many promising works train Bayesian neural networks to approximate uncertainty [AC17; Blu+15], but it is difficult to converge in a large range of data. [LPB17] introduced a method that estimates the prediction uncertainty by ensembling predictions from multiple models. [GG16] proposed Monte Carlo Dropout (MC-Dropout) to approximate the Bayesian probability. Due to their reliability, these two methods are usually considered as baselines for uncertainty quantification, although they are time consuming.

Recently, deterministic network uncertainty quantification (DUQ) methods are proposed to efficiently estimate the prediction uncertainty in a single forward pass. The key factor is distance awareness in the representation space. [Van+20a] adopted two sides *Lipschitz* constrains to enforce the gradient smoothness and sensitivity to meaningful changes. Follow the two sides *Lipschitz* constrains, [Liu+20a] considered the residual connections without kernels as natural low *Lipschitz* bound and use spectral normalized kernel instead of normal convolutional kernels on the mainstream to constrain the weight update max gradient and guarantee distance awareness in feature space. However, the residual connections in GCNs have processing kernels and shift the *Lipschitz* bounds to a higher range. More evidence are listed in the experiment section. However, general Spectral Normalization models exhibit a substantial number of trainable parameters and necessitate considerable computational resources.

In this study, we note that the inclusion of a residual connection plays a crucial role in preserving input distances within proximity while also offering the advantage of fewer trainable parameters compared to mainstream approaches. However, this comes at the cost of an increase in the *Lipschitz* bounds. To augment the effectiveness of distance awareness in the feature space, we introduce Spectral Normalization to the residual connections (SN-Res). Further empirical evidence supporting this observation is presented in the experimental section.

To quantify uncertainty, the Gaussian Process has been widely employed. In [Liu+20a], the Gaussian Process prior distribution was approximated by a learnable neural kernel, and the likelihood was subsequently obtained through the Laplace approximation. While this approach enhanced the efficiency of likelihood computation, it also compromised the accuracy of the likelihood.

In another work, [Li+21b] improved flash radiography reconstruction by identifying and removing outliers with high uncertainty, estimated using the Gaussian probability density function with a mean of zero and a measured covariance ma-

trix. Additionally, [Su+23] utilized a multivariate Gaussian distribution to estimate the uncertainty of each corner of predicted bounding boxes in a LiDAR point cloud, thereby enhancing the performance of object detection for autonomous vehicles.

Inspired by these previous works, we follow the uncertainty quantification techniques for deterministic networks, and propose a novel efficient spectral normalized residual connection to better balance the distance awareness and sensitivity of our graph convolutional network.

2.5 Human Tracking

Traditional human tracking methods that rely on biological information, such as face recognition [Wri+08] and iris recognition [Ma+04], are often infeasible in nursing homes due to ethical issues. On the contrary, recognition methods based on visual features are often more reliable than recognition methods based on biological information. People's appearance, such as items carried by a person or clothes of pedestrians, can be more reliably used for person re-identification. Note that we assume people will not change clothes within a short period of time ($3-10$ minutes). Traditional methods rely on manual features and cannot adapt to complex environments with large amounts of data. In recent years, with the development of deep learning, a large number of deep learning-based person re-identification methods have been proposed.

The task of human tracking mainly includes two steps: feature extraction and similarity measurement. The traditional method is to manually extract image features, such as HOG (Histogram of oriented gradient) [DT05], SIFT (Scale invariant feature transform) [Low99] and LOMO (Local maximal occurrence) [Lia+15]. After that, use XQDA (Cross-view quadratic discriminant analysis) [Lia+15] or KISSME (Keep it simple and straightforward metric learning) [Kös+12] to learn the best similarity metric. However, the capability of traditional manual feature description is limited, and it is difficult to adapt to large data tasks in complex scenarios. Moreover, in the case of large amounts of data, traditional metric learning methods will also become very difficult to solve.

Before the advent of deep learning technology, early person re-identification research mainly focused on how to manually design better visual features and how to learn better similarity measures. In recent years, with the development of deep learning technology, deep learning based methods for re-identification have been widely used. Unlike traditional methods, deep learning methods can automatically extract better user image features and learn to obtain better similarity measures at the same time. Of course, deep learning based person re-identification methods have also experienced a from simple to complex development process. At first, researchers mainly focused on using the network to learn the global features of a single frame picture. According to the type of loss, it can be divided into representation learning [BCV13] and metric learning [Kul+13] methods. After performance bottlenecks of global features of a single frame picture, researchers begin to introduce local features and sequence features to further research of person re-identification.

In recent years, deep learning represented by convolutional neural networks has achieved great success in the field of computer vision, it defeat traditional methods

in many tasks and even beyond the human level to some extent. On the problem of person re-identification, deep learning based methods can automatically learn complex feature descriptions, and use simple Euclidean distance to measure similarity to achieve good performance [WBP17]. In other words, deep learning can achieve the task of person re-identification end-to-end, which make the problem much more easier. At present, the person re-identification method based on deep learning has greatly surpassed the traditional method in performance. These advantages have made deep learning popular in the field of person re-identification. A large number of related research work has been published in high-level conferences or journals, and the research on person re-identification has also entered a new stage.

However, the current research on person re-identification has many difficulties. Firstly, the resolution of user images is low. Limited by the imaging quality of the monitoring equipment and the distance between the user and the equipment, a large part of the images are very blurry. Therefore, characteristic information needs to be extracted, such as the clothing of the human body, posture and hairstyle. Secondly, environment of monitor changes. Because different videos or images are taken at different locations and times, there are differences in the perspective, illumination, and posture of pedestrians, and there will be huge deviations in the characteristic information of the same user. Thirdly, pedestrians are blocked. In real scenes, pedestrians are usually in an environment with a large flow of people and a complex background. It is difficult to avoid the situation where the user parts are blocked [Ye+21].

User's pictures taken by different cameras have problems such as low resolution, viewing angle, illumination changes, and background occlusion, which will cause certain changes in the appearance characteristics of pedestrians. Another problem is that due to the different viewing angles of the camera and the impact of light changes, the appearance characteristics of different people are often more similar than the appearance of the same person. These difficulties also make people re-identification and general image retrieval problems different. In addition to expanding training dataset and improving network structure, deep learning methods currently design algorithms that can dedicate to person re-identification task in view of these difficulties.

Chapter 3

Approach

This chapter introduces our methods in graph representation of actions, human actions recognition, human-object interaction recognition and segmentation, emergency event detection, and uncertainty quantification in learning-based models.

3.1 Graph Representation of Actions

Most of daily actions occur between body parts or between human and objects, regardless of the background or appearance of people and objects, for example, the action of *drinking* happens between hand, mouth and cup nodes. Furthermore, most of human body can be viewed as an articulated system with rigid bones connected by joints, which are not sensitive to the background and the appearance of human [XB22b], [XB22a]. Therefore, action recognition using skeletal information has been widely investigated and attracted a lot attention.

3.1.1 Graph Representation of Skeletal Information

Conventionally, raw skeleton data is presented as a sequence of vectors, with each vector encapsulating a set of human joint coordinates in 2D or 3D. The definition of a bone involves the difference between its two end joints. In a graph representation, the joint and bone (spatial) information can be conceived as vertices, where their inherent connections form edges, as illustrated in Fig. 3.1 (a). Besides joint and bone information, we also generate their velocity (temporal) graph information, as shown in Fig. 3.1 (b).

The traditional skeleton graph is based on the framework established by ST-GCN [YXL18], which forms a graph using the inherent structure of the human body, illustrated in Fig. 3.1 (a) on the left. Nevertheless, this approach overlooks the robust connections between body parts that often exhibit significant movements, such as hands, head, and feet.

Hence, we introduce additional connections between these body parts, as depicted in Fig. 3.1 (a) on the right. Each connection maintains the same incoming, outgoing, and self-connecting edges as the traditional graph. All edges collectively constitute a binary adjacency matrix $\mathbf{A}init$, where $aij = 1$ signifies that vertices v_i and v_j are connected. Thus, a spatial graph can be formally expressed as:

$$\mathbf{G} = \mathbf{A} \cdot \mathbf{F}_{in} \tag{3.1}$$

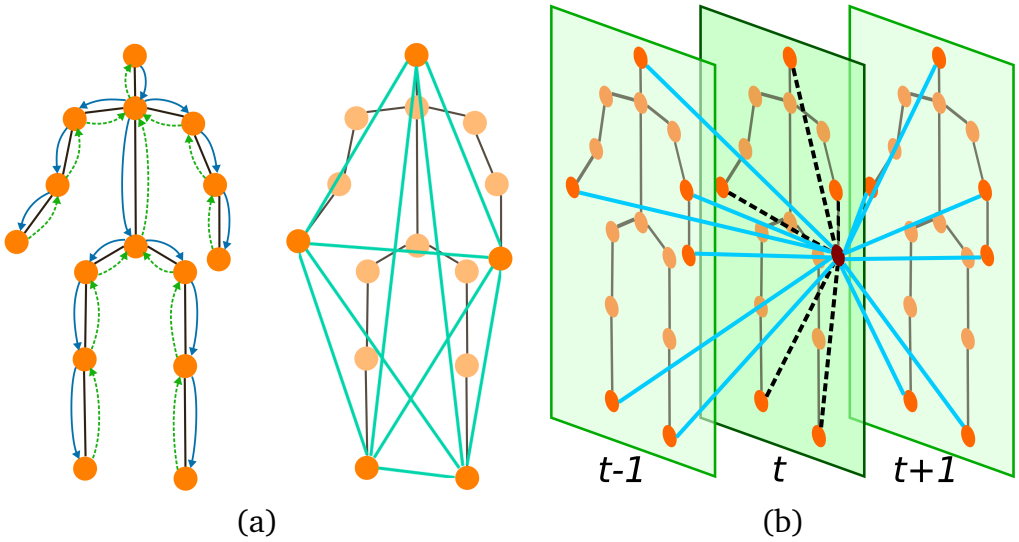


Figure 3.1: Illustration of skeleton graph [XB22a]: (a) Left: Spatial graph with notes and edges (solid arrow: outward edges; dash arrow: inward edges); Right: Additional connections between head, hands and feet; (b) Temporal (solid blue) and spatial (dash black) edges of the left wrist node.

where $\mathbf{F}_{in}$ is the input skeleton feature map, $\mathbf{G}$ is the graph feature map and $\mathbf{A}$ is column-wise normalization of $\mathbf{A}_{init}$. The temporal graph is constructed by connecting vertices and their neighbor pairs in consecutive frames in the same way, as shown in Fig. 3.1 (b).

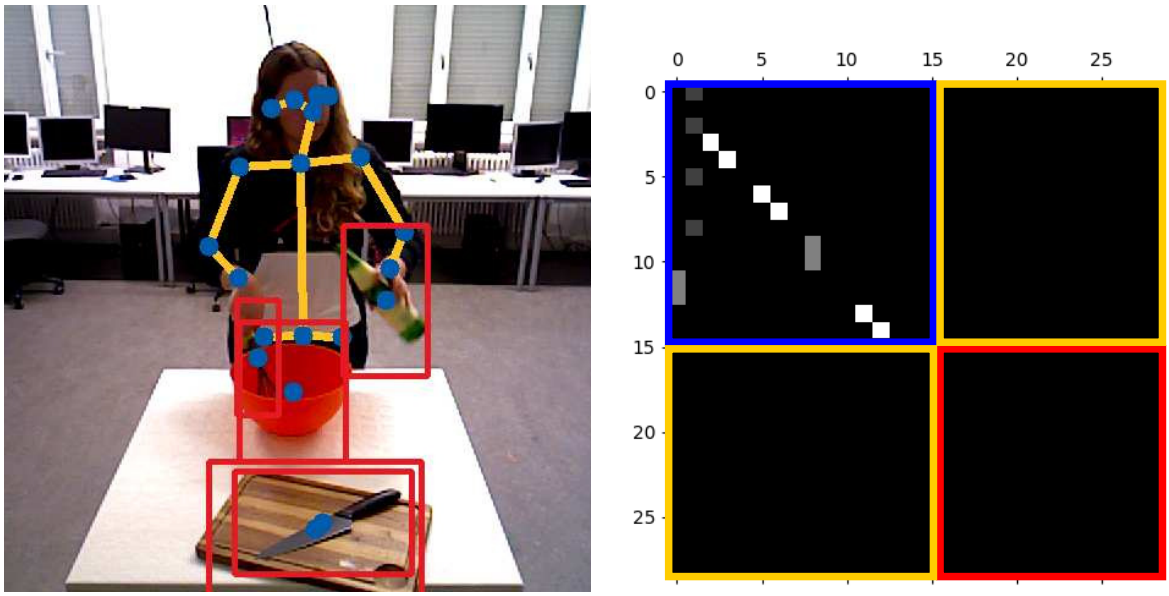
3.1.2 Graph Representation of Human-Object Interactions

Building upon the preceding explanation, we can seamlessly depict a human-object interaction (HOI) scene through skeletons and center points of objects. This representation accurately captures the intricate relationships between individuals and objects, uninfluenced by texture information. The construction of an action graph is subsequently delineated into three distinct components: the human (skeleton) graph, the human-objects graph, and the objects graph.

All skeleton joints and object points serve as vertices in a graph, with their connections represented as edges between vertices. Each vertex possesses inward, outward, and self-connecting edges [YXL18]. The connections among skeleton joints are naturally defined by the pose architecture, featuring inward connections from each joint to adjacent joints that are closer to the center of the body, and outward connections in the reverse direction. However, establishing connections related to objects (human-objects and objects-objects) poses challenges due to the dynamic nature of the scene. In this study, we assume the absence of initial connections between objects-objects and between human-objects joints, as depicted in Fig. 3.2 (a).

All edges collectively constitute a binary adjacency matrix $\mathbf{A}$, where $a_{ij} = 1$ signifies that vertices v_i and v_j are connected from i to j, as mentioned earlier. Given the absence of initial connections between objects-related pairs of vertices, both inward and outward edges remain unoccupied, as illustrated in Fig. 3.2 (b).

Given the initial adjacency matrix of HOI, a spatial scene graph feature map can



(a) (b)

Figure 3.2: The initial spatial relation graph of human-object interaction [XB22b]: (a) Spatial graph with notes (blue) and edges (orange) on example of Bimanual Actions dataset [DWA19]; (b) Initial inwards adjacent matrix with skeleton inward edges (blue block), empty human-objects (orange blocks) and objects-objects edges (red block).

be obtained by the Eq. 3.1. Experimental evaluation of the effectiveness of graph representations is introduced in Section 4.3 and 4.4.

3.2 Human Action Recognition using Graph Convolutional Network

The utilization of skeletal information for action recognition has garnered significant attention and has been extensively explored. This is due to the fact that the human body can be conceptualized as an articulated system, consisting of rigid bones connected by joints. Such an approach is advantageous as it is less sensitive to background interference and the appearance variations of the human body [Shi+19b; YXL18].

Many existing methods employ Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to model the spatial structure and temporal dynamics of human skeletons, respectively. However, both approaches often fall short of fully capturing the concurrent spatial and temporal features of the human skeleton [YXL18]. In these networks, the skeleton input is typically processed either as a pseudo-image or as a sequence of joint coordinate vectors, neglecting the inherent spatial connections between skeleton joints [Shi+19b]. Additionally, RNNs face limitations associated with short-term memory, hindering the comprehensive analysis of global temporal features. Furthermore, prior methods struggle to generalize the graph structure of skeleton data to accommodate diverse forms of skeletal configurations.

Recently, with development of Graph Convolutional Networks (GCNs), a compatible solution has been proposed. In GCNs, spatial features can be represented by a spatial graph, which combines joints (vertices) and their natural connections (edges). Similarly, temporal features are depicted by a temporal graph that connects each vertex with its neighbors in consecutive frames using temporal edges [YXL18]. Typically, these spatial and temporal edges are defined by natural connections, such as those between the elbow and the wrist or the shoulder, which remain consistent across various actions. However, this approach is not well-suited for body-parts related activities, such as drinking and eating, which involve strong relations between different body parts, like hands and the head. To extract dynamic relations in different actions, there is a need for an adaptive mechanism.

In the field of Natural Language Processing (NLP), the attention mechanism has been successfully employed to identify potential relations between words at different positions [Vas+17]. Drawing inspiration from this, we introduce a hybrid spatial attention mechanism in Graph Convolutional Networks (GCN) for a comparable purpose. This mechanism facilitates the generation of new edges between strongly related vertices during the training process, automatically adapting to distinct graph descriptions of actions and different input streams.

3.2.1 Adaptively Update of Dynamic Relations between Nodes in Spatial Dimension

Given the graph representation of actions with an initial adjacency matrix, we define natural spatial connections between nodes, such as body parts. However, the predefined adjacency matrix is inadequate for handling dynamic relations during an activity. Therefore, we introduce an attention mechanism to adaptively update the initial adjacency matrix through the attention score map, as demonstrated in Fig 3.3. The attention map can be generally calculated in the following manner:

$$a_{ij} = \phi\left(\frac{\boldsymbol{f}_i^T \cdot \boldsymbol{f}_j}{\sqrt{n}}\right) \text{ or } \phi(\bar{f}_i - \bar{f}_j) \tag{3.2}$$

where ϕ is an activation function, e.g., *hyperbolic tangent* function, a_{ij} is an element of the attention map $\boldsymbol{A}_{tt}$, and i, j are its index. $\boldsymbol{f}$ represents a vector of feature map, and $\bar{f}$ means the average value of feature vector, and n serves as a normalization factor, typically selected to be the length of the feature vector. Note that the selection of activation function depends on the type of task. In the case of relative distance attention mechanism $(\bar{f}_i - \bar{f}_j)$, the attention scale values are constrained in the range of $[-1, 1]$ by the *hyperbolic tangent* function, where larger absolute values represent larger distance between the inputs, and 0 means the inputs are the same.

A Simple Spatial Attention Mechanism

As crucial dynamic interaction information is absent in the constructed graph, we propose a straightforward attention-based graph network. This network adaptively updates the initial adjacency matrix through the attention score map. The attention

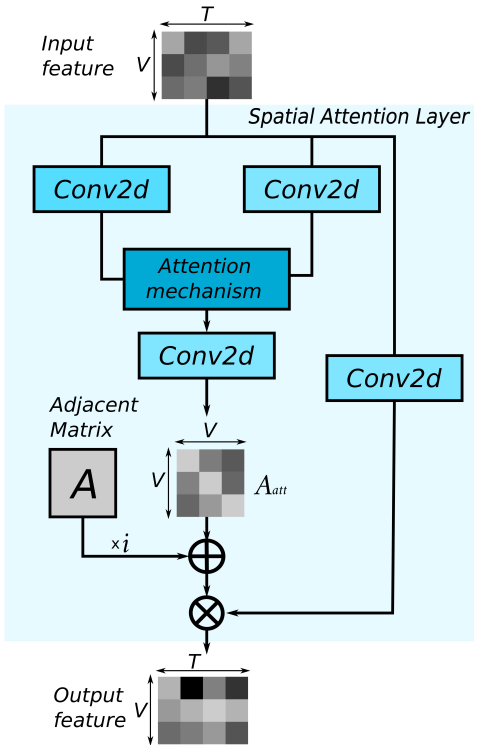


Figure 3.3: General Structure of the Spatial Attention Mechanism: spatial graph convolutional layer with the attention map $\boldsymbol{A}_{att}$ and a trainable parameter a for adaptive update of the adjacency matrix $\boldsymbol{A}$.

score is calculated using the dot product between nodes, as follows:

$$M_{ij} = \frac{f_i \cdot f_j^T}{\sqrt{n}} \tag{3.3}$$

where M is the attention mask map, f is the node feature vector and i, j are the indices of nodes, and n serves as a normalization factor, typically selected to be the length of the feature vector.

In this work, we find that feeding mask maps into a 1-dimensional convolution layer contributes to the relationship learning process. As shown in Fig. 3.4, the input feature map undergoes two parallel 2D convolution layers to produce two output maps of the same size. The dot product of these maps is then processed by a 1D convolution layer with a *sigmoid* activation function, extracting the attention mask. The final attention map is generated through the combination of the attention mask with the adjacency matrix, as follows:

$$\mathbf{A}_{final,i} = \mathbf{M}_i + \hat{\mathbf{A}}_i = \mathbf{W}_i(\mathbf{F}_{1,i}^T \cdot \mathbf{F}_{2,i}) + \hat{\mathbf{A}}_i \tag{3.4}$$

where $\mathbf{M}$ is the attention mask that is extracted by the 1D convolution kernel on the dot product of feature maps $\mathbf{F}_1$ and $\mathbf{F}_2$, $\mathbf{W}$ is the kernel weight, $\mathbf{A}$ is the adjacent matrices and i is the index of the three connection types (*inwards*, *outwards*, *self-connecting*) [Shi+19b]. In order to give more flexibility to the spatial graph, we set adjacency matrices as learnable parameters with given initial values.

In this study, we have discovered that integrating mask maps into a 1-dimensional convolution layer enhances the process of learning relationships. As illustrated in Fig.

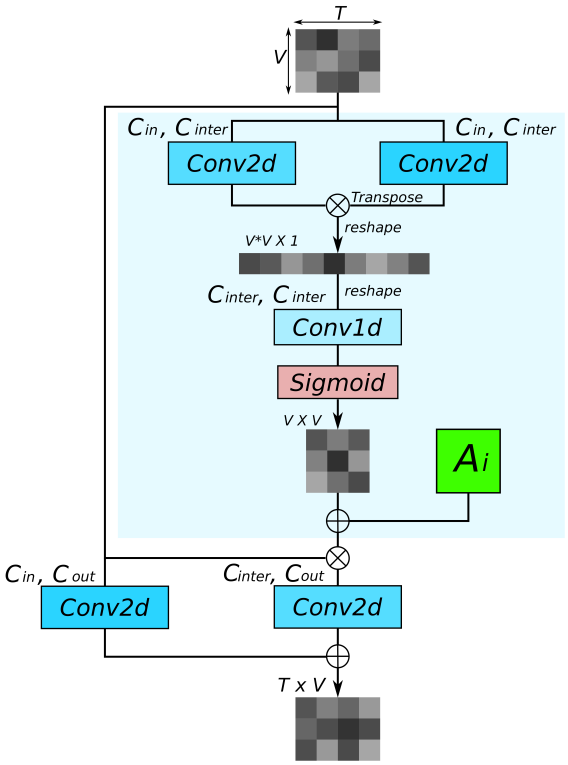


Figure 3.4: Structure of the Spatial Attention Mechanism [XB22a]: a simple attention mechanism with an additional 2D convolutional layer.

3.4, the input feature map undergoes two parallel 2D convolution layers, generating two output maps of the same size. The dot product of these maps is then processed by a 1D convolution layer with a *sigmoid* activation function to extract the attention mask. The final attention map is produced by combining the attention mask with the adjacency matrix, as follows:

$$\mathbf{A}_{final,i} = \mathbf{M}_i + \hat{\mathbf{A}}_i = \mathbf{W}_i(\mathbf{F}_{1,i}^T \cdot \mathbf{F}_{2,i}) + \hat{\mathbf{A}}_i \tag{3.5}$$

where $\mathbf{M}$ is the attention mask that is extracted by the 1D convolution kernel on the dot product of feature maps $\mathbf{F}_1$ and $\mathbf{F}_2$, $\mathbf{W}$ is the kernel weight, $\mathbf{A}$ is the adjacent matrices and i is the index of the three connection types (*inwards*, *outwards*, *self-connecting*). To enhance the flexibility of the spatial graph, we have designated adjacency matrices as learnable parameters, each initialized with predefined values.

The output feature map of the spatial attention layer is expanded to C_{out} output channels via an additional 2D convolution layer, and subsequently integrated with the residual stream. This process can be mathematically articulated as follows:

$$\mathbf{G}_i = \text{Conv2d}(\mathbf{A}_{final,i} \cdot \mathbf{F}_{in}) + \text{res}(\mathbf{F}_{in}) \tag{3.6}$$

where $\mathbf{F}_{in}$ is the input feature map, res is the residual layer, and $\mathbf{G}$ is the $i-$th output graph feature map. The final block output feature map is obtained by summing the outputs of all three types of connections as $\mathbf{G} = \sum_{i=1}^{3} \mathbf{G}_i$

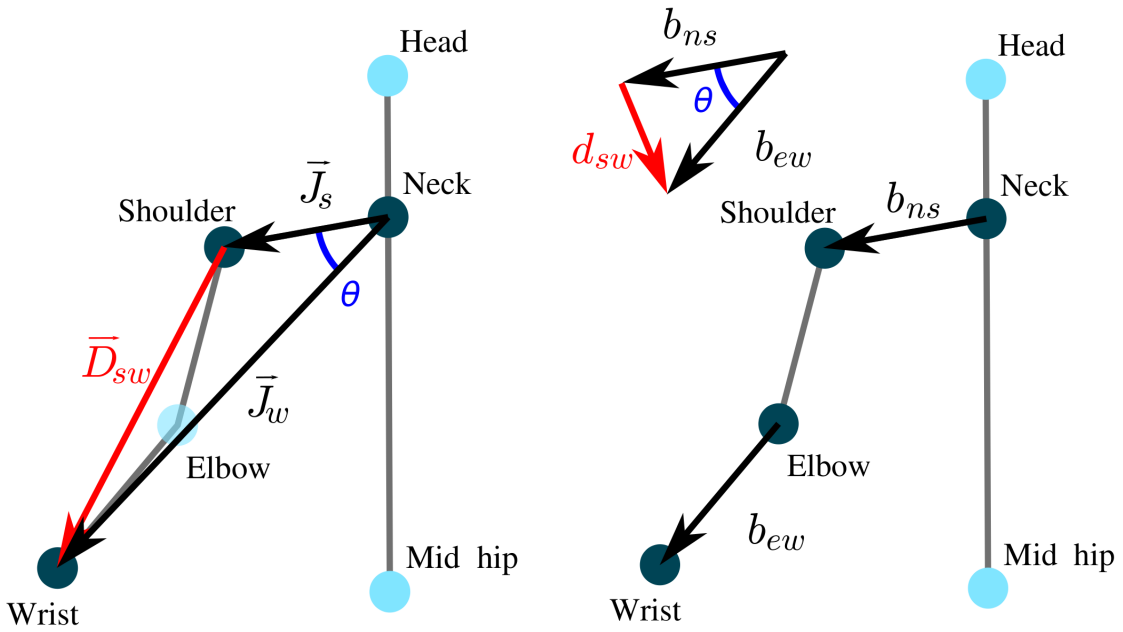


Figure 3.5: Examples of Relative Distance and Relative Angle in the spatial domain over the (a) Joint and (b) Bone streams [XB22a]. In the joint stream (a), the origin point is *neck*, $\boldsymbol{j}_s$ and $\boldsymbol{j}_w$ are the joint vectors of *shoulder* and *wrist*. In the bone stream, $\boldsymbol{b}_{ns}$ and $\boldsymbol{b}_{ew}$ are bone connections of *neck-shoulder* and *elbow-wrist*. The θ is the relative angle, the $\boldsymbol{d}_{sw}$ is the relative distance

Hybrid Spatial Attention Mechanism

The attention layer relates different features of a same input and generates a mask map that contains the importance of each element in feature map. The importance (score) can be expressed as follows:

$$m_{ij} = \text{score}(\boldsymbol{f}_i, \boldsymbol{f}_j) = \frac{\boldsymbol{f}_i^T \boldsymbol{f}_j}{\sqrt{n}} \tag{3.7}$$

where m is an element of mask map $\mathbf{M}$, $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are elements in the feature map, and n is a normalizing parameter, it can be the length of a vector, when $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are column feature vectors. Given such a mask map, typically, a *softmax* function is applied to normalize the scores into range $[0, 1]$. In this study, we observe that incorporating mask maps into a 2-dimensional convolution layer enhances the process of learning relations.

In the spatial dimension, we adhere to the three types of spatial graph structures outlined in 2s-AGCN [Shi+19b], which are generated by identity, inwards, and outwards adjacency matrices, respectively. On each graph, we employ a newly designed hybrid attention layer to extract spatial attention information. The hybrid attention comprises two branches: Relative Distance (RD) attention and Relative Angle (RA) attention, each offering substantial advantages for the bone stream and joint stream, respectively.

For both attention branches, the input feature map undergoes initial compression in the channel dimension through a 2D convolutional kernel. This compression serves to reduce the computational load for attention, enhance feature distinctions

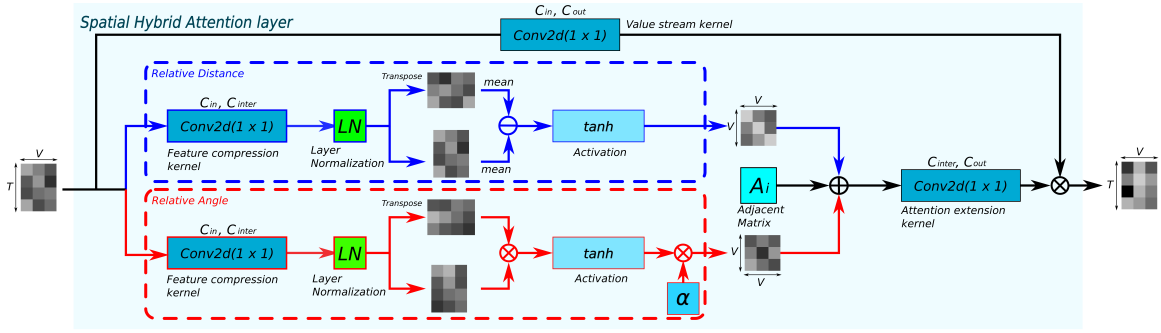


Figure 3.6: Illustration of the spatial hybrid attention convolution layer [XB22a], where blue stream (top) is "*Relative Distance*" attention branch and red stream (bottom) represents '*Relative Angle*" attention branch. C_{in}, C_{inter} and C_{out} stand for input, inter and output channel, respectively. V and T represent the spatial and temporal size of feature (attention) map. "*mean*" is an average process in temporal dimension.

between channels, and facilitate the generation of a unique attention map for each channel. In this study, the compression ratio is set at $C_{inter}/C_{in} = 8$. To ensure stable attention across different distributions of input action cases, Batch Normalization (BN) is typically employed before calculating the attention score. However, this practice can compromise model performance with small-batch sizes, as such batches may not provide a representative distribution of examples [Iof17]. Consequently, we opt for the Layer Normalization (LN) function, allowing each input case to standardize only within its own batch.

As illustrated in Fig. 3.6, the compressed feature map is normalized using the *Layer Normalization* function for each batch. Subsequently, it is inputted into the respective attention function along with its transposed feature map. This process can be expressed as follows:

$$\mathbf{F} = LN(\mathbf{W}_c\mathbf{X} + \mathbf{B}_c) \tag{3.8}$$

where $\mathbf{W}_c \in \mathbb{R}^{1\times1}$ and $\mathbf{B}_c \in \mathbb{R}^{1\times1}$ are parameters of the feature compression kernel, LN is the *Layer Normalization* function, and $\mathbf{X}$ are the input feature map, and $\mathbf{F}$ are the compressed feature map.

As the attention score is computed between nodes, let us exemplify our attention mechanism by considering two node feature vectors, denoted as $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, extracted from the compressed feature map $\mathbf{F}$. These vectors are of size $1 \times T$.

Relative Distance attention: The RD attention information is derived from the relative distance between nodes, formulated as follows:

$$a_{RD,ij} = \tanh((\bar{f}_i - \bar{f}_j)), \text{ with } i, j \in [1, V] \tag{3.9}$$

where a_{RD} is an element of RD attention mask $\mathbf{A}_{RD}$, *tanh* is the *Hyperbolic Tangent* activation function, $\bar{f}$ is the average value of feature vector $\boldsymbol{f}$ over temporal dimension, i, j are the indices of nodes. The final RD attention mask $\mathbf{A}_{RD}$ is with size of $V \times V \times C_{inter}$, where C_{inter} is the number of channel.

Relative Angle attention: The RA attention information is acquired through the channel-

wise dot product between node feature vectors, formulated as follows:

$$\begin{aligned} a_{RA,ij} &= \tanh(\boldsymbol{f}_i^T \cdot \boldsymbol{f}_j) \\ &= \tanh(|\boldsymbol{f}_i||\boldsymbol{f}_j| cos(\theta)), \text{ with } i,j \in [1,V] \end{aligned} \tag{3.10}$$

where the θ is the angle between two vectors. Note that we simplify the dot product attention in Eq. 3.7 by removing the scale $1/\sqrt{n}$, since n is the number of nodes and it is constant. Equation 3.10 yields an RA attention mask of the same size as the RD mask. As demonstrated in the examples presented in Fig. 3.5, it is evident that relative distance and relative angle mechanisms focus on distinct features in joint and bone streams. For certain actions, such as drinking or eating, vector pairs with small relative distance (*head-wrist*) should exert a significant influence on action prediction. In such cases, the relative angle mechanism directs attention to these pairs, given that $cos\theta \approx 1$. Conversely, for other actions like stretching or celebrating, the relative distance attention should play a dominant role in determining attention values.

Since the two attention mechanism (RA and RD) have different effects on different actions, we adopt a learnable parameter α to combine them. The sum hybrid attention score A_h are formed by the following equation:

$$\mathbf{A}_h = \mathbf{A}_{RD} + \alpha \cdot \mathbf{A}_{RA} \tag{3.11}$$

As aforementioned, the predefined adjacency matrix serves as a local attention map, and the spatial attention mechanism adapts to various input action classes, thereby enriching the local attention into a global map. Consequently, the ultimate attention map is produced through the combination of the hybrid attention mask with the adjacency matrix, formulated as follows:

$$\mathbf{A}_{final} = \mathbf{A}_h + \mathbf{A}_i = \mathbf{A}_{RD} + \alpha \cdot \mathbf{A}_{RA} + \mathbf{A}_i \tag{3.12}$$

Note that the initial graph mask $\mathbf{A}_i$ is added in channel-wise, since it is of size $V \times V \times 1$, while the size of the attention masks are $V \times V \times C_{out}$.

In addition to the two introduced attention mechanisms, we have incorporated several other widely adopted attention approaches, including 2s-AGCN [Shi+19b] and CTR-GCN [Che+21]. Further details can be explored in the respective original papers. Experimental evaluation of the effectiveness of different attention mechanisms are introduced in Section 4.3 and 4.4.

3.2.2 Temporal Graph Convolutional Layer

Following the spatial processing, the features of distinct nodes are grouped per frame, and this grouped feature is subsequently processed in the temporal dimension. In this study, we incorporate two distinct temporal graph convolution layers. Initially, we adhere to the ST-GCN [YXL18] approach, employing a 2D convolution kernel with a size of $K_t \times 1$ on $C \times T \times V$ feature maps, where K_t is set to 9 in this work.

The dynamic feature undergoes further processing in the temporal dimension, where we employ a single 2D convolutional kernel with a size of 9×1. However, a fixed-size kernel has limitations when processing long activities due to its restricted

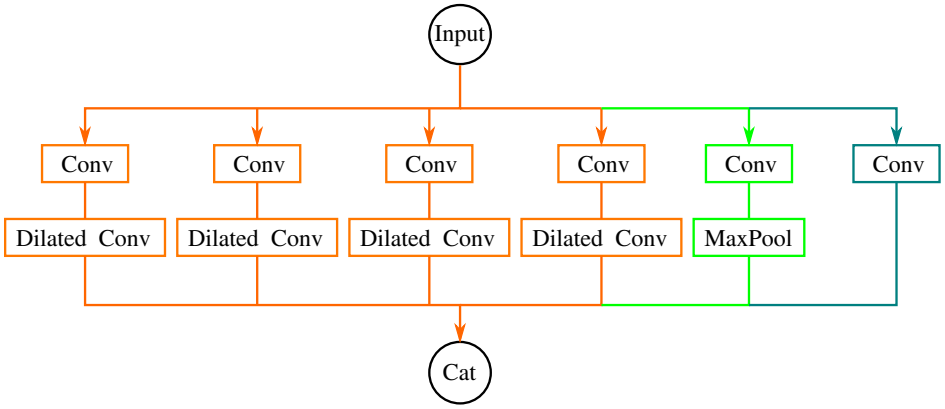


Figure 3.7: Multi-scale temporal graph convolutional layer: four dilated convolutional kernel with different dilation size; a max-pooling branch provides the max feature; additional residual connection offers the raw feature from the input. All feature maps from parallel branches are concatenated (*cat*) into the final output.

receptive field. To address this, we introduce another multi-scale temporal graph convolutional layer. Illustrated in Fig 3.7, this layer consists of six parallel branches. Four dilated convolutional layers with varying dilation sizes enable the capture of temporal features within a larger receptive field. Additional max-pooling and a residual connection are incorporated to extract the maximum value and raw features from the input. In this study, the four dilation sizes are chosen as $1, 2, 3, 5$. Experimental analyse of temporal convolutional layers is introduced in Section 4.3 and 4.4.

3.2.3 Hybrid Attention Graph Convolutional Network

Given the defined spatial and temporal layers, a hybrid attention-based graph convolutional block is constructed. As depicted in Fig 3.8 (a), spatial ($Conv_S$) and temporal ($Conv_T$) layers are succeeded by a batch normalization layer (*BN*) and a *ReLU* activation function. A residual connection is also incorporated alongside the Spatial-Temporal block.

After many tests of training time, number of parameters and accuracy performance, we select the optimal architecture, which has 10 basic blocks with output channels sized as 64, 64, 64, 64, 128, 128, 128, 256, 256, 256 respectively, as illustrated in Fig 3.8 (b). A *BN* function is applied initially to normalize the input data. Subsequently, a global *Average Pooling* (Avg_pool) layer is employed to pool the feature map and reshape it to a uniform size. To mitigate overfitting, an additional dropout layer is introduced with a dropout rate of 0.1. Finally, at the end of the network, a *Softmax* function is applied for the final prediction.

Experimental analysis of the hybrid attention graph convolutional network on human action recognition datasets, the effectiveness of temporal graph convolutional layers, and the adaptability of attention mechanisms are introduced in Section 4.3.

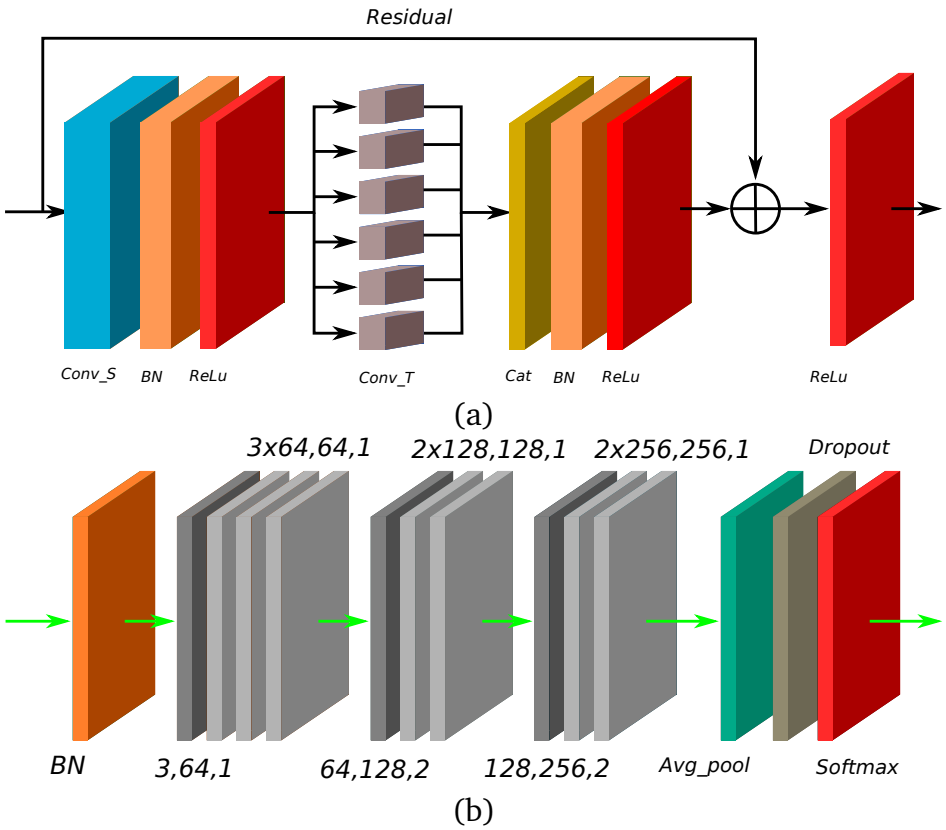


Figure 3.8: Illustration of hybrid attention based graph convolutional network [XB22a]: (a) Hybrid attention based graph convolutional block unit consisting of spatial convolutional layer (*Conv_S*), *Batch Normalization* (*BN*), temporal convolutional layer (*Conv_T*), concatenate function (*Cat*) and *ReLU* activation function; (b) Hybrid attention based graph convolutional network that consists of 10 HA-GCN blocks, where the input channel, output channel and stride parameters are listed besides the block, such as $3, 64, 1$ mean 3 input channel, 64 output channel, and 1 stride, respectively, $2\times$ and $3\times$ represent 2 and 3 same blocks, and *Avg_pool* is the average pooling function.

3.3 Human Activities Segmentation using Encoder-Decoder Structure

As integral components of human activities, *human-object interactions* (HOIs) bear close connections to the surrounding environment and the objects within a scene. The recognition of HOIs in videos stands as a foundational task in comprehending human activities. This involves segmenting and recognizing sub-activities per frame by scrutinizing the interactive relationships between humans and objects [Mor+21]. When humans and objects are represented simply by skeletons and center points, these relationships inherently create a graph structure in both spatial and temporal dimensions, effectively capturing the relative positions and dynamic interactions during the activity. With the advancements in deep learning within the field of vision, constructing a spatial relation graph becomes more accessible through the detection of humans and objects in scenes. However, unveiling the temporal structure of sub-actions in a complex task remains a challenging endeavor.

Currently, existing graph convolutional networks, including the aforementioned

studies, predominantly concentrate on recognizing the overarching, prevalent action being executed. In these approaches, only a single action is performed in one set of clips. Such methods commonly leverage cascaded structures and effectively extract and concentrate spatio-temporal features. However, their scope is constrained to the task of assigning action labels to the provided segments [PD+20; XB22a]. The question arises: Can the extracted spatio-temporal information be harnessed to explore the temporal structure of activities, specifically for action segmentation?

In addressing this question, we draw parallels to the distinction between image classification and segmentation. Image classification typically employs a cascade structure, extracting global high-level features to classify the entire image [KSH12]. On the other hand, image segmentation concentrates on discerning pixel-level distinctions by upsampling the cascaded features back to the original scale [Wu+19a].

3.3.1 Pyramid Graph Convolutional Network

The concept behind the Pyramid Graph Convolutional Network (PGCN) is inspired by upsampling methods used in image semantic segmentation tasks. In both image segmentation and action segmentation, the shared objective is to predict each elemental unit of the input data. This involves extracting various levels of semantic features and subsequently mapping these features back to the input data to construct a segment map. The fundamental idea of PGCN is to downsample the large-scale data to distill valuable spatial information, typically with a smaller temporal scale. Subsequently, the distilled information is upsampled back to the same temporal scale as the input, a structure commonly referred to as an encoder-decoder.

Encoder

Given that the constructed graph lacks important human-object interaction information, we introduce an attention-based graph network. This network adaptively updates the initial adjacency matrix through the attention score map. The attention score is computed through the dot product between nodes, as follows:

$$M_{ij} = \frac{f_i \cdot f_j^T}{\sqrt{n}} \tag{3.13}$$

where M represents the attention mask map, f denotes the node feature vector, and i, j are the indices of nodes, and n serves as a normalization factor, typically selected to be the length of the feature vector.

In this study, we observe that incorporating mask maps into a 1-dimensional convolution layer enhances the relationship learning process. As depicted in Fig. 3.4, the input feature map undergoes parallel processing through two 2D convolution layers, generating two output maps of identical size. The dot product of these maps is then passed through a 1D convolution layer with a *sigmoid* activation function to extract the attention mask. The final attention map is created by combining the attention mask with the adjacency matrix, formulated as follows:

$$\mathbf{A}_{final,i} = \mathbf{M}_i + \hat{\mathbf{A}}_i = \mathbf{W}_i(\mathbf{F}_{1,i}^T \cdot \mathbf{F}_{2,i}) + \hat{\mathbf{A}}_i \tag{3.14}$$

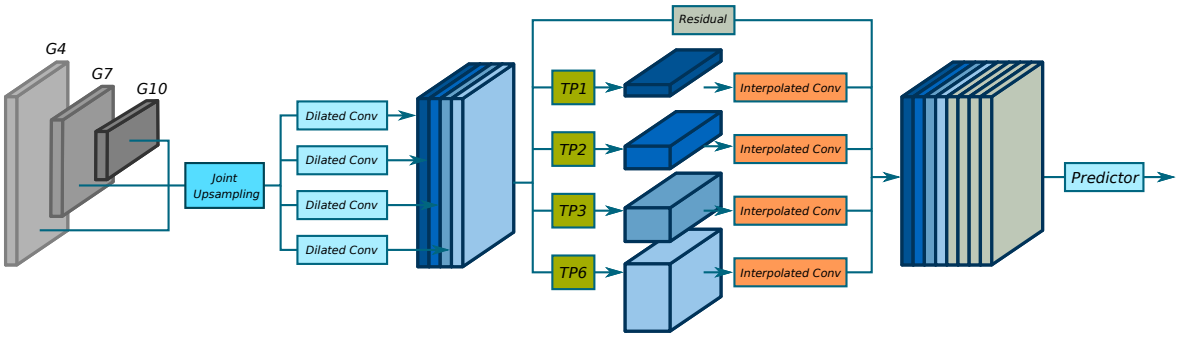


Figure 3.9: Framework of temporal pyramid pooling decoder with three input graph feature maps [XB22b]: $\mathbf{G}^4$, $\mathbf{G}^7$ and $\mathbf{G}^{10}$, where $TP\ i$ is temporal pooling block with output size i.

where $\mathbf{M}$ represents the attention mask extracted by the 1D convolution kernel applied to the dot product of feature maps $\mathbf{F}_1$ and $\mathbf{F}_2$. $\mathbf{W}$ denotes the kernel weight, $\mathbf{A}$ signifies the adjacency matrices, and i stands for the index of the three connection types (*inwards*, *outwards*, *self-connecting*). To introduce greater flexibility to the spatial graph, we designate adjacency matrices as learnable parameters with predefined initial values.

The output feature map of the spatial attention layer undergoes expansion to C_{out} output channels via an additional 2D convolution layer, and subsequently, it is integrated with the residual stream. This process can be mathematically expressed as follows:

$$\mathbf{G}_i = \text{Conv2d}(\mathbf{A}_{final,i} \cdot \mathbf{F}_{in}) + \text{res}(\mathbf{F}_{in}) \tag{3.15}$$

where $\mathbf{F}_{in}$ is the input feature map, res is the residual layer, and $\mathbf{G}$ is the $i-th$ output graph feature map. The final block output feature map is obtained by summing the outputs of all three types of connections as $\mathbf{G} = \sum_{i=1}^{3} \mathbf{G}_i$

In the temporal dimension, we adhere to the ST-GCN [YXL18] approach, employing a 2D convolutional kernel with a size of $K_t \times 1$ on $C \times T \times V$ feature maps, where K_t is set to 9 in this work.

With the defined spatial and temporal layers, an attention-based graph convolutional block is established. In the encoder, we employ 10 basic blocks connected through the standard cascade structure, as introduced in [YXL18; Shi+19b].

The encoder is constructed by concatenating the 10 aforementioned basic spatial-temporal graph convolutional blocks with different channel sizes.

Temporal Pyramid Pooling Decoder

Given the introduced encoder, three graph feature maps $\mathbf{G}_{in} = \mathbf{G}^4, \mathbf{G}^7, \mathbf{G}^{10}$ from the 4th, 7th, and 10th blocks are collectively fed into the temporal upsampling module, encompassing diverse levels of semantic information. Since these feature maps have varying sizes, we standardize the number of channels using a 2D convolutional kernel and interpolate all feature maps to the initial time scale, concatenating them along the channel dimension. Subsequently, segmentation feature extraction is carried out through four parallel dilated convolution operations [Wan+16], as outlined below:

$$\mathbf{G}_{out} = \sqcup_{i=1}^{4} \sigma(\mathbf{G}_{in,u}\mathbf{W}_i^d + \mathbf{B}_i^d) \tag{3.16}$$

where $\mathbf{G}_{out}$ is output graph feature map, $\mathbf{G}_{in,u}$ is upscaled input graph feature map, $\sqcup_{i=1}^{4}$ indicates the concatenation operation with 4 streams, σ is the *ReLU* activation function, $\mathbf{W}_i^d$ and $\mathbf{B}_i^d$ are parameters of i-th dilated convolutional kernel.

To capture a global contextual prior for prediction, a temporal pyramid pooling module is employed before the predictor. In semantic segmentation tasks for images, global average pooling is commonly used as the global contextual prior. However, in the context of action segmentation, features in temporal and spatial dimensions must be treated differently. As the final segmentation is in the temporal dimension, i.e., a sequence of predicted labels per frame, four pyramid temporal average pooling blocks of varying scales are initially applied along the temporal dimension to extract a segment prior with multiple receptive fields.

Given the time series dilated graph feature map $\mathbf{G}out \in \mathbb{R}^{N\times T}$ with N spatial nodes and T frames, it can be represented as a set of time segments at level i denoted as $\mathbf{G}out = \mathbf{G}1, \cdots, \mathbf{G}i$. A temporal filter with an average pooling operator is applied to each time segment $[tmin, tmax]$, yielding a single feature vector for each segment as follows:

$$\mathcal{O}(\mathbf{G}_i) = \frac{\sum_{t_{min}}^{t_{max}} g_t^i}{t_{max} - t_{min}} \tag{3.17}$$

Subsequently, a convolutional layer is applied in the spatial dimension to extract global spatial information across different temporal scales, as follows:

$$\mathbf{F}_{out} = \sigma(\mathbf{G}_{out}\mathbf{W}_s + \mathbf{B}_s) \tag{3.18}$$

where $\mathbf{W}_s \in \mathbb{R}^{k\times 1}$ and $\mathbf{B}_s \in \mathbb{R}^{k\times 1}$ are parameters of the spatial convolutional kernel, and $k \times 1$ indicates the kernel size.

The four low-dimension output feature maps are upsampled directly using bilinear interpolation to match the temporal and spatial lengths of the original feature maps. Finally, the feature maps from the four different levels are concatenated with the residual feature map. Once the feature map, containing global contextual priors with various scales and framewise local features, is obtained, a convolution-based predictor is employed to generate framewise interaction labels. The framework is depicted in Fig. 3.9.

3.3.2 Temporal Fusion Graph Convolutional Network

The efficacy of an action recognition system hinges on its ability to identify cues defining action labels and the spatial-temporal relations demarcating consecutive actions within a task. We represent activities using spatio-temporal graphs, wherein the joints of a human skeleton and the center points of bounding boxes encompassing objects serve as graph nodes, and graph edges delineate active relations between nodes. Sub-activities are recognized and segmented frame-wise by analyzing dynamic connections between graph nodes [XB22b]. An Attention-based Graph Convolutional Network (GCN) stands out as one of the most widely applied solutions for processing dynamic Human-Object Interaction (HOI) graph relations. It adaptively updates node correlations through an attention mechanism, iteratively parsing features in spatial and temporal dimensions [Kre+21; Mor+21]. When combined with a decoder, the processed graph features are further upsampled to the original time scale,

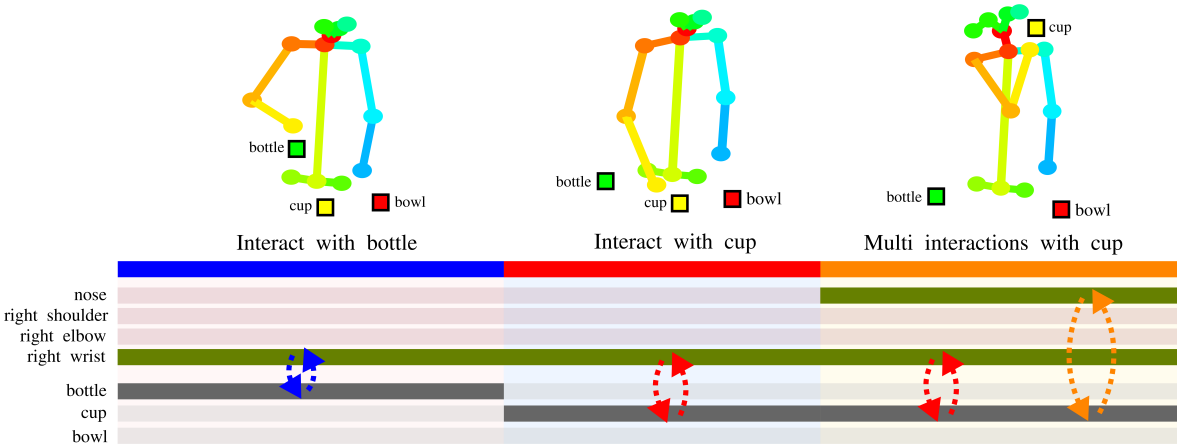


Figure 3.10: Dynamic relations (depicted by dashed arrows) exist between the body parts and objects involved in the action of *drinking*, with the human represented by a skeleton and objects represented by square boxes. The highlighted time bar signifies active participation in the interaction by either objects or skeleton joints.

and then the graph sequences are classified and segmented frame by frame [XB22b]. Nevertheless, overcoming segmentation inaccuracies and over-segmentation in the temporal dimension remains a challenging hurdle for researchers.

To enhance HOI recognition and segmentation performance, we introduce a novel Temporal Fusion Graph Convolutional Network (TFGCN) comprising an attention-based graph convolutional encoder and a newly designed Temporal Fusion (TF) decoder. The innovative decoder extracts global features through multiple parallel temporal-pyramid-pooling blocks and augments temporal features by fusing high-dimensional features from the encoder with processed low-dimensional features. Experimental results on public datasets demonstrate superior performance in terms of recognition accuracy and mitigation of boundary shifts and over-segmentation.

Similar to the PGCN, the fundamental concept behind the Temporal Fusion Graph Convolutional Network is inspired by image segmentation approaches that predict the semantic meaning of each pixel unit by extracting global spatial features and mapping them to the corresponding spatial positions. However, in our case, we feed the graph representations of Human-Object Interactions (HOI) into the network instead of images, as graph representations are insensitive to background and appearance noise. Our Temporal Fusion Graph Convolutional Network processes graph features not only in the spatial dimension but also in the temporal dimension, as illustrated in Fig 3.10. Dynamic relationships are evident in the process of *drinking*. The pertinent targets, including objects and skeleton joints, are highlighted along the time axis to depict sub-actions. Specifically, the *right wrist* node sequentially interacts with the *bottle* and *cup* node. In the final phase of the action, the *cup* node concurrently engages with both the *right wrist* and *nose*. These features are compressed into a low temporal dimensional space by the encoder and then upsampled to the original temporal dimension by the decoder.

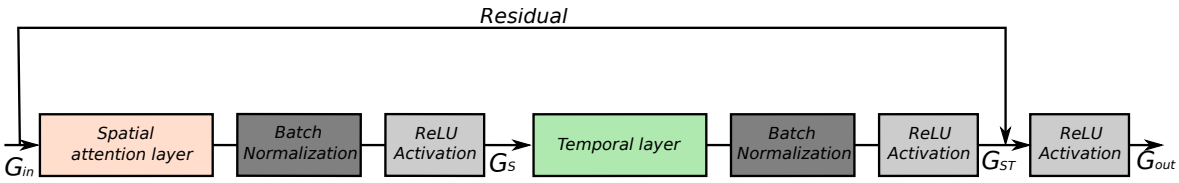


Figure 3.11: Basic block unit consisting of spatial convolutional layer, Batch Normalization, temporal layer and ReLU activation function with a residual side branch.

Encoder

As shown in the Fig 3.4, the central principle of the attention mechanism involves updating the predefined adjacency matrix by incorporating global correlations —the attention map, which can typically be calculated as shown in Eq 3.2.

Given an input graph $\boldsymbol{G}_{in} \in \mathbb{R}^{C_{in} \times T \times V}$, the spatial graph feature map $\boldsymbol{G}_s \in \mathbb{R}^{C_{out} \times T \times V}$ can be obtained by the following equation:

$$\begin{gathered} \boldsymbol{G}_s = Conv2d(\boldsymbol{G}_{in}) \cdot (a\boldsymbol{A}_{att} + \boldsymbol{A}), \\ \boldsymbol{A}_{att}, \boldsymbol{A} \in \mathbb{R}^{V \times V} \end{gathered} \tag{3.19}$$

where C_{in}, C_{out}, T and V are input channel number, output channel number, temporal size and spatial size, respectively. Note that the kennel size of the convolutional kernel in spatial layer is 1×1, since the spatial features are processed by the attention map and adjacent matrix.

The spatial graph feature undergoes further processing through a temporal graph convolutional layer to yield the spatio-temporal processed feature map $\boldsymbol{G}st$, as illustrated in Fig 3.11. An example of a simple temporal layer is the one used in ST-GCN [YXL18], which employs a single 2D convolutional kernel with a kernel size of 9 in the temporal dimension. The final output graph feature map $\boldsymbol{G}out$ is then obtained by merging the spatial-temporal processed feature map $\boldsymbol{G}_{st}$ with a residual connection, formulated as follows:

$$\boldsymbol{G}_{out} = res(\boldsymbol{G}_{in}) + \boldsymbol{G}_{st} \tag{3.20}$$

For optimal performance, we evaluate the effectiveness of various popular attention-based graph convolutional networks as encoders. ST-GCN [YXL18] is implemented as a baseline without an attention mechanism. AGCN [Shi+19b] is a variant of ST-GCN that integrates a product attention mechanism into the spatial graph convolutional layer. In the case of PGCN [XB22b], the encoder passes the attention map through an additional 1D convolutional layer to adjust its weights. CTR-GCN [Che+21] refines the spatial attention mechanism in the channel dimension to learn different dynamic features in each channel. HA-GCN [XB22a] proposes a hybrid attention mechanism that combines product and subtract attention maps to enrich the dynamic features of different input streams. In the temporal dimension, ST-GCN, AGCN, and PGCN process features using a single 2D convolutional kernel, while HA-GCN and CTR-GCN employ multi-scale temporal convolutional kernels as introduced in the work [Liu+20b].

The encoder is formed by concatenating 10 aforementioned basic spatial-temporal graph convolutional blocks with different channel size.

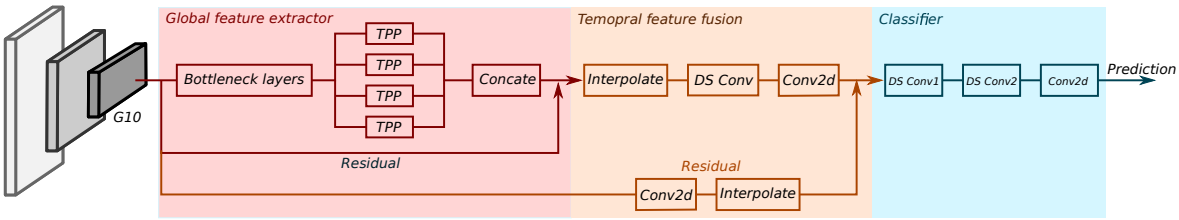


Figure 3.12: Framework of temporal fusion decoder including three blocks: Temporal feature extractor, feature fusion and classifier. The "Concate" block concatenates all feature maps from *temporal pyramid pooling* (TPP) layers into one. "DS Conv" represents depth-wise 2D convolutional layer.

Temporal Fusion Decoder

To upsample the condensed features back to the original time scale and predict action labels for each frame, we propose a novel temporal feature fusion decoder. As shown in Fig. 3.12, the decoder feature map is subscribed separately by two blocks, namely temporal feature extractor and feature fusion. The temporal feature extractor further compresses and extracts temporal features through three serial linear bottleneck layers [San+18] and four parallel temporal pyramid average pooling [XB22b] blocks. The outputs from the four averaged features are concatenated, processed through a 2D convolutional kernel, and merged with a temporal residual connection. In the feature fusion block, the condensed feature is upsampled to the original time scale using an interpolate module. Depth-wise separable convolutional (*DS Conv*) [Cho17] and 2D convolutional layers are applied to process the interpolated features. In the residual branch, the encoded condensed features are initially processed by a 2D convolutional kernel and interpolated to the same size as the main branch. The residual connection integrates original high-dimensional features into the mainstream, fusing them with the low-dimensional features, contributing to enhanced performance accuracy.

Furthermore, owing to its excellent compatibility, we substitute the proposed decoder with two existing upsampling methods, i.e., *Fast-FCN* [Wu+19a] and *Temporal-Pyramid-Pooling* (TPP) [XB22b]. Originally designed to address image segmentation tasks, *Fast-FCN* demonstrated promising results by jointly upsampling three processed feature maps from different depths of the encoder. To adapt the model for action segmentation tasks, we modify the joint upsampling module to upsample solely along the time axis. TPP is another recent decoder that incorporates four parallel *temporal pyramid pooling* modules after a joint upsampling block. Leveraging dilated convolutional kernels with different scales, TPP exhibits a broad range of receptive fields and delivers commendable performance in action segmentation.

In the classifier, the fused features are first compressed in the spatial dimension by two depth-wise separable kernels, and subsequently, they are mapped to the class space in the channel dimension.

Experimental analyses of different encoder-decoder setups on human action segmentation datasets is introduced in Section 4.4.

3.4 Event Detection using Sparse Coding and Dictionary Learning

Another crucial aspect of human-robot interaction involves recognizing trigger events that necessitate a response on the robot's part [LP07; Tur+08; Fan+09]. Typically, such events involve unexpected actions or motions on the part of the human subject. These events might prompt additional learning of new motions or trigger an emergency response in the case of accidents. In this study, we specifically focus on the common event detection of humans falling down due to tripping or health conditions.

Given that the majority of the human body can be considered an articulated system with rigid bones connected by joints, human actions can be expressed as the movement of the skeleton [Lie+19]. Existing skeleton-based event detection methods typically fall into two main categories: 2D skeleton-based approaches [LLL18; Avo+19; Zhe+19] and 3D skeleton-based approaches [Min+18; Wu+19b; Zha+16]. In comparison to 2D skeleton-based methods, 3D skeletons provide more extensive spatial information at the expense of increased time consumption and manual labeling requirements. Many existing research methods still face the challenge of an ill-posed and inverse problem when attempting to extract 3D skeletons from monocular images [Zhe+20].

With the introduction of Microsoft Kinect [Pöh+16] and RealSense [Kes+17] cameras has made multidimensional observation of human events feasible without imposing high processing loads on the system. However, the noise inherent in depth measurements from these cameras significantly impacts event detection. To address this issue, we employed a gradual filtering process on skeleton sequences extracted from RGB images using a lightweight Deep Learning toolbox with aligned depth information.

In addition to detecting the event action, learning and establishing structure representation of the action is also essential and challenging. Different actions may share the same start and end positions and exhibit similar pose transformations and rotations, such as lying down and falling down. However, their latent temporal features are distinct. Modeling the latent spatio-temporal structures of actions is one of the most widely-used techniques for action recognition and representation [Rab89; WM10; TFK12]. A latent spatio-temporal structure consists of two parts: the action unit with spatial information and the temporal model. The action units are the sequences and constituent elements of the action. The temporal feature defines the length of the step from the previous state to the next state [Qi+18]. For the fall-down event, the temporal feature is the sharp height change of the skeleton [Ma+14].

For extracting the latent action unit, Sparse Coding Dictionary (SCD) is a well-known approach [Chi+13; BDB18; Mai+10]. This method approximates a given video sequence $\mathbf{Y}$ through the manipulation of a low-rank dictionary $\mathbf{D}$ and its coefficient matrix $\mathbf{X}$. Online Dictionary Learning (ODL) stands out as one of the most successful SCD methods and is widely employed in the field of action recognition. Given that fall event detection is just one extreme case of action recognition, we consider the ODL algorithm in this work as a baseline method. Its cost can be expressed

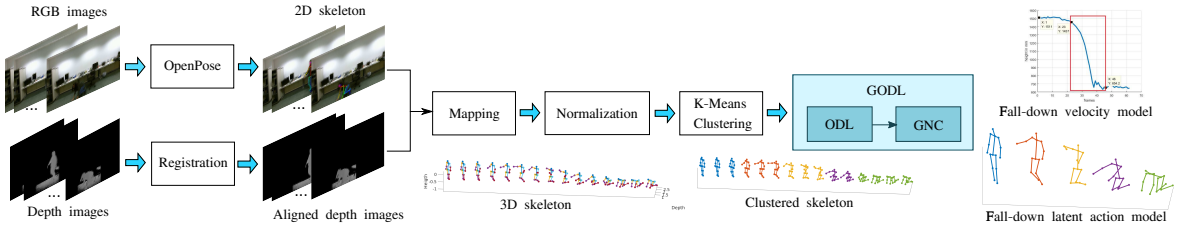


Figure 3.13: Overview of human fall event detection training process using the gradual online dictionary learning (GODL) [Xin+21].

in the least squares problem with a regularizer as:

$$\min_{\forall \mathbf{D}_i \in \mathcal{D}, \forall \mathbf{X}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{1}{2} \| \mathbf{Y}_i - \mathbf{D}_i \mathbf{X}_i \|_F^2 + \lambda \| \mathbf{X}_i \| \tag{3.21}$$

where F means Frobenius norm, N is the number of action unit and λ is the regularization parameter. Unfortunately, in the presence of outliers, Eq (3.21) yields a poor estimation for $\mathbf{D}$ and $\mathbf{X}$ [Yan+20]. This issue is exacerbated in the context of 3D skeleton-based human fall event detection due to the increased prevalence of outlier sources, such as skeleton estimation and depth measurement.

In this work, an attempt to improve event detection latency and temporal resolution is presented and performed at the example of fall detection. We separate the fall event into five latent action atoms "*standing*", "*bending knee*", "*opening arm*", "*Knee landing*" and "*arm supporting*".

3.4.1 Task Definition

Formally, let $\mathbf{Y} = \{\bar{y}_1, \ldots, \bar{y}_t\}$ denote a fall-down 3D pose sequence and $\bar{y}_j$ is the j-th column vector of skeleton joints. We assume that the sequence $\mathbf{Y}$ is segmented into N sub-sequences $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$ and each sub-sequence corresponds to an action unit $\mathbf{D}_i = \{\bar{d}_1, \ldots, \bar{d}_k\}$. Then the dictionary can be expressed as $\mathbf{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_N\}$ and their coefficient matrix is defined as $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$.

3.4.2 Prepossessing of Data

An overview of the fall event detection training process is shown in Fig 3.13. RGB images are fed into a pose estimator to get 2D skeleton joints. Concurrently, depth frames are aligned with RGB images, and 3D skeleton joints are derived by projecting pixel positions to 3D space along with the aligned depth values. To account for variations in the initial position from which a person may fall in image coordinates, a normalization function is applied to maintain consistent skeleton magnitudes and ratios for each direction: $x \in [0,1]$, $y \in [0, y_{max}/(x_{max} - x_{min})]$, $z \in [0, z_{max}/(x_{max} - x_{min})]$. To balance the impact of spatial and temporal information, we introduce a weight parameter $w_{s/t}$ with a value of 0.1. It is defined as $w_{s/t} = p/v$ in the paper. A K-means-based clustering method is employed to segment a sequence into N clusters.

Algorithm 1: Gradual Online Dictionary Learning

Input : Fall-down 3D skeleton sequence $\mathbf{Y}$
Output: Dictionary matrix $\mathbf{D}$ and coefficient matrix $\mathbf{X}$

while $i < N$ **do**
  Initialization: $\bar{w}_i^{T(0)} = \bar{\mathbf{1}}^T$, $\mu_0 = 2 * e_{i,max}^2 / c^2, \mathbf{D}_i^{(0)}, \mathbf{X}_i^{(0)}$;
  while $\mu \geq 1$ **do**
    Filter outlier from $\mathbf{Y}_i$: ($\cdot$ is column dot-production)
    $\hat{\mathbf{Y}}_\mathbf{i} = \bar{w}_i^T \cdot \mathbf{Y}_i$ and $\bar{\lambda}_w = \lambda \bar{w}_i^T$;
    repeat
      Update $\hat{\mathbf{X}}_i^{(k)}$ with fixed $\mathbf{D}_i^{(k-1)}$;
      Update $\mathbf{D}_i^{(k)}$ with fixed $\hat{\mathbf{X}}_i^{(k)}$;
    until *end of ODL iteration or reach convergence*;
    Update weight vector:
    for $j \leftarrow t_i^1$ **to** t_i^{end} **do**
      $w_{i,j}^{(k)} = \arg\min_{w_i \in [0,1]} \mathcal{O}_{i,j} + \Phi_{g_\mu}$
    end
    Update $\mu = \mu / 1.4$;
  end
end

3.4.3 Train Phase

For each sub-sequence, we employ gradual online dictionary learning (GODL) to iteratively update the coefficient matrix $\mathbf{X}_i$ and its action unit matrix $\mathbf{D}_i$ until the cost converges or the maximum iteration number is reached. The general framework for GODL is outlined in Algorithm 16. The primary concept is to facilitate the iteration process to automatically filter outliers, ensuring that the latent action units are learned from inliers.

Graduated non-convexity is a widely used method for optimizing general non-convex cost functions, such as the Geman McClure (GM) function. The GM function is expressed as follows:

$$g_\mu(e) \doteq \frac{\mu c^2 e^2}{\mu c^2 + e^2} \tag{3.22}$$

where c^2 is a given constant that is the maximum accepted error of inliers, μ determines the shape of GM function and e^2 is Frobenius norm of error between training sequence $\bar{y}_{i,j}$ and approximation model $\mathbf{D}_i \bar{x}_{i,j}^T$ as follow:

$$\begin{aligned} e_{i,j}^2 = &\| \bar{y}_{i,j} - \mathbf{D}_i \bar{x}_{i,j}^T \|_F^2 + \lambda \| \bar{x}_{i,j}^T \| \\ &\text{with } i \in [1, N],\ j \in [t_i^1, t_i^{\text{end}}] \end{aligned} \tag{3.23}$$

At each outer iteration, a new μ is updated, and we optimize Eq (3.24). The solution obtained at each iteration serves as an initial guess for the next iteration. The final solution is computed until the original non-convexity function is restored

$(\mu = 1)$.

$$\min_{\forall \mathbf{D}_i \in \mathcal{D}, \forall \bar{x}_{i,j} \in \mathcal{X}} \sum_{j=t_i^1}^{t_i^{\text{end}}} g_\mu \left(e(\bar{y}_{i,j}, \mathbf{D}_i \bar{x}_{i,j}^T) \right) \tag{3.24}$$

We use the Black-Rangarajan duality to combine the GNC-GM function with weighted ODL cost as follow:

$$\min_{\forall \mathbf{D}_i \in \mathcal{D}, \forall \bar{x}_{i,j} \in \mathcal{X}} \sum_{j=t_i^1}^{t_i^{\text{end}}} \mathcal{O}_{i,j}(w_{i,j}, \mathbf{D}_i, \bar{x}_{i,j}^T) + \Phi_{g_\mu}(w_{i,j}) \tag{3.25}$$

with weighted cost:

$$\begin{aligned} \mathcal{O}_{i,j} =& w_{i,j}^2 \left(\frac{1}{2} \parallel \bar{y}_{i,j} - \mathbf{D}_i \bar{x}_{i,j}^T \parallel_F^2 + \lambda \parallel \bar{x}_{i,j}^T \parallel \right) \\ =& \frac{1}{2} \parallel w_{i,j} \bar{y}_{i,j} - \mathbf{D}_i (w_{i,j} \bar{x}_{i,j}^T) \parallel_F^2 + \\ & \lambda w_{i,j} \parallel w_{i,j} \bar{x}_{i,j}^T \parallel \end{aligned} \tag{3.26}$$

and penalty term:

$$\Phi_{g_\mu} = \mu_i c^2 (w_{i,j} - 1)^2 \tag{3.27}$$

With simplified expression of $\bar{\hat{x}}^T = w\bar{x}^T$, $\bar{\hat{y}}^T = w\bar{x}^T$ and $\lambda_w = w\lambda$, the Eq (3.26) can be described as following:

$$\mathcal{O}_{i,j} = \frac{1}{2} \parallel \bar{\hat{y}}_{i,j} - \mathbf{D}_i \bar{\hat{x}}_{i,j}^T \parallel_F^2 + \lambda_{w\,i,j} \parallel \bar{\hat{x}}_{i,j}^T \parallel \tag{3.28}$$

During the first inner iteration, all weights are set to 1. Throughout the inner iterations, the weighted ODL is optimized with a fixed weight $(w_{i,j})$, and then we optimize over $w_{i,j}$ with a fixed cost of ODL. At a specific inner iteration k within the weighted sub-sequence $\hat{\mathbf{Y}}_i$, we follow these steps:

1) **Dictionary Learning**: minimize the Eq (3.25) with respect to $\mathbf{D}_i^{(k)}$ and $\bar{x}_{i,j}^{(k)}$ with fixed $w_{i,j}^{(k-1)}$. This problem is the original ODL, but with weighted training sequence:

$$\min_{\forall \mathbf{D}_i \in \mathcal{D}, \forall \mathbf{X}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{1}{2} \parallel \hat{\mathbf{Y}}_i - \mathbf{D}_i \hat{\mathbf{X}}_i \parallel_F^2 + \lambda_{w\,i} \parallel \hat{\mathbf{X}}_i \parallel \tag{3.29}$$

In the ODL optimization, we first update the coefficient matrix $\mathbf{X}i^{(k)}$ with the fixed action unit $\mathbf{D}i^{(k-1)}$ (Sparse Coding). We assign the weight parameter to the training sequence $\mathbf{Y}i$ and the coefficient matrix $\mathbf{X}i^{(k)}$. Then, we update the action unit $\mathbf{D}i^{(k)}$ with the fixed weighted coefficient matrix $\hat{\mathbf{X}}i^{(k)}$ and the weighted input matrix $\hat{\mathbf{Y}}_i$ (Dictionary Learning):

- Assign Weight: $\hat{\mathbf{Y}}_i = \bar{w}_i^T \cdot \mathbf{Y}_i$ and $\bar{\lambda}_w = \bar{w}_i^T \lambda$, where $\cdot$ is column dot-production.
- Sparse Coding: we use Lasso-Fista algorithm to update $\hat{\mathbf{X}}_i^{(k)}$ with fixed $\mathbf{D}_i^{(k-1)}$, see [Mai+10].

- Dictionary Learning: minimize the following equation with fixed $\hat{\mathbf{X}}_i^{(k)}$:

$$\begin{aligned}\mathbf{D}_i^{(k)} = &\arg\min_{\forall \mathbf{D}_i \in \mathcal{D}} -2\mathrm{tr}(\mathbf{E}_i^T \mathbf{D}_i^{(k-1)}) \\ &+ \mathrm{tr}(\mathbf{D}_i^{(k-1)} \mathbf{F}_i \mathbf{D}_i^{T(k-1)}) \\ &\text{with } \mathbf{E}_i = \hat{\mathbf{Y}}_i \hat{\mathbf{X}}_i^{T(k)} \\ &\text{and } \mathbf{F}_i = \hat{\mathbf{X}}_i^{(k)} \hat{\mathbf{X}}_i^{T(k)}\end{aligned} \tag{3.30}$$

2) **Weight update**: minimize the Eq (3.25) with respect to weight $w_{i,j}^{(k)}$ with fixed dictionary matrix $\mathbf{D}_i^{(k)}$ and coefficient vector $x_{i,j}^{(k)}$.

$$\begin{aligned}\bar{w}_i^{T(k)} = &\arg\min_{w_{i,j} \in [0,1]} \sum_{j=t_i^1}^{t_i^{\text{end}}} \{\mathcal{O}_{i,j}(w_{i,j}^{(k-1)}, \mathbf{D}_i^{(k)}, \bar{x}_{i,j}^{T(k)}) \\ &+ \Phi_{g_\mu}(w_{i,j}^{(k-1)})\}\end{aligned} \tag{3.31}$$

Using introduced ODL function Eq (3.26) and penalty function Eq (3.27), the weight update at iteration k can be solved in form as:

$$w_{i,j}^{2(k)} = \left(\frac{\mu_k c^2}{\mu_k c^2 + e_{i,j}^2} \right)^2 \tag{3.32}$$

where $e_{i,j}^2$ is Frobenius norm of error between training sequence $\bar{y}_{i,j}$ and approximation model $\mathbf{D}_i \bar{x}_{i,j}$, see Eq (3.23).

In the implementation, we start with an initialization $\mu_0 = 2 * e_{i,\text{end}}^2 / c^2$ with

$$e_{i,\text{end}}^2 \doteq \max_{\forall \mathbf{e}_{i,j} \in \mathcal{E}_i, j \in [t_i^0, t_i^{\text{end}}]} e_{i,j}^2 \tag{3.33}$$

At each outer iteration, update $\mu_k = \mu_{k-1}/1.4$ and stop when μ_k is blow 1, see [Yan+20].

3.4.4 Inference Phase

In the inference phase, we assume that the error between sub-sequence $\mathbf{Y}_i$ and the model $\mathbf{D}_i \mathbf{X}_i$ is normally distributed. Hence, the measured error e_i between real-time skeleton frames of a fall-down action and the action unit model should fall within the confidence interval as follows:

$$\frac{|e_i - e_{i,\text{mean}}|}{\sigma(e_i)} < \alpha \tag{3.34}$$

where $e_{i,\text{mean}}$ is the mean error of training set, $\sigma(e_i)$ is the standard deviation of error e_i ,and α is an acceptance parameter.

Since the fall event has a strict order of sub-actions, progressing from "standing" to "on the ground," each sub-action detection will be performed only when the previous action is completed.

In addition to action unit extraction, the temporal feature of falling down is crucial as well. A fall is defined as an event that results in a person moving from a higher to a lower level, typically rapidly and without control. From this definition, we can infer that the action "fall down" involves a rapid change in a person's height over a very short time. For the height change, we don't need all the skeleton information; only the skeleton information in the y-direction is necessary, as shown in the following equation:

$$h = y^T_{max} - y^T_{min} \tag{3.35}$$

where y is the value of skeleton in y axis, h means the height of skeleton and T is the width of time interval shifted from beginning of video to end. Since the first action unit is "*standing*", we define its height as an initial value h_{init}. The height change of fall event inside a time interval should meet following two conditions:

$$\begin{cases} \frac{h^0}{h_{init}} > 0.9, & h^0 \text{ is begin of interval.} \\ \frac{h^{T-1}}{h^0} < 0.5, & h^{T-1} \text{ is end of interval.} \end{cases} \tag{3.36}$$

where these thresholds are obtained through experiments.

Experimental analyses of fall event detection using the proposed Gradual Online Dictionary learning method is presented in Section 4.5.

3.5 Understanding Human Activity with Uncertainty Measure for Novelty

Understanding Human-Object Interactions plays a crucial role in intelligent systems, particularly for robots learning from demonstrations and collaborating with humans. This entails not only the recognition and segmentation of interaction relations per frame but also the quantification of prediction uncertainty.

The proposed PGCN and TFGCN have substantially enhanced the performance of action recognition and segmentation. Nonetheless, learning-based models often exhibit overconfidence in incorrect predictions, whereas real-world scenarios involve numerous unforeseen situations, including noise and unknown data. These factors heighten the risk and complexity of application. Consequently, the detection of novel human actions becomes imperative for the implementation of our model.

Multi-object tracking algorithms provide inspiration for addressing the problem, often assigning IDs based on the distance between representation features and the existing feature space [WB18]. In essence, this necessitates the model to be distance-aware in the representation space [Liu+20a], as articulated below:

$$\begin{aligned} \alpha \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel_X < \parallel g(\boldsymbol{x}) - g(\boldsymbol{x}') \parallel_G \\ < \beta \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel_X \end{aligned} \tag{3.37}$$

where g means the graph convolutional layer and maps the input data from manifold X (input space) to the representation space G (feature space), $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two different inputs. The parameters α and β are the lower and upper bounds with a constraint of $0 < \alpha < \beta$. In this *bi-Lipschitz* condition, the upper bound affects the

sensitivity of hidden representations to the novel observations (out-of-distribution, OOD) and the lower bound guarantees the distance in hidden representation space for meaningful changes in the input manifold [Liu+20a].

Traditional cascaded convolutional networks establish an upper bound for the hidden representation space distance through normalization and activation functions [RHK18]. However, they encounter challenges related to exploding and vanishing gradients.

Residual connections demonstrate the capability to mitigate gradient-related issues [VWB16], but they can result in a broader range and less distinguishable features in the representation space for out-of-distribution (OOD) detection. To maintain a meaningful isometric property in our deterministic model, we introduce a Spectral Normalized Residual (SN-Res) connection, imposing an upper *Lipschitz* constraint on the residual flow. We construct an Uncertainty Quantified Temporal Fusion Graph Convolution Network (UQ-TFGCN) using this innovative approach, wherein the hidden representation space is confined to a reasonable region. Consequently, the final label and similarity of unknown data are predicted through maximum likelihood in a Gaussian Process (GP) kernel.

3.5.1 Research Background of Uncertainty Quantification

Deep neural networks are designed to mimic the way how human brain works, by processing complex information through multiple layers of interconnected nodes. Each node in a neural network (NN) performs simple mathematical operations on its inputs and then passes the results to the nodes in the next layer. Macroscopically, the whole network can be considered as a model $f_\theta(x)$ controlled by θ, including the weights and bias parameters of all network nodes. During training, $f_\theta(x)$ is feed by a dataset $D = (X, Y) = (x_n, y_n)_{n=1}^N$, in which x_n is the input n-th data and Y is the corresponding output n-th label. The network is then supervised to map data from an input dataset to a given output dataset, i.e., find the optimal θ^* for $f_{\theta*}(X) = \bar{Y}$, such that output result $\bar{Y}$ converges to given labels Y. After training on the whole dataset D, during the inference process (also stated as the prediction process), the trained network gives $\tilde{y} = f_{\theta*}(\tilde{x})$, in which $\tilde{x}$ is a new input data sample and $\tilde{y}$ is the corresponding prediction.

During the training and prediction process, the factors affecting the prediction results appear in two main areas: the model and the data. Uncertainty in most research is thus distinguished into model uncertainty and data uncertainty.

Model Uncertainty

Model uncertainty, also known as epistemic uncertainty [HW21; KG17] or knowledge uncertainty [MG18], signifies that the model's own estimation during the fitting process of input data may be inaccurate, or the model used to represent the fitting process is constrained. This limitation can arise from factors such as incomplete or insufficient training data, deficiencies in the training process, perturbations during training, etc., and is independent of the specific input data provided. The factors influencing model formation can be categorized based on the learning process. During the training process, insufficient training data, network structure, and network pa-

rameters can impact model uncertainty by influencing the training outcomes, specifically the final model's performance. In the inference process, out-of-distribution (OOD) data can affect model uncertainty by yielding suboptimal predictions from the pre-trained model.

Effect of Network Structure The term "network structure" encompasses various elements in the design and construction of a model, including the model architecture, layer specifications, activation functions, loss functions, and optimization techniques. These elements collectively define the complexity and expressive capacity of the model, influencing its ability to capture the underlying patterns present in the data. Key components of network structure also involve considerations such as regularization methods, dropout, early stopping, and other architectural features that contribute to the overall design and functionality of the neural network.

If the network structure is overly simplistic or lacks the required complexity, it may result in underfitting the data, leading to increased epistemic uncertainty. In this scenario, the model struggles to capture the underlying patterns within the data, resulting in inaccurate predictions. Conversely, an excessively complex network structure may lead to overfitting, also contributing to elevated epistemic uncertainty. In such cases, the model tends to memorize noise in the data rather than discerning the general underlying patterns, hampering its ability to generalize effectively to new, unseen data [LPB17; Guo+17].

Moreover, specialized network structures are designed to cater to specific scenarios. For instance, recurrent networks are often employed in natural language processing, while convolutional networks are well-suited for tasks like image recognition and segmentation.

Effect of Network Parameters Network parameters, also known as hyper-parameters, encompass the settings adjusted during the training process, influencing aspects such as model initialization and optimization. The uncertainty of a machine learning model is subject to the influence of various network parameters, covering those associated with the training process (e.g., batch size, learning rate, regularization strength) and parameters tied to randomness (e.g., initialization, optimization parameters).

The learning rate dictates the speed at which the model adjusts its weights during training. A higher learning rate can expedite convergence but might introduce instability and elevate uncertainty. The batch size, representing the number of samples used in each training iteration, influences the stability of the learning process. A larger batch size can enhance stability but may reduce sensitivity to individual samples, potentially increasing uncertainty.

Regularization techniques, such as L1 and L2 regularization, dropout, and early stopping, are employed to counteract overfitting in deep neural networks (DNNs). Parameters like the regularization coefficient and dropout rate control the strength of regularization. A higher level of regularization can amplify model uncertainty by introducing more variability into predictions. For instance, a larger dropout rate leads to varied updates for sets of nodes in each iteration, making it challenging to rely on any specific set of weights.

DNNs guide the network towards a local optimum by descending along the gradient of the loss function, but the network often possesses more than one local minimum. The stochastic optimization strategy employed by DNNs, along with the choice of the loss function and initial values, can all influence the model. It is unlikely that the network will converge to the same optimal solution, given the stochastic nature of network training, involving random initialization, optimization, and regularization. Networks arriving at the same optimal solution may exhibit different combinations of parameters.

Effect of Insufficient Data Insufficient data, particularly in the case of unbalanced training data, introduces challenges that can impact the effectiveness of the trained model. It refers to a situation where the number of samples in each class or category is not evenly distributed. This can have several effects on the model trained on such data: Firstly, Models trained on unbalanced data often exhibit a bias toward the majority class. This means that the algorithm tends to perform better on predicting the majority class, as it has more examples to learn from compared to the minority class. Secondly, if the test dataset has a different class distribution than the training dataset, for example, the test dataset is balanced, a model cannot generalize well to this new distributed testing dataset. Thirdly because of the lack of minority class data, the model will learn to fit noise in the data instead of a general pattern, causing overfitting on this class. Gawlikowski et al. state another insufficient data problem as distribution shift [Gaw+23]: During data acquisition, training data collected should cover all real-world situations, so that the model can represent human performance to overall real-world circumstances. But the real world environment is constantly changing. It is called a distribution shift when the real world situation changes compared to the training set. Neural networks are sensitive to distribution shifts [Gaw+23].

Effect of Out-of-Distribution Data Out-of-distribution (OOD) data refers to data that differs from the training data used to train the model. This can include data from different sources, domains, or distributions. The impact of OOD data on uncertainty can vary depending on the model's nature and the OOD data.

In general, OOD data can elevate the uncertainty of a model's predictions. This is because the model has not encountered this type of data during training and may struggle to handle it correctly. Consequently, the model may make inaccurate predictions or attribute high uncertainty to these predictions. For instance, consider a researcher building a model to predict which animals pose a threat to life based on a series of animal pictures. If the model is trained on pictures of lions and cats but encounters a zombie during the prediction process, its uncertainty regarding zombies would be very high, as it has not been exposed to them before. Introducing enough zombie photos during training can reduce the model's uncertainty accordingly.

In some cases, the increase in uncertainty due to OOD data can be advantageous. For example, in safety-critical applications like autonomous driving, it is crucial for the model to detect situations with high uncertainty and transfer control to a human operator.

Data Uncertainty

Data uncertainty, also referred to as aleatoric uncertainty [HW21; KG17], in DNNs pertains to the uncertainty associated with the data used for training and testing the network. This uncertainty may arise due to various factors, including noisy or incomplete data, sampling bias, or measurement errors. Importantly, this type of uncertainty cannot be diminished even with the collection of more data. For instance, when a camera experiences slight shakes, resulting in blurred images, increasing the number of photos cannot eliminate this data noise. Therefore, the typical approach to addressing this issue involves enhancing the stability of the data collection process. Kendall et al. [KG17] further categorize aleatoric uncertainty into homoscedastic uncertainty, which remains constant for different inputs, and heteroscedastic uncertainty, where some inputs may yield more noisy outputs than others. An example of heteroscedastic uncertainty occurs in depth estimation, where moving figures might have higher confidence compared to flat walls.

Another scenario of heteroscedastic uncertainty involves ambiguous training samples, which still originates from an in-distribution (ID) training dataset and, therefore, cannot be classified as out-of-distribution (OOD) epistemic uncertainty [CZG20]. For instance, in the MNIST handwritten single digits dataset, there are ambiguous samples that cause confusion during the training process. These ambiguous samples exhibit features that are closer to 4, resulting in elevated uncertainty in the training outcomes for samples falling between the boundaries of 4 and 6.

3.5.2 Uncertainty Quantification by Ensemble and Dropout Methods

The ensemble method typically involves training multiple networks with different initializations or network randomness on the same dataset. Each individual network is trained to minimize the prediction error on the training data. Once the individual networks are trained, their outputs are combined using a simple averaging or voting mechanism to make the final prediction on the test data.

Similarly, MC-dropout incorporates multiple results from the same trained model during inference. To ensure prediction accuracy, a small dropout rate, e.g., 5%, is applied. In this work, we implement both methods as baselines. For dropout, we apply output-layer dropout only during inference. Normally, to prevent overfitting, researchers apply dropout during the training phase. However, our model is not overfitting. In contrast, adding dropout during the training phase harms the model's performance, leading to underfitting. For models trained with the same structure, the final prediction is obtained by averaging over all outputs, as follows:

$$F_{out}(X) = \frac{1}{N}\sum_{i}^{N} f_i(X, \theta_i) \tag{3.38}$$

where $f_i(X, \theta_i)$ is the predicted output from i-th network with parameter θ_i . N is the number of neural networks in the ensemble and F (X) is the final prediction of the ensemble method. Then Uncertainty score function is applied to the aligned results like the other methods.

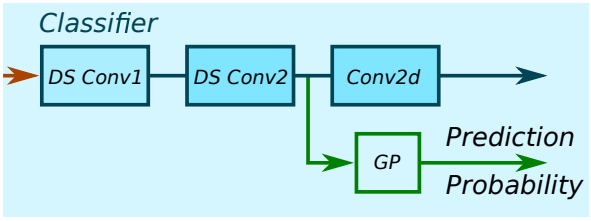


Figure 3.14: The Gaussian Process (GP) kernel collects high dimensional features from network before predictor and gives predictions with probabilities.

3.5.3 Distance-aware Feature Space by Spectral Normalized Residual Connection

In this section we introduce Uncertainty Quantified Temporal Fusion Graph Convolutional Network (UQ-TFGCN) with Spectral Normalized Residual connection, which balances the distance-preserving ability in representation space and high-accuracy performance. The baseline is the introduced temporal fusion graph convolutional network in Sec. 3.3.2.

In the development of activity segmentation networks, we observed that residual connections enhance prediction performance by consolidating features. However, in the process, the distance in representation space becomes blurred, further compromising the ability to detect out-of-distribution instances. Therefore, we introduce a Spectral Normalized Residual connection to replace the traditional residual connection in the graph convolutional models, where the main stream consists of cascaded layers.

Proposition: Restricting the upper *Lipschitz* bound in residual connections is essential to preserve feature space distances. The proof is in the appendix.

Considering a traditional residual connection using one convolutional kernel, where $r(\boldsymbol{x}) = \phi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, ϕ, $\boldsymbol{M}$ and $\boldsymbol{b}$ are activation function, weight matrix and bias respectively. We apply the spectral normalization on the weight matrix as following:

$$\boldsymbol{W}_{sn} = \begin{cases} \boldsymbol{W} \cdot c/\lambda & c < \lambda \\ \boldsymbol{W} & c \geq \lambda \end{cases} \tag{3.39}$$

where $c > 0$ is a coefficient to adjust the norm bound of spectral normalization, and λ is the spectral norm, i.e. the largest singular value of the weight matrix $\boldsymbol{W}$ [Beh+19]. In doing so, we control the *Lipschitz* upper bound of the residual connection by adjusting the hyperparameter c, since:

$$\begin{aligned} ||\sigma(\boldsymbol{W}_{sn}\boldsymbol{x} + \boldsymbol{b})||_{lip} &\leq ||\boldsymbol{W}_{sn}\boldsymbol{x} + \boldsymbol{b}||_{lip} \\ &\leq ||\boldsymbol{W}_{sn}\boldsymbol{x}||_{lip} \leq ||\boldsymbol{W}_{sn}||_{sn} \leq c \end{aligned} \tag{3.40}$$

where $||\cdot||_{lip}$ means *Lipschitz* norm, e.g., $||\boldsymbol{W}_{sn}\boldsymbol{x}||_{lip} = ||\boldsymbol{W}_{sn}\boldsymbol{x}_2 - \boldsymbol{W}_{sn}\boldsymbol{x}_1||/||\boldsymbol{x}_2 - \boldsymbol{x}_1||$, and $||\cdot||_{sn}$ represents the spectral norm.

3.5.4 Feature Space Distance Measurement using Gaussian Process

By implementing the aforementioned model, we obtain a distance-aware feature space. We collect the high-dimensional features of all known data (trainset) output by the second separable kernel in the classifier, as shown in Fig 3.14, and fit a multivariate normal distribution per class to quantify the prediction distance in the feature space, as follows:

$$\boldsymbol{F} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} \in \mathbb{R}^{n \times c}, \ \boldsymbol{\Sigma} \in \mathbb{R}^{n \times c \times c} \tag{3.41}$$

where c is the channel dimension of feature map $\mathbf{F}$, n is number of action categories, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrices, respectively.

In evaluation phase, we calculate the marginal likelihood of the unknown feature representation $\boldsymbol{f}'$ under the prior density $\boldsymbol{F}$ per class:

$$p(\boldsymbol{f}')_i = \sum_j^c p(f'_j | f_{i,j}) p(f_{i,j}), \quad i \in [0, n-1] \tag{3.42}$$

where p is the Gaussian log probability, c is the number of channels, $f'j$ and fi, j are the scalar elements of $\boldsymbol{f}'$ and $\boldsymbol{F}$, respectively. Since the log-probability does not have the normalization ability like the *softmax* function, the predicted label is selected by the one with the largest log probability and greater than a threshold. In doing so, the certainty of the prediction is directly demonstrated by the log probability, and the feature space distance is transformed into the log probability space distance.

In comparison, we utilize several existing measuring modules: the exponential distance [Van+20a] and Laplace-approximated neural Gaussian process [Liu+20a]. Experimental analyses of the proposed uncertainty quantification method are introduced in Section 4.6.

3.6 Multiple Objects Tracking

Tracking multiple targets, including both humans and objects, provides valuable insights into user behavior within crowd dynamics. This is particularly crucial in the field of healthcare, where elderly individuals may have diverse care needs. Developing an effective tracking algorithm is essential for understanding and detecting these needs.

3.6.1 Human Tracking

As a sub-task of multiple object tracking, human tracking primarily addresses the recognition and retrieval of individuals across different cameras and scenes. It employs computer vision technology to determine the presence of specific pedestrians in an image or video sequence. This technology intelligently recognizes pedestrians based on attributes such as clothing, posture, and hairstyle. Person re-identification is widely recognized as a subproblem of image retrieval, enabling the retrieval of a

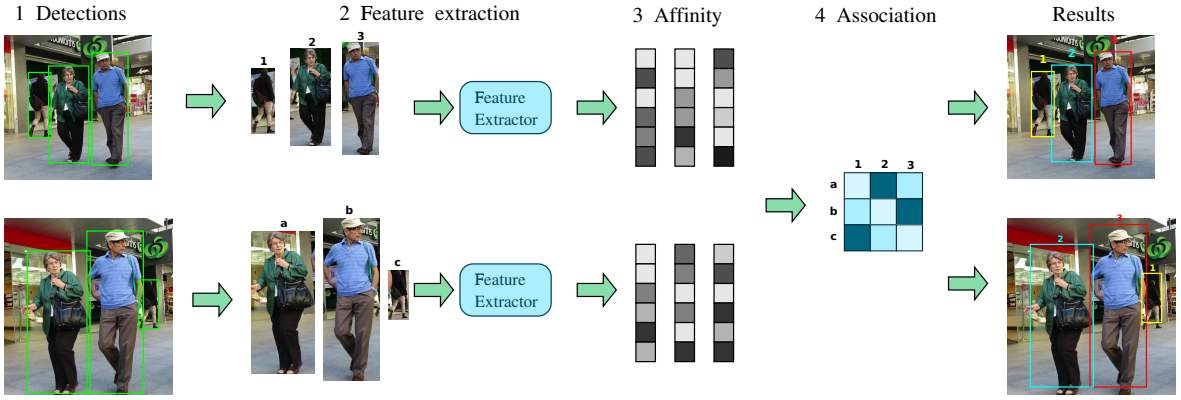


Figure 3.15: The workflow of person re-identification on an example of the Multi-Object Tracking benchmark [Mil+16].

monitored user image across various devices. It serves to overcome the visual limitations of fixed cameras and can be seamlessly integrated with person detection and tracking technologies. Consequently, person re-identification finds extensive applications in intelligent video surveillance and security. In our case, this technology is implemented to establish a personal schedule within the dynamics of a crowd.

The algorithm consist of four steps as follows:

1. **Detection stage:** as shown in Fig 3.15, an human detection algorithm analyzes each input frame to identify where is human bounding boxes on image, also known as 'detections'.

2. **Feature extraction stage:** one or more feature extraction algorithms analyze the detections and/or the tracklets to extract appearance, motion and/or interaction features. Optionally, a motion predictor predicts the next position of each tracked target;

3. **Affinity stage:** features and motion predictions are used to compute a similarity/distance score between pairs of detections and/or tracklets;

4. **Association stage:** the similarity/distance measures are used to associate detections and tracklets belonging to the same target by assigning the same ID to detections that identify the same target.

Detection Stage

The prevalent approach in multiple object tracking (MOT) is tracking-by-detection, leveraging the rapid advancements in object detection. Notably, object detection has made significant strides, especially with the evolution of deep learning technology. Tracking-by-detection involves detecting a set of bounding boxes from a video sequence, which are then utilized in the tracking process. During tracking, the current detections are associated with their historical counterparts and assigned either an existing ID or a new ID. Consequently, the quality of detection plays a crucial role in determining the tracking performance.

Within the detection-based tracking framework (tracking-by-detection), this study compares various multiple object tracking methods. The evaluation of performance metrics relies on the MOT Challenge [Mil+16], a standardized platform that not only provides a unified verification video but also includes a specific type of detection method in the standard video at times. The performance metrics are based on proved detections from MOT16 and MOT17, each generated by different detectors. For instance, MOT16 utilizes DPM (Deformable Parts Model) [Fel+09] three different detectors. Additionally, this study employs YOLO [Red+16], an end-to-end detector, to acquire the detections. The specifics of each detector are elucidated in the subsequent subsections in the order of DPM, Faster R-CNN, YOLO, and SDP.

The *Deformable Part Models (DPM)* [Fel+09], proposed by Felzenszwalb in 2008, represents a fundamental and extensively employed traditional method in the field of object detection, particularly before 2012. Prior to the advent of Convolutional Neural Networks (CNN), DPM stood as the state-of-the-art object detector for several years. Renowned for its robustness to object deformation, DPM has served as a foundational component in numerous classification, segmentation, and pose estimation algorithms. In both MOT16 and MOT17, DPM is utilized as the detector, and the ensuing detection results play a pivotal role in the subsequent tracking phase of this study.

The DPM algorithm represents an object as a combination of parts with a certain spatial relationship between them. The workflow of DPM is that extract features from the input image, make a corresponding model template for a certain component, slide and calculate the score in the original image, and determine the target location according to the distribution of score.

For the feature extraction part, the Histogram of Oriented Gradients (HOG) is commonly used as a feature descriptor, in which the distribution of gradients or edge directions are captured. These features are useful for object recognition because the appearance and shape of an object can be characterized by the distribution of local intensity gradients. The model template represents a certain part of the object, and it is equivalent to a manual designed convolutional kernel. For example, if the object to be detected is a person, the DPM model could have separate templates for the head, torso, arms, and legs. The next step is to slide the model templates over the feature map obtained from the input image. At each position, a score is calculated based on the similarity between the model template and the feature map. This score indicates how well the model template matches the local features of the image. The position of the object in the image is determined based on the distribution of scores. The location with the highest score is usually considered the position of the object.

Since the gradient direction calculated by the HOG is 0 to 180 degrees of insensitive feature, a lot of feature information will be lost. The DPM implemented an improved HOG, which extracts 0 to 360 degrees sensitive features and remove the feature normalization of four corner neighboring cells. It first extracts the features between 0 and 180 degrees to obtain the 4×9 dimensional features which are spliced to obtain 13 dimensional feature vectors, and then extract the features between 0 and 360 degrees to get the 18 dimensional feature vector, and add the two vectors to get the final feature vector with size of 31.

Although the DPM is intuitive and simple, it is not universal, because the model template used to detect people cannot be used to detect kittens or puppies. So when

you do a detection of an object, you need to manually design the model template in order to obtain a better detection effect which take a lot of time and work. Furthermore, it is not able to adapt to large rotations and has poor stability, which is the general shortcoming of traditional hand-crafted object detection methods.

The *Faster R-CNN* [Ren+15] is one of the three detectors of MOT17. It is the most classic network in the R-CNN series which is also the benchmark work of the two-stage method of object detection. The faster R-CNN is the first end-to-end, and the first near-realtime deep learning detector. By sharing convolutional features with the down-stream detection network, the region proposal step is nearly cost-free. The learned Region Proposal Network (RPN) also improves region proposal quality and thus the overall object detection accuracy. An RPN is a fully convolutional network that simultaneously predicts object bounds and confidence scores at each position. Note that the input of the RPN network is the feature map extracted by the feature extraction network in the Fast R-CNN from the original image. The RPN first predicts multiple region proposals at each feature point on the feature map, where the sliding window locates. The specific method is to map each feature point back to the center point of the receptive field in the original image as a reference point, and then select k anchors with different scales and aspect ratios around this reference point. In the Faster R-CNN, there are 3 scales multiply 3 aspect ratios totally 9 possible anchors. After the sliding window processing and intermediate layer, each feature map will have a channel number 256.

Scale Dependent Pooling and Cascaded Rejection Classifiers (SDP-CRC) [YCL16] is one of the three detectors used in MOT17 challenge. The SDP-CRC proposes a object detection method with both accuracy and efficiency. The SDP stands for scale-dependent pooling, which is used to improve accuracy. The CRC stands for cascaded rejection classifiers, which is used to improve efficiency. SDP-CRC is built based on Fast R-CNN but has made certain improvements to the shortcomings of Fast R-CNN.

Firstly, Fast R-CNN can not detect small objects well, which is due to the fact that Fast R-CNN only pools from the last convolutional layer to get the information of bounding box. Secondly, multi-scale input fundamentally limits the applicability of very deep architecture due to memory constraints and additional computational burden. Thirdly, pool a number of region proposals and feed them into fully connected layers with high dimension are time consuming and redundancy. Therefore, SDP-CRC Figure 3.14 uses the convolutional features of every layers to reject easy negatives with cascaded rejection classifiers and evaluate surviving proposals using scale dependent pooling to increase performance in both accuracy and efficiency.

The SDP divides the input region of interests (RoIs) into 3 different groups (small region, mid region and large region) according to their scale and give these input RoIs into different SDP layers. For example, if the height of a proposal region is between 0 to 64 pixels, the features on the third convolutional layer (such as Conv3 in VGG) are used. If the height of a proposal region is larger than 128 pixels, the features of the last convolutional layer (for example, Conv5 in VGG) are used. These three branches (Conv3, Conv4, and Conv5) project RoIs into a high dimensional feature map and feed these projection into a pooling function. Each branch contains two subsequent fully connected layers, ReLU activation and Dropout layers, which regress bounding box and obtain class score. The advantage of this structure is that more information mainly for small objects can be saved. Instead of artificially re-

sizing the input images, the SDP selects a proper feature layer to describe an object proposal. It reduces computational cost and memory overhead. The CRC is designed to further reduce the number of proposals. Instead of directly fusing the features of each layer, the SDP-CRC establish it's own classifiers of every different feature layers. According to the principle of boosting classifiers, SDP-CRC can quickly negate an easy negative thanks to the previous base layer is a weak classifier. The scale-dependent pooling (SDP) improves detection accuracy especially on small objects by fine-tuning a network with scale-specific branches attached after several convolutional layers. The cascaded rejection classifiers (CRC) effectively utilize convolutional features and eliminate negative object proposals in a cascaded manner, which greatly speeds up the detection while maintaining high accuracy.

YOLO [Red+16] is employed as a detector in this work as well, which is able to give the object detection results for the following part of object tracking. It is the most classic network in the field of object detection which is also the benchmark work of the one-stage method. Differently from R-CNN series, YOLO series detect object as a regression problem based on deep learning. YOLO series do not show the process of obtaining the proposal region, but use the entire image as the input, and then obtain the location and category of the bounding box through a regression process.

Feature Extraction and Motion Prediction Stage

Feature extraction refers to the process of transforming unrecognizable original data into features that can be recognized by the algorithm. For example, a picture is composed of a series of pixels (original data), these pixels themselves cannot be used directly by the algorithm. But if these pixels are converted into a matrix (numerical features), then the algorithm can use them. A feature is a piece of information related to solving a computing task related to a certain application. Features may be specific structures in the image, such as points, edges, or objects. Features may also be the result of general neighborhood operations or feature detection applied to the image. In this work, a novel convolutional neural network (CNN) is introduced to generate feature descriptor. The final results is compared with a traditional algorithm (the HOG descriptor).

The CNNs descriptor: as shown in Table 3.1, the proposed CNN architecture is based on Deepsort [WBP17] with residual connections. It consists of two convolutional layers and followed by six residual blocks. The global feature map with dimension 128 is computed in dense layer 10. According to person re-identification, we can add a linear layer as the classifier to cluster different people. Except the CNN architecture with 128 dimension, architectures with 32, 64, 256 dimension are also evaluated on Market-1501 [Zhe+15] and Mars [Zhe+17] datasets.

The HOG descriptor. The basic idea of this algorithm is implementing the HOG method to generate its gradient features and to cluster through a Support Vector Machine (SVM). The detection window is scanned at all positions and scales of the entire image, and non-maximum suppression is performed on the output pyramid used to detect the target. In reality, targets will appear in different environments, and the lighting will be different. Color space normalization is to normalize the color information of the entire image to reduce the impact of different lighting and backgrounds. In order to improve the robustness of detection, Gamma and color space

Table 3.1: Overview of the CNN architecture with 128 dimension.

Layer name	Kernel size/stride	Output size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 32 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 32 \times 8$
Dense 10		128
Batch norm		128

normalization are introduced as preprocessing methods for feature extraction. HOG also evaluated the expression of different image pixels, including gray space, and finally verified that RGB and LAB color space can make the detection results roughly the same and can have a positive impact. On the other hand, in their research, the author used two different Gamma normalization methods on each color channel, taking the square root or using the logarithmic method, and finally verified that this preprocessing has almost no effect on the detection results. Furthermore, Gaussian smoothing cannot perform on the image, because the smoothing will reduce the recognition ability of the edge information of the image and affect the detection result.

Kalman filter is implemented to predict the next position of each tracked target. Kalman filter is widely used in the fields of unmanned aerial vehicle, autonomous driving, satellite navigation, etc. In simple terms, its function is to update the predicted value based on the measured value of the sensor to achieve a more accurate estimation. Suppose we want to track the position change of the car, Kalman filter is divided into two processes: prediction and update. Prediction process: when a small car is moved and its initial positioning and moving process are Gaussian distributions, the final estimated position distribution will be more scattered, whic leads to less accurate. Update process: when a small car is observed and positioned by the sensor, and its initial positioning and observation are Gaussian distribution, the position distribution after observation will be more concentrated, which is more accurate.

In the tracking phase, the following two states of track need to be estimated: mean and covariance values. Mean is the position information of the target, which is composed of the center coordinates of the bounding box (c_x, c_y), the aspect ratio $r = height/width$, the height h, and the respective speed change values. It is represented by an 8-dimensional state vector as $x = [c_x, c_y, r, h, v_x, v_y, v_r, v_h]$, each speed value is initialized to 0. Covariance value represents the uncertainty of the target location information, represented by an 8×8 diagonal matrix. The larger the number in the matrix, the greater the uncertainty, and it can be initialized with any value. Assume

that person move with constant acceleration, the state vector is estimated as follows:

$$x_t = Ax_{t-1} \tag{3.43}$$
$$P_t = AP_{t-1}A^T + Q \tag{3.44}$$

where A is the state-transition model and describe the state changes, e.g., $c_{x,t} = c_{x,t-1} + v_{x,t-1}\Delta t$.

In the update phase, the Kalman gain, the optimal state x and the convariance matrix P are obtained as follows:

$$K_t = P_t H^T [HP_t H^T + R]^{-1} \tag{3.45}$$
$$x_t = x_t + K_t[y - Hx_t] \tag{3.46}$$
$$P_t = [I - K_t H]P_t \tag{3.47}$$

where y is measurement of $[cx, cy, r, h]$. R is the noise matrix of the detector, which is a 4×4 diagonal matrix. The values on the diagonal are the two coordinates of the center point and the noise of the width and height, which are initialized with any value, and the noise of the width and height are generally greater than the noise of the center point. This equation first maps the covariance matrix $P_t$to the detection space, and then adds the noise matrix R.

Affinity Stage

In the affinity stage, the similarity or distance score between the predicted states and newly arrived measurement are calculated. The most classic affinity method is the Hungarian algorithm. In this work, we improve Hungarian algorithm by design an optimal cost matrix between the detections and the trackers.

The squared Mahalanobis distance (covariance distance) is used to measure the distance between the predicted Kalman states and newly arrived measurements as Equation 3.48. Since both are represented by Gaussian distribution, it is very suitable to use Mahalanobis distance to measure the distance between the two distributions.

$$d_{i,j}^m = (\mathbf{d}_j - \mathbf{y}_i^T)\mathbf{S}_i^{-1}(\mathbf{d}_j - \mathbf{y}_i) \tag{3.48}$$

where ($\mathbf{y}_i$, $\mathbf{S}_i$) are the projections of the i-th track distribution into measurement space and $\mathbf{d}_j$ is the j-th bounding box detection.

The reason why Euclidean distance is not used is that the spatial distribution of d j and y i are different. The calculation result of Euclidean distance ignoring the spatial distribution cannot accurately reflect the true distance between these two. The Mahalanobis distance takes state estimation uncertainty into account by measuring how many standard deviations the detection is away from the mean track location. An indicator is used to denote the similarity as follows:

$$b_{i,j} = 1, \quad d_{i,j}^m \leq t \tag{3.49}$$

where t is manual selected threshold value. When the detection and tracker is admissible, the value is equal to 1. However, there is no upper limit to the value range of the Mahalanobis distance, which is not conducive to determining the threshold.

Therefore, the connection between Mahalanobis distance and Chi-square distribution is utilized to determine the threshold. The Mahalanobis distance meets the 95% confidence threshold in different dimensional states according to different dimension of states. For dimensions $1-9$, there are thresholds 3.842, 5.992, 7.815, 9.4877, 11.070, 12.592, 14.067, 15.507, 16.919. The speed state is not considered during the comparison, so there are only four dimensions left, namely the threshold is 9.4877.

The Mahalanobis distance is a suitable correlation measure when motion uncertainty is low. However, the Kalman filter's predicted state distribution only provides a rough estimate of the object's location. In particular, unaccounted camera motion can introduce rapid displacements in the image plane, making the Mahalanobis distance a rather uninformed metric for tracking through occlusions. Therefore, we integrate a second metric into the assignment problem, as shown in Equation 3.50. The cosine distance is used to measure the smallest distance of the appearance features between the i-th tracker $\mathcal{T}^i$ and j-th detection $\bar{f}_j$ in appearance space, which assign the ID more accurately.

$$d^f_{i,j} = min\{1-\bar{f}_j^T \bar{t}^i_k \mid \text{for } \bar{t}^i_k \in \mathcal{T}^i\} \tag{3.50}$$

where $\bar{f}_j$ is the feature vector of the detected bounding box processed by the network. $\bar{t}^i_k$ is an element of i-th tracker $\mathcal{T}^i$, and each tracker save the last 100 associated feature vectors. The feature vector is saved based on the similar indicator as Equation 3.49 with threshold $t = 0.2$.

The final distance between detection and tracker is obtained by combing the Mahalanobis distance and feature distance as follows:

$$d_{i,j} = \lambda d^m_{i,j} + (1-\lambda) d^f_{i,j} \tag{3.51}$$

The influence of each metric on the combined association cost can be controlled through the hyperparameter λ. In experiments, $\lambda = 0$ is a reasonable choice when the camera movement is large. Because the camera shakes, the uniform motion model based on Kalman's prediction does not work well, so the Mahalanobis distance actually has no effect. But note that the Mahalanobis gate is still used to disregard infeasible assignments of possible object locations, which are inferred by the Kalman filter.

Association Stage

After the distance is obtained, the Hungarian algorithm is utilized to find the best associated pairs between detection and trackers. However, it fails when a person is covered by another person for a long time. Therefor, a new matching strategy is introduced in this work - cascade matching.

The cascade matching strategy can improve the matching accuracy and is dedicated to solve the situation that the target is occluded for a long time. In order for the current detection to match the track that is closer to the current moment, the detection will give priority to the track that has a shorter disappearance time when matching. From the trajectory with missing age= 0 (the ones are matched in every frames and never lost) to the trajectory with missing age= 70 (the trajectory have lost for 70 frames that is the maximum missing time), the detection results are matched one by one. In other words, the trajectory that has not been lost will given priority to match, and the trajectory that has been lost for the longest time is matched last.

In the final matching stage, there will be a IOU matching between unconfirmed tracks and unmatched detection, unmatched tracks, which are the results of the matching cascade. This helps to solve the problem for the sudden changes of appearance, which caused by partial occlusion with static scene geometry. This extra IOU matching between the detection and tracks can increase robustness against erroneous initialization.

Compare to the cascade matching strategy, the IOU matching is aimed to deal with overlapping. More specially, when two targets are entering an overlapping situation, the IOU matching method selects detection with a similar scale to the tracker. So only the front target is tracked and covered target does not affect the assignment cost matrix.

3.6.2 Objects Tracking

Besides re-identify person, the related objects information is important as well. Some practices of 3D object tracking have been developed over the past three decades. For example, the work [ZWZ19] track objects by registering point cloud with meshed CAD model. However, the method relying on a high quality point cloud or a short distance between objects and depth camera. Many other researchers focus on extracting 2D features from detected bounding boxes and assign the corresponding ID based on feature distance [WB18]. These methods normally have a promising performance in the scene with less occlusions. When occlusions occurs, the ID of two targets are easily exchanged because their features overlap. For the calibrated cameras, there is an option to combine 2D image information and 3D points of the same scene.

Depth-based tracking has become more popular in the last decade, as RGB-D devices became more available. Depth maps are powerful in the area of feature extraction, considering the price-quality ratio. Depth images contain relevant information about the distance to the scene objects from a viewpoint, geometrical relations in the scene and shape features. This data provides multiple opportunities for 3D modeling, simulations, etc. However, the disadvantage of this method in a single depth channel lies in an inability to receive information about the objects from other angles and perspectives.

Similar to task of person re-identification, YOLO [Red+16] is applied to detect objects. The algorithm is used every time a new frame is captured. Nonetheless, object detectors, even the best ones, are not perfect and cannot detect all objects continually. In every object detector, there is a backbone network, which is responsible for extracting features of each detected objects [Li+20b]. Additionally, there may be objects, that need to be detected according to the task. However, it is possible that the backbone network of an object detector is not trained to recognize them. These are the reasons why a user should give a number of frames as a parameter *–frame_number*. It represents a number of times when undetected objects should be chosen manually. After the first selection of undetected objects, the user is asked if in the rest of frames, the objects do not change their coordinates. If the user is confident in this fact, the rest of the frames are simply demonstrated after that. Otherwise, the process should be repeated. This step is needed to create a stable base for the recall

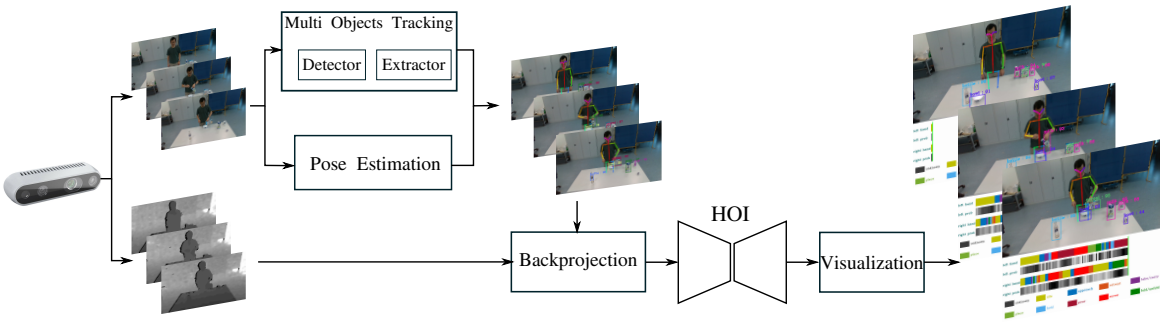


Figure 3.16: An overview of the real-time system for understanding human-objects interaction.

Figure 3.17: A local coordinate on human body from front (right) and 45° side (left) views, where X_0, Y_0 and Z_0 are the x, y, and z axes respectively.

step and the next algorithm, called Kalman filtering.

A Kalman filter helps to predict the next position of objects. It could be the case that an object was not correctly detected or an object cannot be seen in the frame, e.g., a moving ball is hidden under or behind another object for a few frames and appear again. Therefore, the Kalman filter helps to estimate its position and to compare the predicted position with new observation. Instead of observing the bounding box states in the task of person re-identification, here the 3D center position is tracked by the Kalman filter. In the words, the state vector is $x = [C_x, C_y, C_d, v_x, v_y, v_d]$. The rest stages are same as the task of person re-identification.

Experimental analyses of multi-object tracking algorithms are presented in Section 4.2.

3.7 Real-Time System of Understanding Human-Object Interaction

This section describes a real-time system of using our HOI model. Designing a real-time human action recognition system involves considering various factors to ensure its effectiveness, such as RGB frames, depth frames, field of view (FoV), frame rate and accuracy. A high-quality RGB camera provides a rich information of human motion. Additional depth information enables the system to understand the 3D structure of the scene. The accuracy of the depth sensor is crucial for correctly understanding

the spatial relationships between different objects and body parts in the scene. To cover the entire area where human actions need to be recognized, an appropriate field of view need to be ensured. This involves using a wide-angle lens to capture a broader scene. To enable real-time processing, a high frame rate need to be achieved. The frame rate should be sufficient to capture fast-paced human actions without significant latency. Aim for at least 30 frames per second (fps) or higher.

As shown in Figure 3.16, the real-time system takes RGB and depth frames from a depth camera as the input and estimate human pose and track objects features. These features will be stored in an array and fed into a pre-trained HOI model with the encoding-decoding process to get its action prediction and segmentation results. Experimental evaluation of the real-time system on the real world dataset is introduced in Section 4.7.

For the human pose estimation part, an open source skeleton estimation toolbox[Cao+19] is employed to extract 2D skeleton information from a single frame. For the multiple objects tracking, an object detector [Red+16] is used to detect the position of object candidates, and a Kalman filter is combined with a CNN-based classifier to find correspondences for each detected object. For the human-objects interaction model, we utilize our uncertainty quantified temporal fusion graph convolutional network, which can predict action labels with novelty. Note that the extracted pose and object information is in 2D space, and our model requires 3D input. Therefore, we first project all 2D information along the depth direction to a 3D space in the camera coordinate. Then the local coordinates on the human body are established based on the 3D skeletal information, where the origin point the neck joint, the z direction is from the mid-hip to the neck joint, the y direction is from the neck to the left shoulder, and x is the cross product of the y and z axes, as shown in the Fig. 3.17. All joints position information is transformed to the new coordinate.

Chapter 4

Experiments

This chapter presents the experimental setup and results. The setup includes hardware, datasets and the evaluation metrics, and results list the performance of different methods in multi-objects tracking, action recognition, action segmentation, event detection, uncertainty quantification and real-time system for understanding of human-object interaction.

4.1 Experimental Setup

4.1.1 Hardware

Depth cameras play a pivotal role in capturing accurate 3D information for understanding human behavior. The cameras under investigation in this work are the Intel RealSense D415, D435, D455, L515, Microsoft Kinect v2, and Azure Kinect. The comparison is based on key technical specifications, performance metrics, and features of each camera.

Table 4.1: Comparison of popular existing depth cameras.

Cameras	Intel D415	Intel D435	Intel D455	Intel L515	Kinect v2	Azure Kinect
Depth technology	Active IR Stereo	Active IR Stereo	Active IR Stere	LiDAR	Time-of-Flight	Time-of-Flight
Depth resolution	1280×720	1280×720	1280×720	1024×768	512×424	512×512
Depth frame rate (fps)	90	90	90	90	30	30
Depth FoV (°)	65×40	87×58	87×58	70.4×56.2	70.6×60	70.6×60
Depth range (m)	$0.5-3$	$0.3-3$	$0.6-6$	$0.3-9$	$0.5-4$	$0.3-5$
Depth accuracy	2% at 2 m	2% at 2 m	2% at 4 m	5 to 14 mm at 9 m^2	1.5% at 4 m	1% at 4 m
Aligned max resolution	848×480	848×480	848×480	1024×768	512×424	640×576
Aligned frame rate (fps)	30	30	30	30	30	30
Cost ($)	272	334	419	589	199	399

As demonstrated in Table 4.1, depth cameras can be clustered into three classes according to depth technologies: active infrared (IR) stereo, LiDAR and Time-of-Flight.

Active infrared stereo: Active infrared stereo systems use two infrared cameras to capture stereo images and compute depth based on the disparity between corresponding points in the images. While they provide depth information in a high frequency, their accuracy is slightly lower compared to LiDAR and ToF technologies, especially in complex or dynamic environments. Active infrared stereo systems are often more cost-effective compared to LiDAR and ToF solutions. They use off-the-shelf

components and relatively simple algorithms for depth computation, contributing to their affordability. The Intel RealSense D series use this technology.

LiDAR: LiDAR technology is known for its high accuracy in capturing 3D spatial information. It uses laser pulses to measure distances with exceptional precision, making it suitable for applications that require detailed and accurate depth perception. LiDAR systems are generally more expensive due to the complexity of components, including lasers, mirrors, and precise timing mechanisms. This has been a limiting factor for their widespread adoption, especially in consumer applications. This technology is implemented in the Intel RealSense L515 camera.

Time-of-Flight (ToF): ToF technology measures the time taken for a light signal to travel to an object and back, providing depth information. ToF sensors can offer good accuracy, which is slightly lower than that of LiDAR, especially in longer ranges. However, it takes a long warm-up time to achieve such accuracy in Kinect cameras [Kur+22]. ToF sensors are more cost-effective than LiDAR systems. They can be integrated into various devices like smartphones, gaming consoles, and consumer electronics, making them more accessible to a broader range of applications.

While LiDAR and Time-of-Flight technologies offer higher accuracy, active infrared stereo systems can provide a balance between accuracy, frequency of operation, and cost-effectiveness. In this work, we select Intel RealSense D series camera as our perception sensors.

4.1.2 Datasets

NTU-RGB+D [Sha+16b] stands out as one of the largest and most demanding 3D action recognition datasets, comprising 56,000 action clips featuring 3D skeleton data across 60 action classes. The performances are enacted by 40 volunteers and are captured by three cameras from distinct horizontal angles: −45°, 0°, and +45°. Each clip involves at most 2 subjects. The dataset authors recommend two benchmarks for evaluation: 1) **cross-view** (X-View). This benchmark comprises 37,920 videos for training and 18,960 videos for validation. The training set is captured by cameras 2 and 3, while the validation set is captured by camera 1. 2) **cross-subject** (X-Sub). This benchmark includes 40,320 videos for training and 16,560 videos for validation. The two sets involve performances by different subjects. In this study, we adhere to the recommended benchmarks: **cross-view** (X-View) and **cross-subject** (X-Sub).

To further evaluate the performance of the proposed model, we partitioned the X-View dataset into two subsets: Body Parts Related (BPR) and Pose Related (PR) validation datasets. The BPR action classes encompass numbers 1, 2, 3, 4, 6, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 28, 34, 37, 38, 39, 40, 41, 44, 45, 46, 47, 48, while the remaining classes constitute the PR dataset.

Fall event dataset: To distinguish fall-down events from similar actions, we curated a test dataset by extracting 420 sitting-down and 405 ground-lift 3D skeleton examples from the NTU-RGBD dataset [Sha+16b]. Additionally, we included 280 skeleton examples from 46 different actions in the test dataset.

Following the recommendations in the NTU-RGBD dataset [Sha+16b], we adhere to cross-subject (CS) and cross-view (CV) evaluation criteria to benchmark our model against other popular existing methods. For CS evaluation, the subjects are divided

into training and testing groups, with training subjects assigned the IDs 2, 3, 5, 7, and 8. In CV evaluation, training utilizes samples from camera views 2 and 3, consisting of one front view and two side views. The testing set comprises samples from camera view 3, encompassing diagonal views.

Kinetics [Kay+17] stands as a more challenging human action recognition dataset, boasting 300,000 videos spanning 400 action classes sourced from YouTube. Each clip in Kinetics lasts around 10 seconds. The dataset provides solely raw video clips without accompanying skeleton information. At most two people are selected in the multi person videos based on the average joint confidence. The dataset is divided into a training set (240,000 clips) and a validation set (20,000 clips). In this study, we use 2D skeleton dataset (240,000 clips for training, 20,000 clips for validating) that generated by Yan *et al.* [YXL18] using the OpenPose toolbox [Cao+19].

Bimanual Actions Dataset [Kre+21] was meticulously curated for the purpose of human-object interaction detection in a third-person perspective. Comprising 540 recordings, with a cumulative runtime of 2 hours and 18 minutes, this dataset offers framewise predictions for 12 objects (in the form of 3D bounding boxes) and 6 subjects (depicted through 3D skeletons). Notably, the dataset encompasses both hands of the subjects, categorizing them into one of 14 possible interaction categories. Each recording captures a solitary individual engaged in the execution of a complex daily task within one of the two predefined environments: a kitchen or a workshop. For benchmarking purposes, the dataset's creators recommend the **leave-one-subject-out** cross-validation approach, where records from one subject are reserved for validation, and the recordings from the remaining subjects are utilized for training.

The **IKEA Assembly Dataset** [Ben+20] stands out as a notably challenging and intricate human-object interaction dataset, boasting a comprehensive collection of 16,764 annotated actions. Each action is accompanied by an average of 150 frames, contributing to a cumulative duration of approximately 35.27 hours. The dataset's creators have introduced a rigorous **cross-environment** benchmark, wherein the test environments are deliberately distinct from those present in the training set, and vice versa. The training set encompasses 254 scans, while the test set comprises 117 scans.

In this study, we assess the efficacy of the proposed action recognition model on the **NTU-RGBD** and **Kinetics** datasets. The proposed action segmentation models undergo evaluation on the **Bimanual Actions** and **IKEA Assembly** datasets. Additionally, uncertainty quantification experiments are carried out on these two datasets. Furthermore, an event detection experiment is executed using the **Fall event dataset**.

For the NTU-RGB+D dataset, we maintain the same input data size as in [Shi+19b], with a maximum of 2 persons per sample and a frame limit of 300. In the case of the Kinetics dataset, we generate samples with 150 frames and 2 persons each, applying slight random rotations and translations. For the Bimanual Action dataset [DWA19], we utilize the centers of 3D object bounding boxes and 3D human skeleton data provided by the authors [DWA19]. In the ablation study, subject 1 is left out for validation, and in the comparison with other methods, we adopt a leave-one-subject-out cross-validation approach. In the IKEA Assembly dataset [Ben+20], we use the provided centers of 2D object bounding boxes and 2D human skeleton data, following the cross-environment benchmark.

TUM HOI Dataset. In addition to publicly available human-object interaction

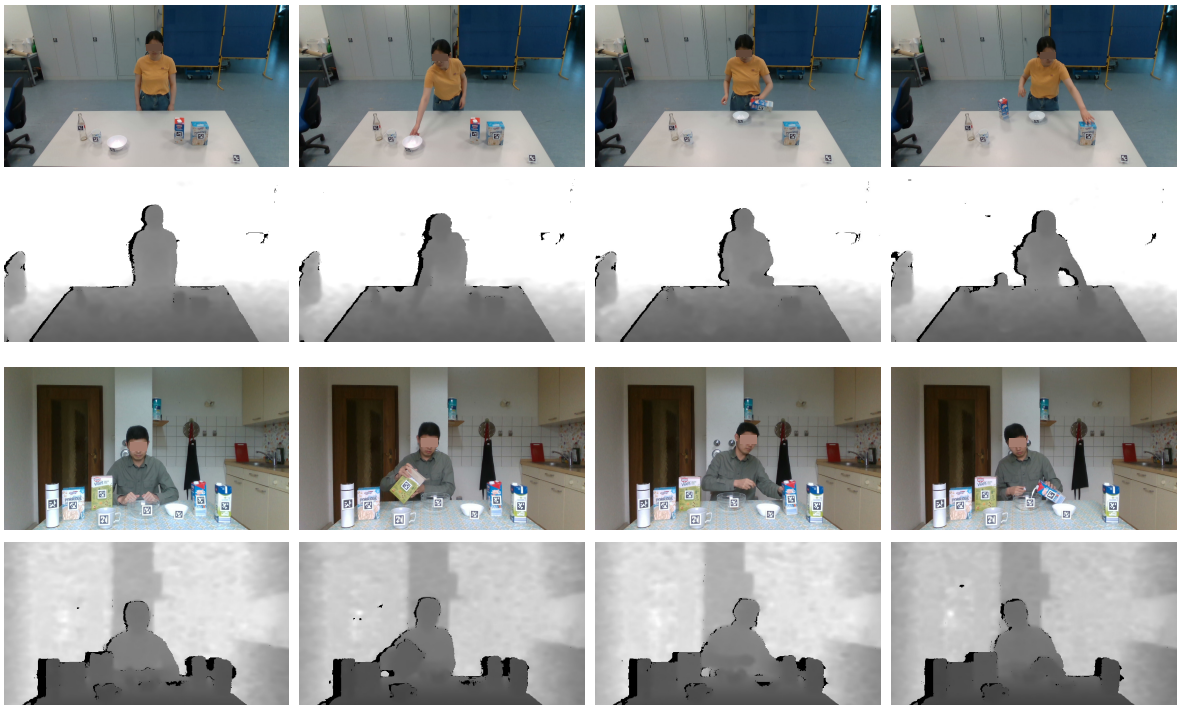


Figure 4.1: Examples of preparing breakfast in standing and sitting from the TUM HOI dataset.

datasets, we have curated our own kitchen action dataset. To account for the impact of human posture on action recognition, participants were instructed to carry out a "making breakfast" task in two distinct poses: standing and sitting, as illustrated in Figure 4.1. Considering the variability in human poses during real-world tasks, particularly between standing and sitting, is crucial, as these different postures can affect action recognition. Factors such as occlusion of body parts and action distortion may vary with pose. In the standing pose segment, six subjects performed the task with ten repetitions each. The recording involved eight objects from five distinct classes (milk, cereal, bowl, cup, and bottle), with cups and bottles serving as non-interacting elements. While milks, cereals, and bowls directly interacted with subjects, a single video sample did not feature multiple objects from the same class simultaneously. Throughout the capture process, all tasks were executed against a consistent background (lab) and under the same camera configuration (approximately 1.75 meters in height). In this segment, the video samples typically lasted around 12 seconds, with the actions being demonstrative in nature. Due to the standing pose, occlusion of body parts was nearly nonexistent, as depicted in the top two rows of Figure 4.1.

In the sitting part, six subjects participated in the recording, performing the task six times each while in a sitting pose. The recording involved the use of eleven different items, including two milks, three cereals, two bowls, two cups, and two bottles. Similar to the standing part, cups and bottles serve as noise, while milks, cereals, and bowls interact with the subjects. This time, multiple objects of the same class are allowed to be used simultaneously, as illustrated in the figure. The actions took place in two distinct environments (lab and family kitchen), each equipped with unique camera setups (approximately 1.05 m in the lab and approximately 1.15 m in the kitchen). The actions closely resemble real-life scenarios, with two subjects authentically performing the actions in a family kitchen setting.

Because of the sitting pose and the numerous objects on the table, occlusions between body parts are common. Moreover, real-life actions usually include many subconscious minor movements, such as hand adjustments due to the weight of milk while pouring or slight shaking when pouring cereal. These subconscious movements are typically absent in demonstrative actions, as depicted in Figure 4.1 in the bottom two rows. Overall, the complexity of this part is higher due to the presence of occlusions and the replication of real-life actions.

For ground truth annotations, the collected dataset includes both spatial and temporal information. Temporally, the dataset provides the start and end frames for each action, defining the action segment points, and assigns an action label to each atomic action. In real-life scenarios, many actions involve the collaboration of both hands, and a single action label may not suffice to describe a bimanual action. Similar to the Bimacs dataset [DWA19], this dataset also includes action labels for both the left and right hands. The actions are categorized into nine classes: idle, approach, retreat, take, place, hold, pour, screw, and fold. All action labels are manually annotated by the same individual to ensure consistency and avoid biases introduced by different annotators.

For spatial information, the dataset includes skeleton data and bounding boxes for objects. The skeleton data are obtained through OpenPose [Cao+19], as discussed in detail in the next section. Regarding the bounding boxes of objects, a semi-automatic annotation algorithm is employed (see Algorithm 1). This algorithm combines pre-trained YOLO [Red+16], a Kalman filter, and Apriltag [Ols11]. Subsequently, it uses 16,663 images labeled by the algorithm and 5,374 images labeled manually to re-train a YOLO network. Finally, the entire dataset is annotated using the re-trained model.

The collected dataset comprises a total of 96 video samples (36 for sitting and 60 for standing), encompassing 60,732 frames (21,872 frames for standing and 38,860 frames for sitting). Figure 4.3 provides the statistical overview of the collected dataset. In the standing part, video durations vary between 200 and 500 frames, with the majority concentrated around 400 frames. Conversely, the sitting part exhibits a significant increase in video duration, ranging from 600 to 2000 frames, with the majority centered around 1000 frames. This duration discrepancy arises from the fact that actions in the standing part are mostly demonstrative and therefore faster, while actions in the sitting part are real-life actions with a slower tempo. Although the action distribution is similar in both parts, variations exist among different actions, such as the idle duration being approximately 3 times longer than the place duration in both standing and sitting parts.

Market-1501 [Zhe+15] serves as a large-scale public benchmark dataset for person re-identification. It comprises 1501 identities captured by six different cameras, with a total of 32,668 pedestrian image bounding-boxes obtained through the Deformable Part Models (DPM) pedestrian detector. On average, each person has around $\sim$ 3 images from each viewpoint. The dataset is partitioned into two segments: 750 identities for training and the remaining 751 identities for testing. The official testing protocol involves selecting 3,368 query images as the probe set to identify the correct match among 19,732 reference gallery images.

MARS (Motion Analysis and Re-identification Set) [Zhe+17] stands out as another extensive video-based person re-identification dataset, expanding upon the

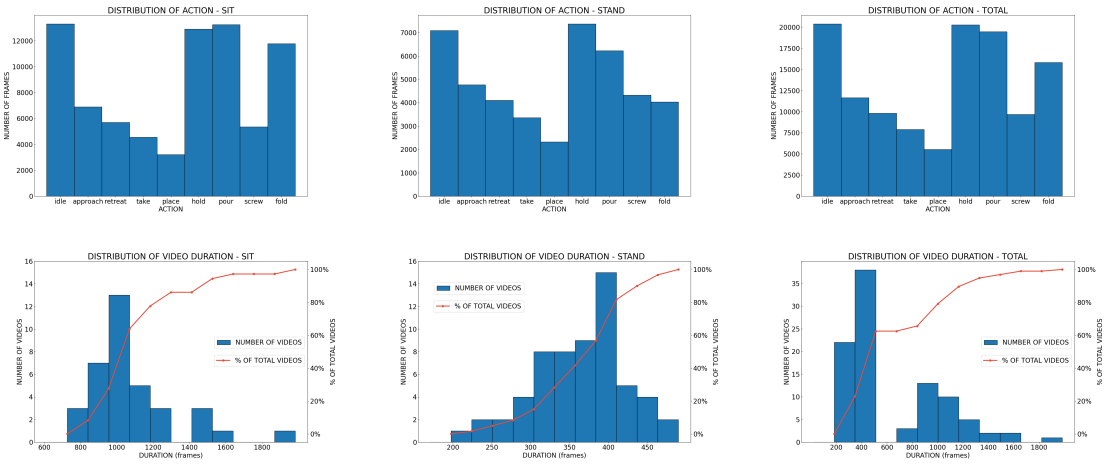


Figure 4.2: Statistics of the TUM HOI dataset: Row 1: frames distribution of different actions in stand (left), sit (middle) and total (right). Row 2: duration distribution of records in stand (left), sit (middle) and total (right).

Market-1501 dataset. It has been curated from six nearly synchronized cameras, encompassing 1,261 distinct pedestrians captured by at least 2 cameras. The dataset's challenge is amplified by variations in poses, colors, illuminations of pedestrians, and poor image quality, posing difficulties for achieving high matching accuracy. Additionally, the dataset deliberately includes 3,248 distractors to enhance realism. To automate the generation of tracklets (mostly $25-50$ frames long), Deformable Part Model (DPM) and GMMCP tracker were employed.

MOT challenge. The MOT17 dataset [Mil+16] is employed as the evaluation dataset for person re-identification. It comprises 14 video sequences, with 7 designated as training sets with annotations, and the remaining sequences serving as test sets.

4.1.3 Evaluation Metrics

To assess the efficacy of the proposed components in action recognition tasks, we introduce an influence ratio metric as follows:

$$r_{J/B} = \frac{Acc_J - Acc_J^*}{Acc_B - Acc_B^*} \tag{4.1}$$

where Acc and Acc^* represent the accuracy of the proposed model and baseline, respectively, J denotes the joint input stream, and B represents the bone input stream. Assuming the improvement is positive, the proposed model is more promising for the joint input stream when $r_{J/B} > 1$ and better for the bone stream otherwise. In the case of a negative influence ratio, i.e., performance drop ratio, the removed component is more beneficial to the joint stream when $r_{J/B} > 1$ and better for the bone stream otherwise.

We evaluate the proposed model on two tasks: HOI framewise recognition and temporal segmentation. For framewise recognition, we utilize two main evaluation

metrics, namely F1-score and F1@k, while for segmentation, we employ the F1@k metric. The F1-score is formulated as: $\mathrm{F1} = \frac{tp}{tp+0.5(fp+fn)}$, where tp denotes true positive predictions, and fp and fn represent false positive and false negative predictions, respectively.

For the F1@k score, we employ common values of $k = 0.10$, 0.25, and 0.50. In this context, the determination of true or false positives for each predicted segment involves comparing the intersection over union (IoU) with a threshold $\tau = k/100$. Incorrect predictions and missed ground-truth segments are categorized as false positives and false negatives, respectively. Additionally, for the multi-class prediction task, both micro-average and macro-average over F1-scores of all classes are adopted as the framewise recognition metrics. In the experiment comparing popular methods on the IKEA Assembly dataset, we utilize top1 and macro-recall metrics to evaluate models.

In assessing fall event detection, a traditional binary classification task, we employ accuracy, recall, and precision as performance metrics.

For measuring the performance of out-of-distribution (OOD) detection, we utilize the area under the receiver operating characteristic (AUROC) and the area under the precision-recall curve (AUPRC). Since the softmax output can lose the true feature distance due to normalization, we use Gaussian log probability values from the multivariate model as the measurement score for AUROC and AUPRC. As recommended in the work by Nixon et al. [Nix+19], we incorporate static calibration error (SCE), adaptive calibration error (ACE), and thresholded adaptive calibration error (TACE) to evaluate calibration error, with the threshold set at 0.01. Additionally, we compare the popular expected calibration error (ECE), although it is originally designed for binary classification methods.

The models and experiments are implemented using the PyTorch deep learning framework, utilizing a single NVIDIA-2080-ti GPU. The optimization strategy employs the widely used stochastic gradient descent (SGD) with Nesterov momentum (0.9), and cross-entropy serves as the loss function for gradient backpropagation.

For the Bimanual Actions dataset [Kre+21] and IKEA Assembly dataset [Ben+20], the training process spans 60 epochs. The initial learning rate is set at 0.1 and is divided by 10 at the 20th and 40th epochs. For the task of action segmentation, a train and test batch size of 16 is chosen due to the large input size. The weight decay is set to 0.0002. In the case of action recognition, a batch size of 16 is used for training, while a batch size of 128 is applied for testing. The weight decay for action recognition is set to 0.0001.

For the NTU-RGBD [Sha+16b] and Kinetics [Kay+17] datasets, the learning rate is initially set to 0.1 and is reduced by a factor of 0.1 at the 60th and 90th epochs. The training process spans a total of 120 epochs.

In the context of the person re-identification task, traditional metrics focus on different types of errors, primarily including Mostly Tracked (MT) trajectories and Mostly Lost (ML) trajectories. In addition to these error metrics, the multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) are utilized

to assess accuracy performance:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{4.2}$$

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{4.3}$$

where m_t, fp_t, and mme_t represent misses, false positives, and mismatches for time t, respectively. The sum of these values is divided by the total number of objects g_t. Additionally, d_t^i denotes the distance between an object and its corresponding hypothesis, while c_t indicates the number of matches for time t. MOTP evaluates the tracker's ability to estimate precise object positions. However, these two metrics alone are insufficient for evaluating performance in preventing ID swapping. Therefore, identification false negative (IDFN), identification false positive (IDFP), and identification true positive (IDTP) are employed to compute identification precision (IDP), identification recall (IDR), and the corresponding F1 score IDF1. More specifically:

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{4.4}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{4.5}$$

$$IDF1 = \frac{2 \times IDTP}{2 \times IDTP + IDFP + IDFN} \tag{4.6}$$

For the real-time system introduced in Section 3.7, we utilize the OpenPose [Cao+19] open-source toolbox to extract 2D skeleton information from a single frame. Multiple objects tracking is achieved by employing YOLO [Red+16] for detecting the position of object candidates. To establish correspondences for each detected object, a Kalman filter is integrated with our CNN-based classifier. For the human-objects interaction model, we leverage our uncertainty-quantified temporal fusion graph convolutional network, capable of predicting action labels with novelty.

4.2 Multiple Objects Tracking

This section introduces the experimental results of multiple objects tracking approaches 3.6 in person re-identification sub-task. All models are trained on Market-1501 [Zhe+15] and Mars [Zhe+17] datasets, and tested on the MOT17 [Mil+16] benchmark.

4.2.1 Comparison of Detectors

The efficacy of detection results significantly influences the Multiple Object Tracking (MOT) metrics. MOT17 offers diverse detection results obtained through three distinct detectors: DPM [Fel+09], Fast R-CNN [Ren+15], and SDP [YCL16]. This

Table 4.2: Different detectors with CNN based feature extraction on the MOT17 benchmark [Mil+16]

Detector [a]			**Size** [b]				**Evaluation Metrics** (%) [c]						
D	F	S	32	64	128	256	IDF1 ↑	IDP ↑	IDR ↑	MOTA ↑	MOTP ↓	MT ↑	ML ↓
×			×				39.4%	67.9%	27.7%	30.5%	22.0%	8.2%	55.7%
	×		×				55.9%	77.9%	43.5%	49.3%	13.4%	19.8%	28.9%
		×	×				65.6%	79.8%	55.6%	64.6%	15.4%	34.2%	22.0%
×				×			39.7%	69.3%	27.8%	30.4%	21.9%	7.7%	56.4%
	×			×			54.9%	76.9%	42.6%	49.3%	**13.3**%	19.4%	29.9%
		×		×			**66.1**%	**80.8**%	**56.0**%	64.4%	15.4%	33.3%	22.7%
×					×		38.5%	67.9%	26.9%	30.2%	21.9%	6.6%	56.8%
	×				×		54.1%	76.1%	41.9%	49.2%	**13.3**%	18.7%	29.9%
		×			×		64.0%	78.6%	54.0%	64.1%	15.3%	32.8%	23.4%
×						×	34.3%	57.5%	24.4%	30.1%	22.3%	8.2%	54.6%
	×					×	50.7%	70.0%	39.7%	49.1%	13.6%	21.8%	27.1%
		×				×	59.3%	71.2%	50.9%	**64.8**%	15.6%	**37.4**%	**19.6**%

[a] D represents the DPM [Fel+09], F is the Fast-RCNN [Ren+15] and S is the SDP [YCL16], the CNN model are trained on the Market-1501 [Zhe+15] and Mars [Zhe+17] dataset.
[b] The size of output dimension.
[c] The best results comparing all modifications are in **bold**; The best results between detectors with the same output dimension are underlined.

section aims to assess the impact of detector performance on MOT metrics. The outcomes of the different detectors are summarized in Table 4.2. Notably, the SDP detector excels across all metrics, particularly demonstrating a 30% higher MOTA than the DPM detector. Moreover, the Faster R-CNN detector achieves a MOTA performance of 49.2%, nearly 19% higher than the DPM detector. Surprisingly, the dimension size of the output layer has minimal impact on the results. Based on these experimental findings, two promising detectors, namely Fast R-CNN [Ren+15] and SDP [YCL16], are chosen for further analyses.

4.2.2 Comparison of Feature Extractors

To scrutinize the impact of the feature extraction phase, two methods: CNN-based and the HOG algorithm, are compared, varying the feature dimensions. Table 4.3 presents the outcomes of different feature extraction approaches when coupled with the Fast R-CNN and SDP detectors. The results underscore that the use of CNN feature extraction yields superior identification performance compared to the HOG method. For instance, the IDF1 of CNN with 128 dimensions is 38.5%, representing an almost 5% increase over the IDF1 achieved by the HOG feature.

However, the traditional feature extractor demonstrates robustness in terms of MT and ML. For example, the MT of HOG with 128 dimensions is 8.4%, nearly 2% higher than the corresponding CNN feature. Notably, the disparity between CNN-based features and HOG-based features with 256 dimensions is not as pronounced as observed with 128 dimensions.

Under the assumption that human appearance features change gradually during motion, an additional uniform filter is incorporated to extract stable features from recent observations. This involves implementing a sliding window with a size of

Table 4.3: Different feature extractors on the MOT17 benchmark [Mil+16]

Extractor		Detector [a]		Size [b]		Evaluation Metrics (%) [c]						
CNN	HOG	F	S	128	256	IDF1 ↑	IDP ↑	IDR ↑	MOTA ↑	MOTP ↓	MT ↑	ML ↓
×		×		×		54.1%	76.1%	41.9%	49.2%	**13.3**%	18.7%	29.9%
	×	×		×		49.8%	68.7%	39.0%	49.1%	13.6%	21.8%	27.3%
×			×	×		**64.0**%	**78.6**%	**54.0**%	64.1%	15.3%	32.8%	23.4%
	×		×	×		58.5%	70.2%	50.2%	64.6%	15.7%	36.8%	**18.7**%
×		×			×	50.7%	70.0%	39.7%	49.1%	13.6%	21.8%	27.1%
	×	×			×	50.7%	70.1%	39.7%	49.3%	13.6%	21.4%	27.3%
×			×		×	59.3%	71.2%	50.9%	**64.8**%	15.6%	**37.4**%	19.6%
	×		×		×	58.9%	70.8%	50.4%	64.7%	15.6%	36.6%	19.4%

[a] F is the Fast-RCNN [Ren+15] and S is the SDP [YCL16], the CNN model are trained on the Market-1501 [Zhe+15] and Mars [Zhe+17] dataset.
[b] The size of output dimension.
[c] The best results comparing all modifications are in **bold**; The best results between feature extractors are underlined.

Table 4.4: Different feature extractors on the MOT17 benchmark [Mil+16]

Filter		Detector [a]		Size [b]		Evaluation Metrics (%) [c]						
w/o	w	F	S	64	128	IDF1 ↑	IDP ↑	IDR ↑	MOTA ↑	MOTP ↓	MT ↑	ML ↓
×		×		×		54.9%	76.9%	42.6%	49.3%	13.3%	19.4%	29.8%
	×	×		×		55.0%	77.2%	42.7%	49.3%	13.3%	19.6%	29.7%
×			×	×		54.1%	76.1%	41.9%	49.2%	13.3%	18.7%	29.8%
	×		×	×		54.7%	76.8%	42.4%	49.2%	13.3%	18.7%	29.8%
×		×			×	66.1%	80.8%	56.0%	64.4%	15.4%	**33.3**%	22.7%
	×	×			×	66.1%	80.7%	55.9%	64.4%	15.3%	32.6%	**22.2**%
×			×		×	64.0%	78.6%	54.0%	64.1%	15.3%	32.8%	23.4%
	×		×		×	**66.9**%	**81.9**%	**56.5**%	**64.3**%	15.3%	32.6%	23.4%

[a] F is the Fast-RCNN [Ren+15] and S is the SDP [YCL16], the CNN model are trained on the Market-1501 [Zhe+15] and Mars [Zhe+17] dataset.
[b] The size of output dimension.
[c] The best results comparing all modifications are in **bold**; The best results between feature extractors are underlined.

$t = 30$ in each tracker to capture the last 30 frames' features for the same target. The results are detailed in Table 4.4. The application of the uniform filter demonstrates a discernible improvement in terms of IDF1, IDP, and IDR scores. However, its impact is negligible on MOTA, MOTP, MT, and ML, as the tracking algorithm heavily relies on the Kalman filter.

The integration of the uniform filter contributes to a smoother feature representation, facilitating the preservation of continuous features for the same target. In other words, it mitigates ID switching when the target is tracked by the Kalman filter.

4.3 Action Recognition

As introduced in Section 3.2, we propose several adaptively attention mechanisms to update the dynamic relations and temporal convolutional layer to extract temporal

Table 4.5: The performance in terms of accuracy of HA-GCN with RA attention and RD attention layer![XB22a].

Methods	Accuracy		Influence ratio
	Joint stream	Bone stream	$r_{J/B}$
AGCN*	93.7%	93.2%	–
AGCN	93.9%	93.5%	0.67
AGCN (plus)	95.0%	94.7%	0.86
RD-GCN	95.6%	95.2%	0.95
RA-GCN	95.1%	95.4%	0.64
HA-GCN (single T)	95.2%	94.9%	0.94
HA-GCN (full)	**95.8**%	**95.5**%	0.88
HA-GCN (w/o RA)	86.6%	93.6%	4.84
HA-GCN (w/o RD)	94.6%	85.2%	0.12

* means using the original graph, and the rest experiments are conducted with new designed graph. AGCN is a single stream of 2s-AGCN. The AGCN “plus" implements an additional convolutional layer after generating the attention graph. The “single T” HA-GCN uses the temporal convolutional layer from AGCN instead of 4 parallel dilated convolutional layers. The model without RA (w/o RA) turns off the RA branch in the test phase, and its influence ratio (performance drop) is calculated in the respect to the full model, as the same as the model without RD branch(w/o RD).

features in the task of human action recognition. This section introduces the experimental results of human action recognition task on NTU-RGBD [Sha+16b] and Kinetics [Kay+17] datasets.

4.3.1 Ablation Study

We examine the contribution of proposed components to the recognition performance with the X-View benchmark on the NTU-RGB+D dataset.

Qualitative results: The baseline comprises a single stream of 2s-AGCN (AGCN), boasting accuracies of 93.7% and 93.2% for the joint and bone input streams, respectively. Through the incorporation of our newly designed graph, featuring connections between the head, hands, and feet, slight enhancements are observed 0.2% for the joint stream and 0.3% for the bone stream. Notably, the new graph exerts a more pronounced impact on the bone stream, as the introduced edges effectively augment the number of bones in the skeleton map.

Subsequent experiments build upon the new graph, revealing another noteworthy discovery: introducing the final graph mask into an additional convolutional kernel yields significant improvements in prediction results. The enhanced outcomes are detailed in the third row of Table 4.5, showcasing an increase of 1.3% and 1.5%, respectively, over the original AGCN.

As introduced in Section 3.2.1, our model incorporates two types of attention mechanisms: RD and RA. These attention layers are individually added to the attention-based graph convolution network. The performance of RD-GCN and RA-GCN on different input streams is presented in the fourth and fifth rows of Table 4.5. The results demonstrate that both RD and RA attention layers contribute positively to

Figure 4.3: Qualitative results of hybrid attention mask and final identity output of 10-th spatial graph attention layer for different actions. Both have size of 25 × 25, where 25 is number of skeleton joints [XB22a].

skeletal action recognition performance. The combination of both attention layers achieves the best overall performance, as indicated in the last row of Table 4.5. However, the distinct contributions of RA and RD attentions to different input streams remain unclear. This can be observed in the close influence ratio $r_{J/B}$, which is 0.95 for RD-GCN and 0.67 for RA-GCN.

To further investigate the individual impact of each attention branch, we conduct tests by deactivating one attention branch at a time during the test phase. The resulting top-1 accuracy on the X-View benchmark is presented in the last two rows of Table 4.5. Turning off the RA attention branch (w/o RA) leads to a more significant performance degradation in the joint stream compared to the bone stream (9.8% vs. 1.9%). Conversely, turning off the RD attention branch (w/o RD) results in a more substantial performance drop in the bone stream compared to the joint stream (2.2% vs. 10.3%). This outcome confirms that in the complete model, RD and RA attention mechanisms are more favorable for the bone stream and joint stream, respectively. Further evidence supporting this conclusion is available in the quantitative results.

In addition to comparing spatial convolutional layers, we extend our evaluation to include two variants of the temporal convolutional layer: a single convolutional layer and a multi-scale temporal layer. The single convolutional layer, as utilized in ST-GCN [YXL18] and 2s-AGCN [Shi+19b], employs a convolution kernel with a size of 9×1 to extract features from adjacent frames in the temporal domain. On the other hand, the multi-scale temporal layer employs four parallel dilated convolutional kernels, each with different dilation sizes, providing a broader receptive field in the temporal dimension.

The results of the single temporal convolutional layer are presented in the sixth row of Table 4.5 after HA-GCN (single T). In comparison to HA-GCN (single T), the full HA-GCN model utilizing the multi-scale temporal layer exhibits a substantial performance improvement for both input streams.

Qualitative results: The attention mask and final output of the 10th spatial hybrid attention layer are illustrated in Fig 4.3 for three action examples. In the scenario

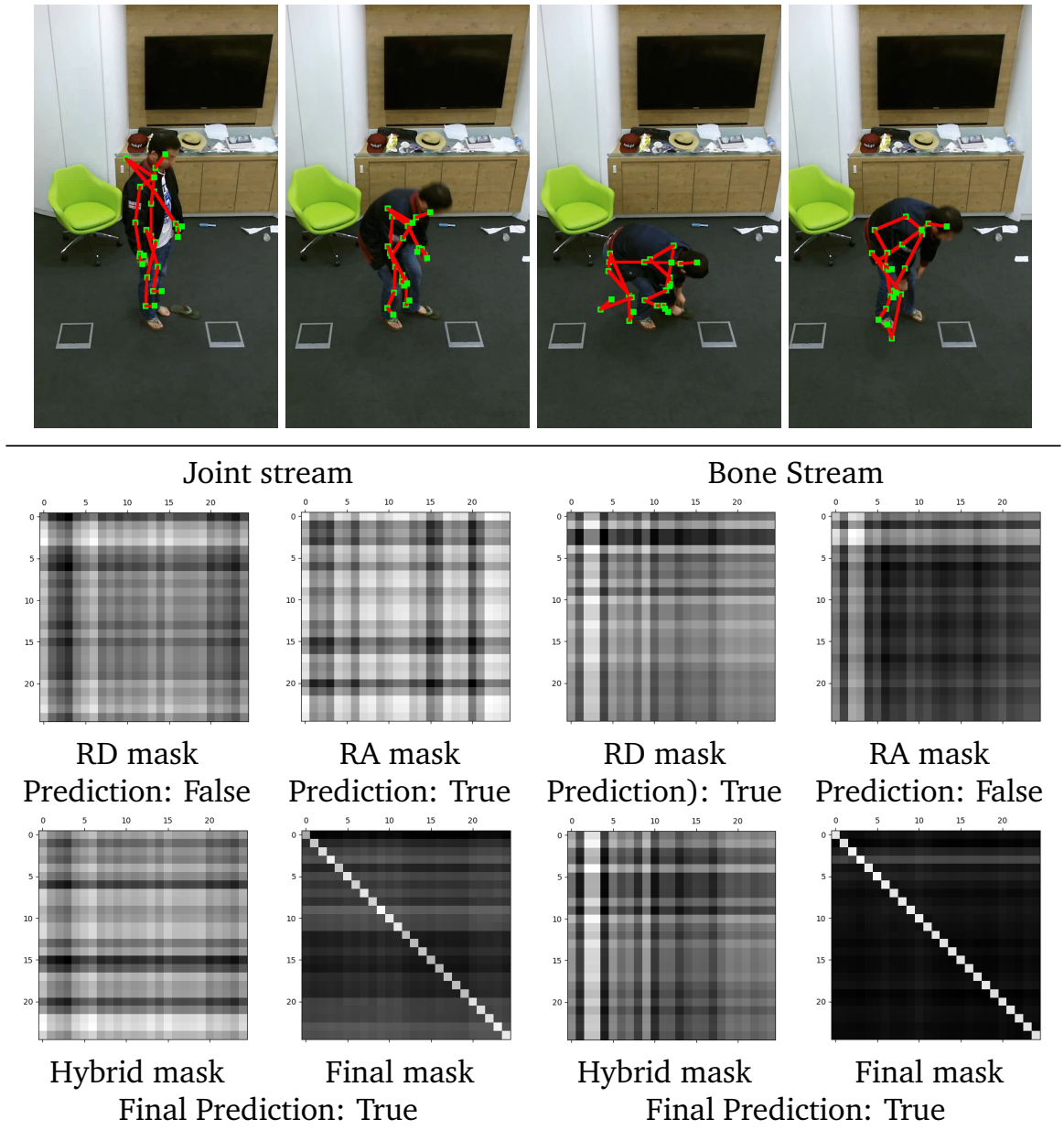


Figure 4.4: Qualitative outcomes of hybrid attention masks are illustrated in the context of the "putting on a shoe" example, utilizing joint and bone input streams [XB22a]. The top row showcases video images synchronized with the skeleton. In the middle row, both RD and RA attention masks for joint and bone input streams are presented alongside their corresponding action predictions. The final row exhibits the hybrid and ultimate attention mask, as well as the conclusive prediction.

of "watching time," where the subject raises their right forearm, the attention mask effectively highlights the right hand joints $(11, 23, 24)$. Conversely, during the "eating" action utilizing the left hand, the attention mask assigns greater importance to the joints $(7, 21, 22)$ corresponding to the left hand. In actions involving both hands, such as the "drinking" example depicted at the bottom of Fig 4.3, the attention mask appropriately emphasizes the positions of both arms.

The final identity output attention mask is presented on the right side of Fig 4.3, representing the sum of the attention mask and the identity adjacent matrix. This

Table 4.6: Comparisons of accuracy (%) with popular existing methods on the NTU-RGB+D [Sha+16b] cross-view and cross-subject dataset (Top-1) [XB22a]

Methods	X-View	X-Sub [2]
Lie Group [VAC14]	52.8	50.1
Deep-LSTM [Sha+16a]	67.3	60.7
VA-LSTM [Zha+17]	87.7	79.2
TCN [KR17]	83.1	74.3
Synthesized CNN [LLC17]	87.2	80.0
3scale ResNet 152 [Li+17]	90.9	84.6
ST-GCN [YXL18]	88.3	81.5
2s AS-GCN [Li+19]	94.2	86.8
2s AGCN [Shi+19b]	95.1	88.5
2s AGC-LSTM [Si+19]	95.0	89.2
4s Directed-GNN [Shi+19a]	96.1	89.9
4s Shift-GCN [Che+20]	96.5	90.7
4s CRT-GCN [Che+21]	96.8	**92.4**
PoseC3D[1] [Dua+21]	97.1	94.1
↳ Pose based [Dua+21]	93.7	-
1s HA-GCN (ours)	95.8	89.4
2s HA-GCN (ours)	96.6	91.5
4s HA-GCN (ours)	**97.0**	92.1

[1] The model used additional texture information, the pose based result is presented in the next line.
[2] X-View and X-Sub are the cross-view and cross-subject benchmark respectively.

combined output provides a comprehensive view of the attention-weighted features and their integration with the underlying structural information.

In Fig. 4.4, we present the hybrid attention masks and their predictions of the “Putting on a shoe" example. It is obvious that RA and RD focus on different characteristics. In the joint stream in the Fig.4.4, RD exhibits a pronounced correlation among nodes 1 to 6 (spine mid, neck, head, shoulder left, elbow left, wrist left). This correlation, resembling the action of "Taking off a shoe," results in a false prediction. In contrast, RA is more attentive to the entire body nodes, successfully rectifying the prediction by amalgamating the attention masks.

In the bone stream, RA concentrates solely on the influence of nodes 2 and 3, while RD encompasses the entirety of bone nodes and places emphasis on nodes 2 and 3. This nuanced focus ultimately corrects the prediction of the action class. These qualitative results further underscore the complementary nature of the two attention mechanisms.

4.3.2 Comparison with State-of-the-Art

In order to verify the performance of the attention based model, we conducted a performance comparison with established skeleton-based action recognition methods on both the NTU-RGB+D dataset and Kinetics dataset. The outcomes of these compar-

Table 4.7: Comparisons of accuracy (%) with popular existing methods on the Kinetics [Kay+17] dataset[1] (Top-1 and Top-5) [XB22a]

Methods	Top-1 (%)	Top-5 (%)
Deep-LSTM [Sha+16a]	16.4	35.3
TCN [KR17]	20.3	40.0
ST-GCN [YXL18]	30.7	52.8
2s AGCN [Shi+19b]	36.1	58.7
PoseC3D [Dua+21]	38.0	59.3
1s HA-GCN (ours)	35.1	58.0
2s HA-GCN (ours)	37.4	60.5
4s HA-GCN (ours)	**38.2**	**61.1**

[1] The pose data of Kinetics dataset is generated by OpenPose [Cao+19].

isons are detailed in Table 4.6. The methodologies under consideration include Lie Group [VAC14], Deep-LSTM [Sha+16a], VA-LSTM [Zha+17], TCN [KR17], Synthesized CNN [LLC17], 3scale ResNet 152 [Li+17], ST-GCN [YXL18], 2s AS-GCN [Li+19], 2s AGCN [Shi+19b], 2s AGC-LSTM [Si+19], 4s Directed-GNN [Shi+19a], 4s Shift-GCN [Che+20], CRT-GCN [Che+21] and PoseC3D [Dua+21]. 1*s* is only using joint data as the input. 2*s* means two streams that include joint and bone data. 4*s* is using four streams of input data, which are joint, bone, joint motion and bone motion, respectively.

Our model consistently demonstrates robust performance on both datasets, with the 4-stream model surpassing the previously leading pure skeleton-based approaches on the NTU RGBD X-View benchmark and Kinetics skeleton dataset. It is essential to note that PoseC3D incorporates not only skeletal information but also texture information, making it unfair to compare directly with other pure skeleton-based methods. Consequently, we present its performance in pose-based recognition separately in Table 4.6.

The main reason of worse performance of all models on Kinetics Skeleton dataset is the limitation of the dataset itself, which exclusively provides 2D skeletal information, in contrast to the 3D skeletal data available in the NTU-RGBD dataset, as indicated in Table 4.7. This discrepancy underscores the critical importance of incorporating multi-dimensional information for effective action recognition. Notably, the results, particularly in terms of the Top-5 metric, highlight the promising advancements achieved by our method in enhancing human action recognition, even in the face of the challenges posed by a dataset with limited skeletal dimensions.

4.4 Action Segmentation

As introduced in Section 3.3, we propose two novel action segmentation approaches. This section evaluate two introduced models (PGCN and TFGCN) in human action segmentation task on the Bimanual Actions [Kre+21] and IKEA Assembly datasets [Ben+20].

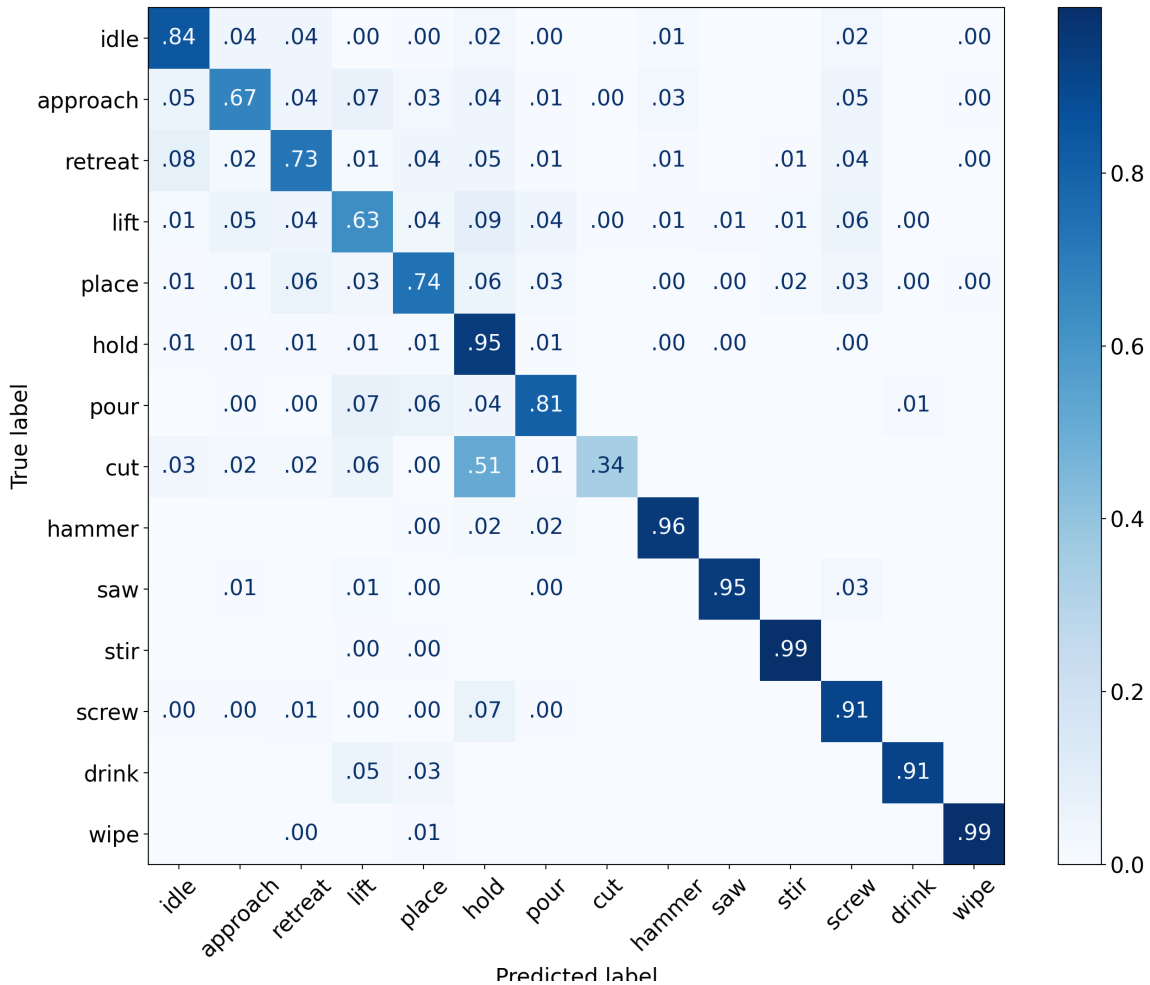


Figure 4.5: Normalized confusion matrix of PGCN [XB22b] for the top prediction of accumulative framewise classification correctness over all folds on Bimanual Actions dataset [DWA19].

4.4.1 Ablation Study

Pyramid Graph Convolutional Network

We investigate the impact of the proposed components on framewise Human-Object Interaction (HOI) recognition and segmentation by conducting experiments with the exclusion of subject 1 on the Bimanual Actions dataset [Kre+21]. The baseline for comparison is the single joint stream of 2s-AGCN [Shi+19b].

To optimize the performance of the attention unit, we assess its effectiveness across spatial, temporal, channel dimensions, as well as their various combinations. Additionally, we compare the proposed Temporal Pyramid Pooling module with the baseline FastFCN [Wu+19a].

The results of ablation studies are presented in Table 4.8, quantifying the performance of each configuration through F1 and F1@k scores. Notably, the F1@k scores in the right column of the table unmistakably demonstrate that the inclusion of the temporal pyramid pooling block yields improvements in relation segmentation across all specific settings. Consequently, we incorporate the temporal pyramid pooling block in the subsequent experiments involving combined attention layers.

Table 4.8: The F1 score of framewise prediction and F1@k score of action segmentation using original baseline model and models with different modifications in each unit [XB22b]

Encoder [a]			**Decoder** [b]		**Evaluation Metrics** (%) [c]				
S	T	C	TPP	Fast-FCN	F1 macro	F1 micro	F1@10	F1@25	F1@50
-	-	-	-	×	65.28	80.01	66.51	62.44	51.37
-	-	-	×	-	65.17	81.80	86.36	83.66	71.84
×	-	-	-	×	70.92	83.09	70.38	66.26	66.27
×	-	-	×	-	**81.50**	**86.92**	**88.38**	**85.06**	73.88
-	×	-	-	×	75.93	83.42	67.83	63.51	52.68
-	×	-	×	-	77.26	84.94	78.77	75.46	61.43
-	-	×	-	×	70.16	83.56	74.16	70.55	58.92
-	-	×	×	-	67.39	82.50	88.23	84.86	**74.10**
×	×	-	×	-	80.29	85.25	84.38	81.46	68.55
×	-	×	×	-	69.65	80.57	84.09	81.07	66.99
-	×	×	×	-	71.42	82.75	85.36	81.45	69.21
×	×	×	×	-	72.39	83.63	85.68	81.94	70.86

[a] We compare the performance of attention layer in the encoder setup on different dimensions, namely spatial (S), temporal (T) and channel (C).
[b] The decoder is the common Fast-FCN [Wu+19a] when there is no temporal pyramid pooling block.
[c] The best results comparing all modifications are in **bold**; The best results between TPP and Fast-FCN in the decoder setup are underlined

Analyzing the F1 scores in the middle column, it becomes evident that all proposed components contribute to enhancing the baseline model AGCN [Shi+19b] in framewise recognition. Furthermore, since the spatial attention unit extracts the basic features representing spatial distribution and relations of nodes per frame, model with spatial attention unit demonstrates superior performance across all model configurations.

The temporal attention unit focuses on extracting temporal relations between consecutive frames, proving beneficial for action recognition involving known segments rather than action segmentation itself. Meanwhile, the channel attention layer emphasizes the importance of distinguishing between channels, facilitating the classification of an entire clip of a single action rather than segmentation. Notably, the performance of combined models is adversely affected by suboptimal attention layers, specifically the temporal and channel attention layers.

In addition to F1 scores, we evaluate the Top-1 accuracy of the proposed model on Bimanual Actions dataset [Kre+21]. Fig. 4.5 depicts the normalized confusion matrices for the top prediction. A significant source of confusion for the classifier is predicting *hold* when the actual action is *cut*. This misprediction arises from using *wrist* joints to represent *hands*, which exhibit a limited range of motion and can be mistakenly associated with *holding* a *knife*. Consequently, the action *cut*, characterized by a larger range of motion, is seldom misidentified as *hold*. Additionally, confusion occurs between actions such as *approach*, *retreat*, *lift*, and *place*, primarily due to unstable object bounding boxes stemming from the object detection method. Notably, *approach* and *retreat* actions, often executed rapidly (sometimes within 5 frames), contribute to confusion. An illustrative example is provided in the qualita-

tive results. Addressing these challenges would require stable object detection and pose estimation methods, although this aspect is not the primary focus of our work and remains unexplored.

Table 4.9: Ablation study of duration of samples and the sampling gap on the TUM HOI dataset. Each duration is combined with three different sampling gaps *.

Duration	Gap	Top-1 acc (%)	F1 macro (%)	F1@10 (%)	F1@25 (%)	F1@50 (%)
	5	77.29	77.58	88.28	84.25	72.26
30	10	77.11	76.70	87.52	83.67	71.78
	15	73.36	74.02	85.25	79.79	67.90
	5	75.46	75.61	85.39	82.01	68.13
60	10	**78.00**	**77.84**	88.49	**85.82**	**74.53**
	15	75.22	75.29	86.86	83.20	71.60
	5	76.55	76.55	86.92	82.56	71.22
90	10	77.20	76.86	86.68	83.57	72.65
	15	76.36	76.43	86.54	82.04	70.20
	5	77.07	76.95	88.00	84.15	73.65
120	10	76.67	76.51	86.29	83.08	71.60
	15	74.70	75.55	87.87	82.46	69.40
	5	76.22	76.71	84.89	80.75	70.13
150	10	73.89	74.12	85.19	79.92	66.92
	15	75.82	75.77	87.45	82.40	71.12
	5	75.16	75.77	83.20	79.49	66.49
180	10	75.23	76.10	86.30	80.44	69.49
	15	76.33	76.72	86.30	82.45	69.04

* The employed network architecture is AGCN+FastFCN with modified graph connection. The best performance of each metric is in **blod**.

The choice of sample duration and sampling gap can significantly influence model performance. A shorter sample duration and smaller sampling gap can provide more training data for the model, which potentially benefit the model. However, a shorter sample duration implies that each training sample contains less temporal information, while a shorter sampling gap results in higher similarity between training samples, both of which could adversely affect the model's performance. To determine the optimal combination of sample duration and sampling gap, experiments with varying setups are conducted on the TUM HOI dataset. The sample duration ranges from 30 frames to 180 frames, with the sampling gap varying between 5 and 15 frames. The results of these experiments are presented in Table 4.9.

As indicated in Table 4.9, experimental results demonstrate that a sample duration of 60 frames outperforms all other durations across all metrics on the TUM HOI dataset. Specifically, with a sampling gap of 10 frames, it achieves the highest performance in top-1 accuracy at 78.00% and F1 macro at 77.84%.

As observed, when the sampling gap is fixed, the F1@k performance for longer sample durations (150 and 180 frames) is noticeably reduced compared to shorter durations (30, 60, and 90 frames). This discrepancy may be attributed to the fact that shorter sample durations typically involve fewer action switches, leading the model to segment actions less frequently and thereby suppressing over-segmentation.

Interestingly, for sample durations less than 120 frames, the poorest results consistently occur with a sampling gap of 15 frames. However, with increasing sample

duration, the critical sampling gap diminishes (e.g., 10 frames for 150 frames and 5 frames for 180 frames). This suggests that the similarity between training samples is not a dominant factor when the sample length is short. For instance, with a sample duration of 30 frames and a gap of 5 frames, the similarity between two adjacent samples is 83.33%. Conversely, as the sample length increases, smaller sampling gaps result in significantly higher similarity (e.g., 97.22% for a duration of 180 frames and a gap of 5 frames), potentially increasing the risk of overfitting due to overly similar samples.

This work also investigates the influence of different kinds of loss functions. A cross-entropy loss $\mathcal{L}_{ce}$ for multi-class classification, a temporal smoothing loss $\mathcal{L}_{ts}$ for reducing over-segmentation errors, and a boundary alignment loss $\mathcal{L}_{ba}$ for addressing the shift-segmentation errors serve as candidates. Different combinations of these loss functions are compared, considering the multiple hyper-parameters associated with temporal smoothing loss ($\mathcal{L}ts$) (i.e., λ and τ) and boundary alignment loss ($\mathcal{L}ba$) (i.e., λ, κ, and γ). To streamline the experiment, an optimal combination of hyper-parameters is selected, specifically ($\lambda = 0.15$ and $\tau = 4$).

As shown in Table 5.5, hyper-parameters α, κ and γ have different impact on the model performance. The best result is achieved by the combination $\gamma = 0.1$, $\kappa = 13$ and $\alpha = 1$. Experimentally, the increase of α reduces the performance (both accuracy and F1 scores) significantly, whereas the value of γ has no significant impact on performance (except the F1@10). Besides, the top-1 accuracy increases gradually as the kernel size increases. Considering that adding the temporal smoothing loss $\mathcal{L}_{ts}$ may have a different impact on the results, removing the combinations with significantly lower accuracy in Table 5.5, this thesis combines the remaining combinations with $\mathcal{L}_{ts}$ and $\mathcal{L}_{ce}$ to find the optimal loss function. As observed in Table 5.6, the increase of γ can not produce favorable results, and although it slightly improves performance on the F1@10 and F1@25 compared to $\mathcal{L}_{ce}$ alone, the results on top-1 accuracy and on F1 macro are notably lower. Also, without $\mathcal{L}_{ba}$, the combination of classification loss and temporal smoothing loss is not sufficient, only a marginal improvement on F1@10, but leads to performance decreases across all the other metrics compared to the classification loss alone. When the kernel size is between 9 and 11, compared to the original loss, all can obtain a noticeable boost on F1@k but accompanied by a tiny drop on top-1 accuracy and on F1 macro. Here, this thesis chooses kernel size 9 for the boundary alignment loss since it has almost no reduction in top-1 accuracy (merely 0.06%). The final loss function is $\mathcal{L}_{ce} + 0.15\mathcal{L}_{ts} + 0.1\mathcal{L}_{ba}$.

Temporal Fusion Graph Convolutional Network

To assess the contribution of each module within the proposed decoder, we conducted an ablation study on the Bimanual Actions dataset [DWA19]. Various configurations were individually integrated into the decoder, encompassing variations with or without *global feature extraction*, *temporal feature fusion*, and the *classifier* modules. The results of the ablation study are presented in Table 4.10.

All three proposed modules significantly contribute to the performance of action recognition and segmentation. Notably, the *classifier* exerts the most substantial influence on performance, acting as the final layer crucial for mapping feature maps to predictions. Additionally, *global feature extraction*, with its wide field of view, leads to notable improvements in both action recognition and segmentation. How-

Table 4.10: Ablation study of introduced modules in the decoder on the Bimanual Actions dataset [DWA19].

GFE[a]		TFF		Cl		Evaluation Metrics - Bimanual Actions dataset[b] (%)				
w/o	w	w/o	w	w/o	w	Top 1 ↑	F1 macro ↑	F1@10 ↑	F1@25 ↑	F1@50 ↑
×		×		×		78.69	79.23	–	–	–
	×	×		×		79.85	80.28	89.73	88.11	79.36
×			×	×		80.15	80.08	88.66	86.23	77.29
	×		×	×		81.80	82.56	90.49	87.99	78.92
×		×			×	82.97	83.73	88.52	86.63	77.23
	×	×			×	86.11	86.61	90.48	89.09	80.43
×			×		×	84.21	85.04	87.83	85.48	76.66
	×		×		×	**89.06**	**89.24**	**93.82**	**92.27**	**85.34**

[a] The configurations are denoted as: “GFE" = global feature extraction; “TFF" = temporal feature fusion; “CL" = classifier; “w" = with; “w/o" = without. In setups without the temporal feature fusion module, the *interpolate* function is employed for upsampling. In configurations lacking a *classifier*, the spatial dimension is eliminated through averaging.
[b] The experiments are conducted on the subject 1 testset of the Bimanual Actions dataset [DWA19]. The best results across all configurations are in **bold**; The best results for each setup with or without the Classifier are underlined.

ever, relying solely on temporal feature fusion does not contribute significantly to enhancement; in the absence of global features, it primarily upsamples the condensed features of the encoder to the original time scale. The synergy of these two modules achieves optimal performance across configurations, with or without the classifier.

To determine the optimal encoder-decoder combination, we explore various popular existing Graph Convolutional Networks (GCNs), including ST-GCN [YXL18], AGCN [Shi+19b], CTR-GCN [Che+21], HA-GCN [XB22a], and PGCN [XB22b] as encoders. These encoders are combined with Fast-FCN [Wu+19a], TPP [XB22b], and the proposed Temporal Fusion (TF) decoders. Table 4.11 illustrates that our TF decoder outperforms the other two decoders across various configurations based on all evaluation metrics. Notably, the configuration of CTR-GCN [Che+21] combined with TF achieves the highest performance in terms of accuracy, F1 macro, and F1@k.

However, it is clear that the novel decoder significantly increases the number of parameters, resulting in heightened computational demands and increased hardware costs during application. From this perspective, the CTR-GCN [Che+21] encoder exhibits promising performance, having the fewest parameters among all encoders. Consequently, the setup combining CTR-GCN [Che+21] as the encoder and TF as the decoder forms the backbone of our Temporal Fusion Graph Convolutional Network (TFGCN).

To analyze the influence of the residual connections on the feature space distance preserving and to evaluate the performance of the proposed Spectral Normalized Residual (SN-res) connection, we execute the experiments concerning three setups, namely without residual connections, with normal residual connections and with Spectral Normalized Residual connections. Table 4.12 summarizes the ablation studies, including Gaussian process (GP) configurations. By comparing the metrics produced by the Gaussian process and the *softmax*, it is clear that Gaussian process facilitates significantly less multi-class calibration errors, namely TACE, ACE and SCE. In contrast, the configurations of *softmax* have less Expected Calibration Error (ECE), because the ECE heavily relies on the overconfident probability of the predicted class from *softmax*.

The results in Table 4.12 confirm that the residual connection is of high significance for accurate predictions and accuracy-related performance, such as F1 macro

Table 4.11: Comparison of encoder-decoder setups on the Bimanual Actions dataset [Kre+21][a].

Encoder[b]	Decoder	Top 1	F1 micro	F1@10	F1@25	F1@50	#
	Fast-FCN	78.97	82.05	71.58	68.34	58.13	4.5*M*
ST-GCN	*TPP*	78.88	82.18	83.42	80.01	69.40	5.1 M
	TF	81.67	84.20	85.53	82.61	71.57	21.8*M*
	Fast-FCN	79.03	81.98	73.70	69.99	59.89	4.9*M*
AGCN	*TPP*	84.13	85.48	85.93	83.61	74.56	5.4*M*
	TF	84.66	87.10	88.51	85.79	75.92	22.2*M*
	Fast-FCN	79.52	81.94	64.97	61.83	52.35	2.8*M*
CTR-GCN	*TPP*	84.70	87.33	86.03	84.02	75.36	3.4*M*
	TF	**89.06**	**89.24**	**93.82**	**92.27**	**85.34**	20.1*M*
	Fast-FCN	85.43	87.30	80.91	78.61	70.59	2.8*M*
HA-GCN	*TPP*	84.81	87.08	81.47	78.83	71.08	3.4*M*
	TF	85.46	87.64	91.08	88.75	77.59	20.1*M*
	Fast-FCN	80.13	82.65	70.06	66.37	56.37	4.9*M*
PGCN	*TPP*	84.86	86.75	88.58	85.82	76.40	5.4*M*
	TF	86.44	88.55	89.58	86.94	76.96	20.1*M*

[a] The experiments are conducted on the subject 1 testset of the Bimanual Actions dataset [Kre+21]. The best results cross all setups are in **bold**. The best results of decoder setup are underlined. # is number of parameters.
[b] The compared encoders: STGCN [YXL18], AGCN [Shi+19b], CTR-GCN [Che+21], HA-GCN [XB22a], PGCN [XB22b]. The decoders include: Fast-FCN [Wu+19a], TPP [XB22b], and the proposed TF.

Table 4.12: Ablation study of the Spectral Normalized Residual connection and the Gaussian process in terms of action segmentation on the Bimanual Actions dataset [Kre+21].

Res[a]			**GP**		**Evaluation Metrics - Bimanual Actions dataset** (%)				
w/o	w	SN	w/o	w	Top 1 ↑	F1 macro ↑	F1@10 ↑	F1@25 ↑	F1@50 ↑
×			×		73.69 ± 1.81	74.24 ± 1.79	86.02 ± 1.01	82.51 ± 1.20	71.26 ± 1.58
×				×	75.46 ± 1.82	76.78 ± 1.79	86.17 ± 1.01	82.61 ± 1.19	69.67 ± 1.57
	×		×		89.34 ± 0.57	89.52 ± 0.59	93.29 ± 0.67	92.17 ± 0.75	85.08 ± 1.17
	×			×	**89.34** ± 0.57	**89.54** ± 0.59	93.30 ± 0.64	92.17 ± 0.71	**85.10** ± 1.14
		×	×		88.14 ± 0.30	88.66 ± 0.29	92.73 ± 0.29	91.72 ± 0.36	84.51 ± 0.53
		×		×	88.44 ± 0.29	88.89 ± 0.29	**93.31** ± 0.29	**92.18** ± 0.33	84.76 ± 0.53

[a] The configurations are denoted as: “Res" = Residual connections; “SN" = spectral normalized; “w" = with; “w/o" = without; “GP" = Gaussian process kernel. The configuration without Gaussian process kernel is using *softmax* to output prediction.

and F1@k. Moreover, the results of standard deviation demonstrate that the residual connections increase the model stability.

The normalized confusion matrices of for the top prediction from the proposed TFGCN is presented in Fig 4.6. The primary challenge lies in accurately predicting actions such as *approach*, *lift*, *retreat*, and *place*, particularly in scenarios involving interactions with diverse objects. This challenge is further compounded by the use of unstable object detection methods. Additionally, the actions of *approach* and *retreat* often occur instantaneously, sometimes within a span of just 5 frames. Another noteworthy observation is the misclassification of "hold" when the actual action is "cut". This misclassification arises from using "wrist" joints to represent "hands," resulting in a limited resolution of motion detection and a potential misunderstanding of the action as "hold." These challenges could be alleviated through the implementation of stable object detection and pose estimation methods.

However, our primary focus is not on developing a flawless algorithm for object

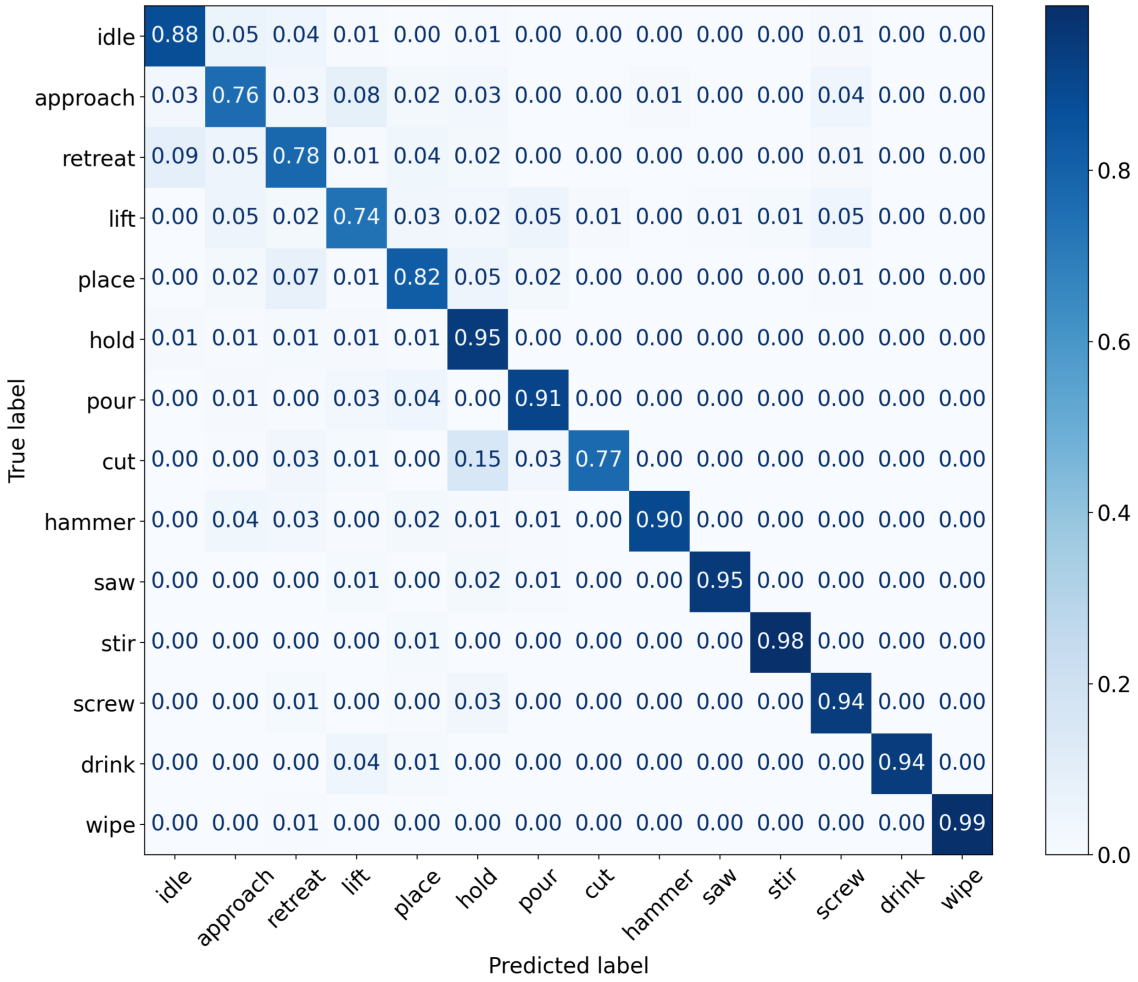


Figure 4.6: Normalized confusion matrix of TFGCN for framewise prediction of accumulative classification correctness over the subject 1 testset on the Bimanual Actions dataset [Kre+21]

detection and pose estimation. Instead, our emphasis is on the feature map's ability to effectively represent input noise (input distance) and how to quantify prediction uncertainty.

4.4.2 Comparison with the State-of-the-Art

Pyramid Graph Convolutional Network

The proposed PGCN model is compared with the state-of-the-art action recognition and segmentation methods on Bimanual Actions [Kre+21] and IKEA Assembly datasets [Ben+20]. The methods used for comparison include the model proposed by Dreher et al. [DWA19], Independent BiRNN, Relational BiRNN, ASSIGN [Mor+21] and several popular graph convolutional networks: ST-GCN [YXL18], AGCN [Shi+19b] and CTR-GCN [Che+21] combining with two decoders, namely FastFCN [Wu+19a] and the proposed temporal pyramid pooling (TPP) module.

The performance metrics, specifically the F1 score and F1@k, on the Bimanual

Table 4.13: Comparison of framewise action recognition of PGCN [XB22b] with state-of-the-art methods on Bimanual Actions dataset [Kre+21]

Model	F1 macro (%)	F1 micro (%)
Dreher et al. [DWA19]	63.0	64.0
AGCN+FastFCN	65.3	80.0
AGCN+TPP	65.2	81.8
ST-GCN+FastFCN	68.7	82.5
ST-GCN+TPP	69.3	82.7
CTR-GCN+FastFCN	71.1	82.3
CTR-GCN+TPP	72.0	82.9
Independent BiRNN [Mor+21]	74.8	76.7
Relational BiRNN [Mor+21]	77.5	80.3
ASSIGN [Mor+21]	79.8	82.6
PGCN (Ours)	**81.5**	**86.9**

Table 4.14: Cross validation results of action segmentation in comparison with state-of-the-art methods of PGCN [XB22b] on Bimanual Actions dataset [Kre+21]

Model	F1@10 (%)	F1@25 (%)	F1@50 (%)
Dreher et al. [DWA19]	40.6 ± 7.2	34.8 ± 7.1	22.2 ± 5.7
Independent BiRNN [Mor+21]	74.8 ± 7.0	72.0 ± 7.0	61.8 ± 7.3
CTR-GCN+FastFCN	74.9 ± 8.1	72.2 ± 8.7	66.6 ± 11.4
Relational BiRNN [Mor+21]	77.7 ± 3.9	75.0 ± 4.2	64.8 ± 5.3
ASSIGN [Mor+21]	84.0 ± 2.0	$81.2 \pm \mathbf{2.0}$	$68.5 \pm \mathbf{3.3}$
CTR-GCN+TPP	84.8 ± 3.2	82.1 ± 4.0	73.5 ± 5.6
PGCN (Ours)	$\mathbf{88.5 \pm 1.1}$	$\mathbf{85.5 \pm 2.0}$	$\mathbf{77.0} \pm 3.4$

Actions dataset [Kre+21] are detailed in Table 4.13 and Table 4.14, respectively. The Progressive Graph Convolutional Network (PGCN) surpasses both the state-of-the-art models and baseline approaches across all configurations of the F1 and F1@k measures. For instance, the F1 macro and micro scores exhibit improvements of 1.7% and 4.3%, respectively.

Additionally, it is noteworthy that the proposed Temporal Pyramid Pooling Block significantly enhances F1@k scores, showcasing improvements of 4.5%, 4.3%, and 8.5% when compared to the ASSIGN method. This underscores the efficiency of the temporal pyramid pooling block in action segmentation. The ASSIGN method employs a Bi-directional Gated Recurrent Unit to amalgamate information from consecutive frames, but this approach has limitations in extracting temporal information, resulting in shift-segmentation. On the other hand, alternative methods using separate segmentation labels lack temporal information, leading to instances of over-segmentation. Further supporting evidence for these observations is evident in the qualitative results.

Table 4.15 presents the top-1 accuracy, micro-recall and F1@k score on IKEA Assembly dataset [Ben+20]. In terms of top-1, and all three F1@k scores, PGCN

Table 4.15: Framewise recognition and segmentation results in terms of top-1 accuracy, macro-recall, and F1@k of PGCN [XB22b] on IKEA Assembly dataset [Ben+20]

Model	top 1	macro	F1@ (%)		
			10	25	50
HCN [Li+18]	39.15	28.18	-	-	-
ST-GCN [YXL18]	43.40	26.54	-	-	-
multiview+HCN [Ben+20]	64.25	**46.33**	-	-	-
ST-GCN+TPP	68.92	25.63	66.92	59.66	41.33
AGCN+TPP	70.53	27.79	76.32	69.85	52.14
CTR-GCN+TPP	78.70	37.98	78.84	72.68	54.40
PGCN (Ours)	**79.35**	38.29	**81.53**	**76.28**	**58.07**

outperforms all other popular methods. This reinforces the notion that our approach, incorporating spatial attention and temporal pyramid pooling modules, provides a more competitive framework for recognizing and segmenting actions on a per-frame basis. It's important to note that the macro-recall for all methods is compromised due to the uneven distribution of the dataset [Ben+20].

Temporal Fusion Graph Convolutional Network

To demonstrate the efficiency and robustness, we compare the performance of the proposed TFGCN with other popular existing methods on two challenging dataset in the field of Human-Object-Interaction recognition and segmentation, i.e., BimActs [Kre+21] and IKEA Assembly [Ben+20] datasets.

First, the quantitative experiment of action recognition and segmentation are conducted on both the BimActs [Kre+21] and IKEA Assembly [Ben+20] datasets, and the corresponding results are meticulously listed in Table 4.16 and Table 4.17, respectively.

From Table 4.16, it is easy to see that our model achieves the best performance cross all of the metrics on the Bimanual Actions dataset [Kre+21]. Especially, the proposed Temporal Fused Graph Convolutional Network (TFGCN) improves significantly the performance in terms of average F1@k score (by 6.2%, 6.4% and 8.4%) compared to the PGCN, which confirms its efficiency for action recognition and segmentation. In addition, the F1@k standard deviation of our results is also the smallest, which indicates that the model is also robust.

The experimental results on Table 4.17 demonstrates that the proposed TFGCN outperforms the-state-of-art methods in terms of Top1 recognition accuracy and f1@k segmentation score. While the PIFL [Yan+23] demonstrates superior performance on the IKEA Assembly dataset, it is important to note that direct comparisons with other methods may be unfair. This disparity stems from the fact that PIFL extracts appearance features through an I3D model, rather than leveraging the provided object detection results from the dataset. Nevertheless, this underscores the importance of comprehensive instance information in the context of action determination. Note that the multiview HCN [Ben+20] achieves the best performance in terms of macro-recall, since it utilizes several images from different view in the same time.

Table 4.16: Comparison of cross-validation results of TFGCN with state-of-the-art methods on the Bimanual Actions dataset [Kre+21][a]

Model	Accuracy[a] (%)	F1 macro (%)	F1@ (%)		
			10	25	50
Dreher et al. [Kre+21]	63.0	64.0	40.6 ± 7.2	34.8 ± 7.1	22.2 ± 5.7
H2O+RGCN [Lag+23]	68.0	66.0	–	–	–
Independent BiRNN [Mor+21]	74.8	76.7	74.8 ± 7.0	72.0 ± 7.0	61.8 ± 7.3
Relational BiRNN [Mor+21]	77.5	80.3	77.7 ± 3.9	75.0 ± 4.2	64.8 ± 5.3
ASSIGN [Mor+21]	82.3	78.8	84.0 ± 2.0	81.2 ± 2.0	68.5 ± 3.3
2G-GCN [Qia+22]	–	–	85.0 ± 2.2	82.0 ± 2.6	69.2 ± 3.1
PGCN[XB22b]	86.8	83.9	88.5 ± 1.1	85.5 ± 2.0	77.0 ± 3.4
TFGCN (Ours)	**89.4**	**89.6**	**94.3** ± 1.2	**92.2** ± 1.6	**86.1** ± 2.9

[a] The models are cross validated on the leave-one-subject-out benchmark, the best results of each class are in **bold**. The F1 micro and macro results are averaged, F1@k are listed with mean and standard deviation.

4.4.3 Qualitative Results

Figure 4.7: Comparison of the qualitative results of PGCN [XB22b] on Bimanual Actions dataset [Kre+21] for a *sawing* example

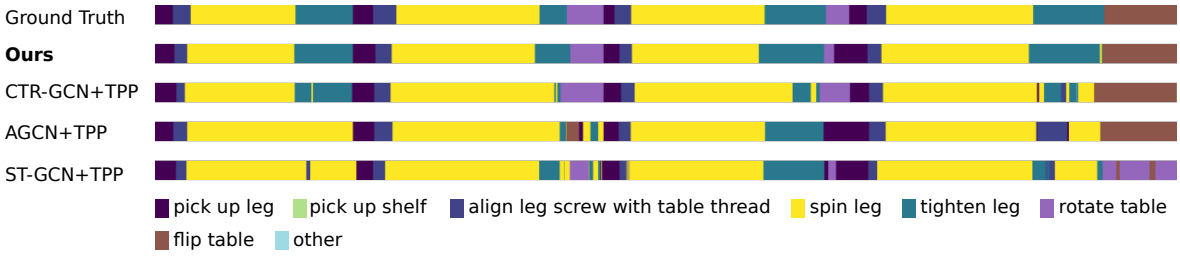


Figure 4.8: Comparison of the qualitative results of PGCN [XB22b] on IKEA Assembly dataset [Ben+20] for an *assembly side table* example

We present the detail outputs of PGCN model and related methods using examples from Bimanual Actions [DWA19] and IKEA Assembly dataset [Ben+20]. Fig. 4.7 shows a straightforward example of *sawing* in Bimanual Actions [DWA19], where both PGCN and ASSIGN have a stronger ability to prevent over-segmentation compared to Relational BiRNN. Our PGCN achieves more precise segmentation than ASSIGN, which even recognizes correctly the frame index between *Approach* and *Hold* (0 frame error) in the given example. As aforementioned, the potential for false predictions is evident, especially towards the end of both the ground-truth and predictions, where even slight movements can lead to inaccuracies.

Table 4.17: Action recognition and segmentation results of TFGCN in terms of top-1 accuracy, macro-recall, and F1@k on the IKEA Assembly dataset [Ben+20]

Model	Accuracy (%)	Macro-recall (%)	F1@ (%)		
			10	25	50
HCN [Li+18]	39.15	28.18	-	-	-
ST-GCN [YXL18]	43.40	26.54	-	-	-
multiview+HCN [Ben+20]	64.25	46.33	-	-	-
ST-GCN+TPP [XB22b]	68.92	25.63	66.92	59.66	41.33
AGCN+TPP [XB22b]	70.53	27.79	76.32	69.85	52.14
MGAF [KJH21]	72.40	49.10	—	—	—
CTR-GCN+TPP [XB22b]	78.70	37.98	78.84	72.68	54.40
PIFL* [Yan+23]	**84.60**	**62.00**	—	—	—
PGCN [XB22b]	79.35	38.29	81.53	76.28	58.07
TFGCN (Ours)	80.39	39.77	**83.99**	**80.04**	**68.00**

* The model extracts appearance features by an I3D [CZ17] model instead of using the provided object detection results from the dataset.

Besides the simple example, Fig. 4.8 presents a segmentation example of the intricate *assembly side table* task within the IKEA Assembly dataset [Ben+20]. Here, we compare the qualitative performance of methods employing the same decoder and different encoders. It can be seen that our PGCN model prevents under-segmentation better than other models with the same decoder, which further demonstrates the effectiveness of our spatial attention unit. Across tasks ranging from simple to complex, our model consistently demonstrates robust and high-performance results.

4.5 Event Detection

In this section, I present the experimental results of the proposed event detection algorithm (Section 3.4) on the NTU RGB+D dataset [Sha+16b]. First, it introduces the dataset for training and evaluation. Second, it displays the tendency of weight parameter with an increasing number of iteration in the training phase and demonstrates how the dimension of action units influences the prediction performance. In the end, we compare our method with other existing well-performed dictionary learning methods on the NTU RGB+D dataset [Sha+16b].

4.5.1 Validating the Effectiveness of GODL

To substantiate the resilience of GODL to outliers, we monitored the changes in the weights of six skeletons (w) throughout the training of the first action unit, as illustrated in Fig 4.9 (a). The plot reveals the weight dynamics of these skeletons during the initial sub-sequence "*standing*" across iterations in the GODL program.

Notably, the weights of outliers (skeletons 5 and 6) exhibit a more pronounced decreasing trend, while those of inliers (skeletons 1 to 4) change at a slower pace.

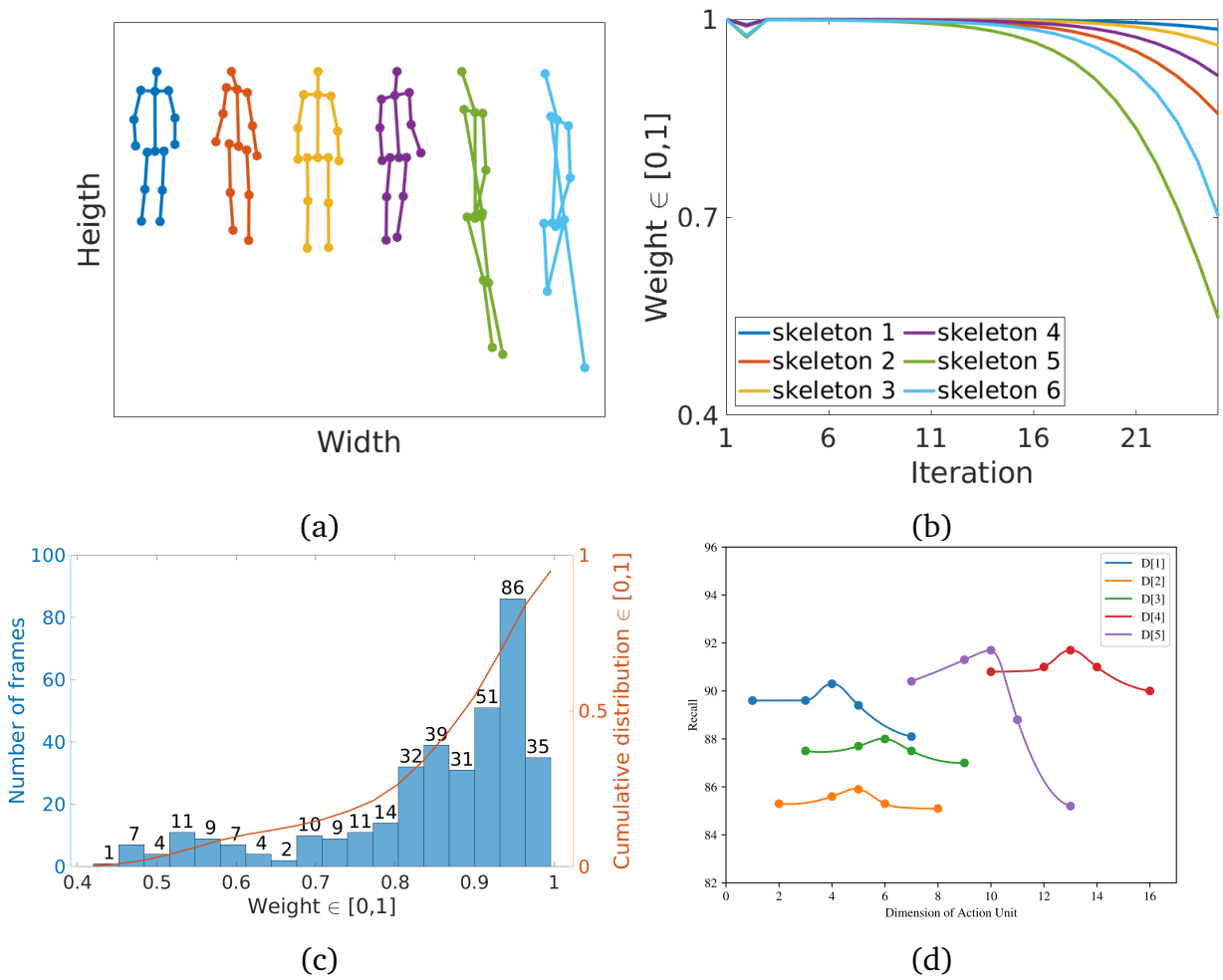


Figure 4.9: Examples from the work [Xin+21]. (a) Example of six skeletons, two of which are deformed (two on the right). (b) The weight tendency of six skeletons in the first action unit "standing" over iteration (c) the histogram of weight value and its cumulative distribution at the last iteration. (d) The recall with different action unit dimension.

By the conclusion of the iteration, outliers are assigned values of 0.547 and 0.717 respectively, whereas the inliers maintain higher values, specifically 0.9847, 0.8564, 0.9604, and 0.9142 for skeletons 1 to 4.

Fig 4.9 (b) presents the histogram of weights and their cumulative distribution at the final iteration. Notably, a swift 90% of skeletons possess weights exceeding 0.6, signifying a substantial influence on the cost function. Consequently, these skeletons are categorized as inliers, while the remaining 10% with lower values are identified as outliers.

Given that the dimension of each action unit significantly impacts prediction performance, we evaluate recall performance across 6 different settings to identify the optimal dimension for each action unit. The dimensionality of the dictionary is closely tied to the complexity of the action unit. For instance, the first three dictionaries ($\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$) have fewer dimensions than the last two dictionaries ($\mathbf{D}_4$ and $\mathbf{D}_5$), reflecting the simplicity of action units such as "*standing*", "*bending knee*", and "*opening arm*" compared to "*knee landing*" and "*arm supporting*".

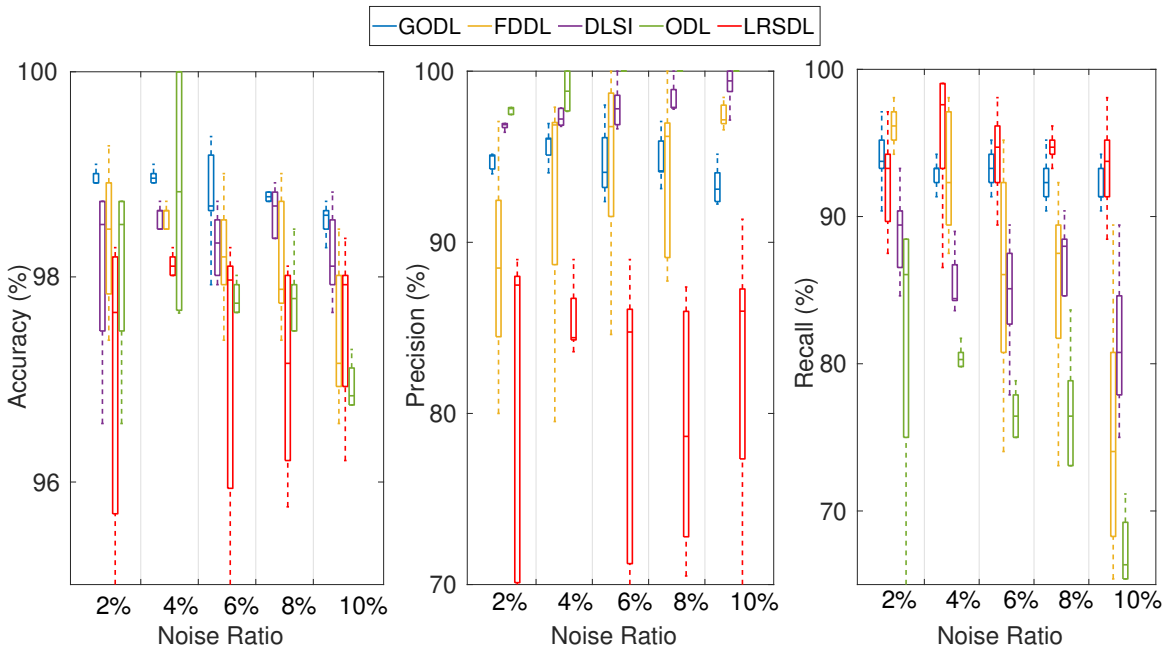


Figure 4.10: The comparison of robustness under different noise level [Xin+21].

The recall initially increases with dimensionality before reaching the optimal point, as the initial dimensions may not be sufficient to adequately represent the action space. Beyond the optimal point, the recall diminishes with increasing dimensionality due to overfitting. Fig 4.9 (c) visually depicts the dimension selection process for each action unit. The optimal combination of dimensions identified is $4, 5, 6, 10, 13$.

4.5.2 Evaluation of Fall-Down using Action Unit and Temporal Structure

The evaluation system comprises two distinct components: the action unit model and the temporal model. In the action unit model, the error (e) between the 3D skeleton frames $\bar{y}$ and the action unit model is considered within a confidence interval. If the error falls within this range, the corresponding action unit is deemed successful, and the evaluation proceeds to the next action unit. This process repeats until all action units have been evaluated.

In the temporal model, the height change of the incoming skeleton must satisfy two conditions. Only when both the action unit and temporal conditions are met, is the action classified as a true fall.

The evaluation results are available in Table 4.18. In comparison to the other four state-of-the-art Dictionary Learning methods, our GODL model attains the highest accuracy and precision. While both FDDL [Yan+11] and LRSDL [VM16] demonstrate commendable recall performance, their precision lags behind by 8% $\sim$ 10% compared to our method. Furthermore, in contrast to the baseline ODL [Mai+10], our approach exhibits superior performance across all evaluated aspects.

To underscore the robustness of our method, we intentionally introduce noise into the training data and subsequently compare the performance with other methods. As

Table 4.18: Performance Comparison with existing Dictionary Learning methods [Xin+21]

Method	Accuracy (%)	Recall (%)	Precision (%)
ODL [Mai+10]	98.86 ± 0.29	$92.40 \pm \mathbf{1.63}$	95.36 ± 2.37
DLSI [RSS10]	$98.71 \pm \mathbf{0.28}$	92.21 ± 2.79	94.01 ± 2.77
FDDL [Yan+11]	98.25 ± 1.34	95.77 ± 3.65	87.23 ± 5.75
LRSDL [VM16]	98.11 ± 0.75	$\mathbf{96.79} \pm 2.10$	85.29 ± 4.15
GODL (**our**)	$\mathbf{99.00} \pm 0.36$	94.23 ± 2.6	$\mathbf{95.62 \pm 1.53}$

The best results of each class are in **bold**.

Table 4.19: Performance Comparison with existing Deep Learning methods (CV + CS) [Xin+21]

Method	Accuracy (% CS)	Accuracy (% CV)
ST-GCN [YXL18]	97.03 ± 0.83	97.45 ± 1.11
Biomechanic, RNN [XZ18]	97.40 ± 1.25	97.20 ± 1.79
Thining, DNN [TH19]	$\mathbf{99.20} \pm 1.10$	$\mathbf{99.20} \pm 1.56$
GODL (**our**)	$98.41 \pm \mathbf{0.04}$	$99.03 \pm \mathbf{0.15}$

The best results of each class are in **bold**.

depicted in Fig 4.10(a), the accuracy, precision, and recall of the methods are showcased across varying noise ratios ($2\% \sim 10\%$). While some methods may outperform ours in terms of recall and precision, our method consistently maintains a high level of accuracy even as the noise increases. This substantiates that our method exhibits greater robustness compared to the other four methods.

By adjusting the drop rate of the training set based on the weight distribution depicted in Fig 4.9 (b) and maintaining a fixed acceptance parameter α, we systematically varied the drop rate from 0% to 20% in increments of 5%. Each experiment was repeated 24 times, and the comparative results are illustrated in Fig 4.10 (left: accuracy error, middle: precision error, right: recall error).

As the drop rate increases, progressively more skeletons with low weights are excluded. The accuracy error of the ODL algorithm initially decreases until the drop rate reaches 10% but rises thereafter. In contrast, our method, GODL, maintains an accuracy error around 2% throughout the varying drop rates. From Fig 4.10, it is evident that the ODL method is more sensitive to changes in the training set. With the full dataset, ODL exhibits 0% average precision error and a recall error of 47.47%. However, as the drop rate reaches 20%, the precision error exceeds 11%, and the recall error falls below 4%.

Conversely, GODL demonstrates greater robustness, with an average precision error fluctuating within the range of $[1.72\%, 3.30\%]$ and an average recall error remaining in the range of $[2.59\%, 5.07\%]$.

In addition to its robustness, GODL exhibits superior average performance compared to ODL. The results presented in Table 4.18 illustrate that the average accuracy

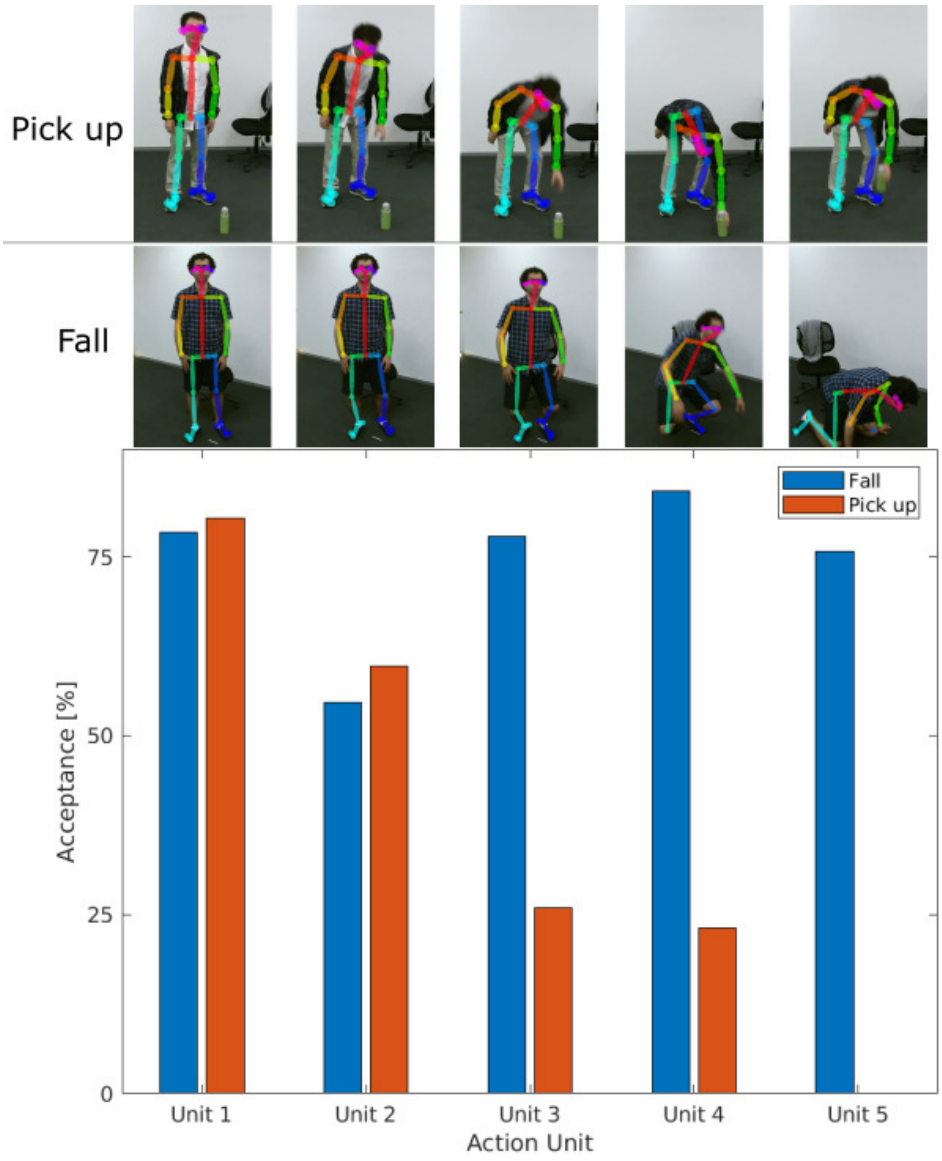


Figure 4.11: Visualization of prediction process [Xin+21].

of GODL is 98.45%, surpassing the baseline ODL by 3.88% over 24 experiments. Particularly noteworthy is the average precision of GODL, reaching 96.6%, showcasing a remarkable improvement of over 13% compared to ODL.

In addition to comparing with dictionary learning methods, we also evaluate our results against state-of-the-art deep learning-based fall-down detection methods, as summarized in Table 4.19. While deep learning-based methods exhibit slightly higher precision, our approach demonstrates more stable prediction results. Moreover, our method encodes spatial-temporal information, making it more interpretable than end-to-end deep learning methods. This interpretability is not achievable in end-to-end deep learning approaches, as they can only detect the fall-down action after it occurs.

In contrast to other spatial-temporal methods, our approach can discern which step of the fall-down action is in progress, as illustrated in Fig 4.11.

4.6 Uncertainty Quantification

As introduced in Section 3.5, we propose a novel uncertainty quantification method to preserve distance into the feature representation space. This section introduces the experimental result of uncertainty quantification of the introduced temporal fusion graph convolutional network (Section 3.3).

4.6.1 Ablation Study

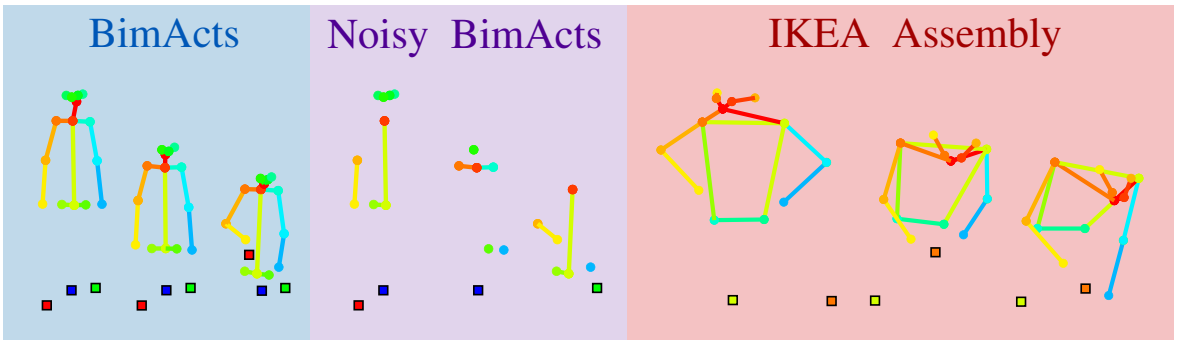
Table 4.20: Ablation study of the Spectral Normalized Residual connection and the Gaussian process in terms of uncertainty quantification on the Bimanual Actions dataset [Kre+21].

Res[a]			**GP**		**Evaluation Metrics - Bimanual Actions dataset**[b] (%)			
w/o	w	SN	w/o	w	ECE↓	TACE↓	ACE↓	SCE↓
×			×		13.98 ± 1.33	11.59 ± 0.42	90.56 ± 0.09	91.42 ± 0.08
×				×	20.98 ± 1.74	2.49 ± 0.14	81.61 ± 0.93	82.43 ± 0.94
	×		×		**8.21** ± 0.21	2.15 ± 0.12	91.75 ± 0.02	92.62 ± 0.02
	×			×	10.47 ± 0.34	**0.70** ± 0.03	**30.98** ± 0.36	**31.69** ± 0.36
		×	×		8.49 ± 0.21	2.59 ± 0.08	91.66 ± 0.02	92.57 ± 0.01
		×		×	11.24 ± 0.19	0.76 ± 0.02	34.57 ± 0.32	35.28 ± 0.33
w/o	w	SN	w/o	w	AUROC [1] ↑	AUPRC[1] ↑	AUROC [2] ↑	AUPRC[2] ↑
×			×		–	–	–	–
×				×	93.04 ± 7.75	90.76 ± 9.05	41.11 ± 7.89	44.61 ± 5.31
	×		×		–	–	–	–
	×			×	72.82 ± 3.41	68.08 ± 3.75	83.74 ± 1.11	84.37 ± 1.26
		×	×		–	–	–	–
		×		×	**99.39** ± 0.97	**99.13** ± 1.00	**90.19** ± 0.92	**92.07** ± 0.93

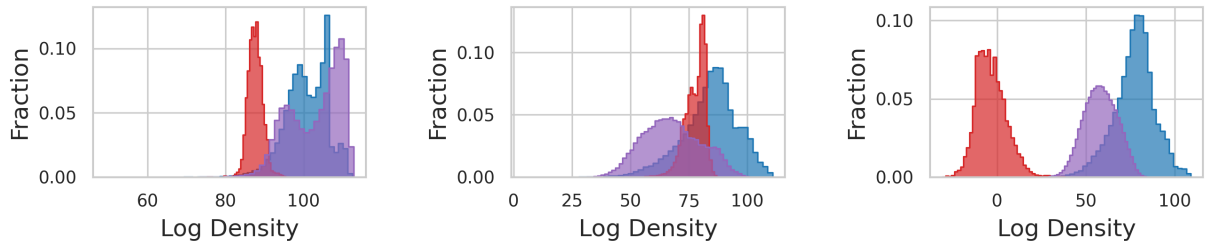
[a] The configurations are denoted as: "Res" = Residual connections; "SN" = spectral normalized; "w" = with; "w/o" = without; "GP" = Gaussian process kernel. The configuration without Gaussian process kernel is using *softmax* to output prediction.
[b] The evaluation results are averaged over 10 seeds. The best results cross all configurations are in **bold**; The best results for each setup with or without Gauss Process kernel are underlined. The evaluation metrics {AUROC[1], AUPRC[1]} and {AUROC[2], AUPRC[2]} are using the **IKEA Assembly dataset** and **Noisy BimActs** as OOD respectively.

n insightful observation from Table 4.20 is that the introduction of residual connections results in a smaller feature space distance (AUROC[1] and AUPRC[1]) between the Bimanual Actions test set (*in-distribution*) and the IKEA test set (*out-of-distribution*). Simultaneously, it leads to a larger feature space distance between the noisy Bimanual Actions (Noisy BimActs) test set and the original test set. These observations validate that the inclusion of residual connections shifts the *bi-Lipschitz* bounds to a higher range, thereby increasing the distance to meaningful changes in the input manifold while reducing sensitivity to *out-of-distribution* samples. It's essential to note that we consider the *noisy BimActs* dataset to encompass meaningful changes in the input manifold, given that it retains 50% of the original data.

The imposition of Spectral Normalized Residual connections establishes an upper constraint on the *Lipschitz* bound, thereby improving Out-of-Distribution (OOD) de-



(a) Left: Original Bimanual Actions dataset (BimActs) [Kre+21].
Middle: Noisy Bimanual Actions dataset. Right: IKEA Assembly dataset [Ben+20]



(b) Gaussian log probability distribution. Left: Without residual connections.
Middle: With normal residual connections. Right: With Spectral Normalized Residual connections

Figure 4.12: The log-probability distribution of feature space on the Bimanual Actions dataset [Kre+21], noisy Bimanual Actions dataset and the IKEA ASM [Ben+20] datasets.

tection while preserving sensitivity to manifold changes. The outcomes are visually evident in Fig 4.12, where the representation of feature space density is presented in Gaussian log-probability space. Nevertheless, it is evident that spectral normalization involves a trade-off between maintaining feature space distance and achieving high accuracy. As depicted in Table 4.12, the accuracy performance of the model with spectral normalization is 1% lower compared to the model with normal residual connections, even though its AUROC and AUPRC achieve the best results.

These observations prompt us to delve deeper into the analysis of the spectral normalization function's coefficient value in a quantitative way. A comparison of the results in Table 4.21 reveals that a higher coefficient value contributes to improved recognition and segmentation accuracy but results in a reduced feature space distance when detecting *out-of-distribution* instances. Conversely, lower coefficients exhibit the opposite trend. Notably, when the coefficient is too small, such as 1, the model converges to a local minimum, significantly diminishing both accuracy and the preservation of distance in the feature space. This trade-off is consistent with our hypothesis that spectral normalization maintains isometric properties within the model, albeit at the expense of the nonlinear mapping capabilities inherent in the learning method. We assert that this compromise is warranted in order to attain a more thorough comprehension of motion dynamics. The adjustment of the coefficient is contingent on the specific task at hand. For instance, in tasks such as image classification and segmentation, similar values can be applied since both involve mapping the pixel range $[0, 255]$ to the respective number of classes. Conversely, the range of human motion is inherently uncertain prior to its occurrence and differs across

Table 4.21: Comparison of coefficient values in spectral normalization function of TFGCN on the Bimanual Actions dataset [Kre+21].

c	Top 1 ↑	F1 micro ↑	F1@10 ↑	TACE ↓	ACE ↓	SCE ↓	AUROC[1] ↑	AUPRC[1] ↑
1	87.06	87.75	91.32	0.86	37.11	37,83	97.12	96.58
2	88.24	88.77	92.63	0.78	34.86	35.29	**99.94**	**99.88**
3	88.44	88.89	93.31	0.76	34.57	35.28	99.39	99.13
4	88.68	89.02	93.59	0.75	32.37	33.17	89.81	87.21
5	**89.27**	**89.60**	**94.37**	**0.71**	**32.20**	**32.91**	85.26	76.39

c is the coefficient parameter. The experiments are conducted on the leave-subject-one-out testset of the Bimanual Actions dataset [Kre+21]. The evaluation experiments are performed with *Gaussian Process* and the results are averaged over 10 seeds. The best results cross all setups are in **bold**.

Table 4.22: Uncertainty quantification performance on the Bimanual Actions dataset[a] [Kre+21].

Method	Accuracy ↑	F1 macro ↑	F1@10 ↑	F1@25 ↑	F1@50 ↑	#Parameters
MC-Dropout	88.32 ± 0.21	88.81 ± 0.20	**93.87** ± 0.20	**92.45** ± 0.23	84.45 ± 0.38	–
Ensemble	88.09 ± 0.69	88.56 ± 0.67	92.96 ± 0.68	91.65 ± 0.80	83.65 ± 1.20	–
DUQ[b] [Van+20b]	83.55 ± 1.06	83.91 ± 1.06	91.22 ± 0.66	89.04 ± 0.86	81.12 ± 1.3	20,264,026
SNGP[c] [Liu+20a]	87.63 ± 0.21	88.27 ± 0.16	91.64 ± 0.35	90.17 ± 0.43	83.03 ± 0.44	21,207,998
UQ-TFGCN	**88.44** ± 0.29	**88.89** ± 0.29	93.31 ± 0.29	92.18 ± 0.33	**84.76** ± 0.53	20,261,282

Method	TACE↓	ACE↓	SCE↓	AUROC [1] ↑	AUPRC[1] ↑	AUROC [2] ↑	AUPRC[2] ↑
MC-Dropout	0.77 ± 0.01	34.12 ± 0.23	34.83 ± 0.23	**99.95** ± 0.02	**99.89** ± 0.02	78.00 ± 0.09	81.38 ± 0.08
Ensemble	0.79 ± 0.05	**14.97** ± 0.55	**15.77** ± 0.56	99.81 ± 0.22	99.68 ± 0.33	86.29 ± 2.61	88.76 ± 2.17
DUQ[b] [Van+20b]	90.27 ± 0.15	90.27 ± 0.15	90.57 ± 0.15	83.58 ± 3.98	80.96 ± 5.56	52.86 ± 7.43	57.00 ± 8.98
SNGP[c] [Liu+20a]	90.78 ± 0.03	90.78 ± 0.03	91.13 ± 0.03	93.16 ± 3.44	84.67 ± 7.91	79.17 ± 1.89	78.03 ± 1.55
UQ-TFGCN	**0.76** ± 0.02	34.57 ± 0.32	35.28 ± 0.33	99.39 ± 0.97	99.13 ± 1.00	**90.19** ± 0.92	**92.07** ± 0.93

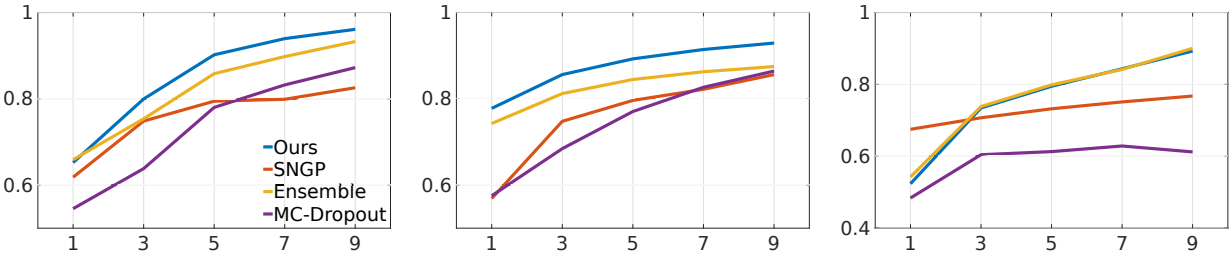
[a] The evaluation results are based on the Bimanual Actions dataset leave-subject-one-out testset and are averaged over 10 seeds. The best results cross all configurations are in **bold**. The evaluation metrics {AUROC[1], AUPRC[1]} and {AUROC[2], AUPRC[2]} are using the **IKEA Assembly dataset** and **Noisy BimActs** as OOD respectively.
[b] The *Radial Basis Funtion* (RBF) kernel [Van+20b] is implemented to measure the distance to class centroids.
[c] The *Laplace-approximated Neural Gaussian Process* [Liu+20a] is utilized instead of our Gaussian Process.

various motion tasks.

Another interesting observation is that substituting normal residual connections with Spectral Normalized Residual connections in UQ-TFGCN results in a decline in accuracy and F1 score performance, reaffirming the disruptive impact of spectral normalization. The potential advantages of Spectral Normalized Residual connections may not be fully realized in these two experiments, as most existing recognition and segmentation models typically overlook the assessment of feature distance.

4.6.2 Comparison with State-of-the-Art

To assess the performance of distance-awareness in the feature space, we conduct an out-of-distribution (OOD) detection experiment and compare the results with other popular deterministic network uncertainty quantification methods, namely, SNGP [Liu+20a] and DUQ [Van+20b]. It's important to note that SNGP [Liu+20a] and DUQ [Van+20b] were not originally designed for action recognition and segmentation. Consequently, we implement their feature space measuring mechanism instead of Gaussian Process (GP) in our model. Additionally, we utilize the *MC-Dropout* and *ensemble* methods as baselines, with a dropout ratio of 10% for *MC-Dropout*, and



(a) Impulse noise (b) Gaussian noise (c) Poisson noise

Figure 4.13: Comparison of AUROC with increasing of noise intensity in the Bimanual Actions Dataset [Kre+21] over different uncertainty quantification methods: Ensemble, MC-Dropout, SNGP and ours.

three ensembled models for the *ensemble* method.

As depicted in Table 4.22, the proposed method outperforms all other listed methods in terms of accuracy, AUROC, and AUPRC in the OOD detection on the noisy BimActs dataset. While both SNGP and DUQ exhibit higher efficiency, leveraging approximations like a *Laplace process* (SNGP [Liu+20a]) or an initially provided covariance scale (DUQ [Van+20b]), our model collects features from the training set and builds a multivariate Gaussian model, yielding superior performance. The covariance values of the Gaussian model range from 0.001 to 15.00 across different action categories. In contrast, DUQ relies solely on the initially provided covariance scale, contributing to its comparatively poorer performance. The efficacy of integrating spectral normalization (SN) into the residual connection is exemplified by the minimal number of parameters in our model. It is noteworthy that each model employed in the MC-Dropout and Ensemble methods shares an identical parameter count with our model.

An interesting observation is that randomly dropping features in *MC-Dropout* leads to a diminished OOD detection performance, indicating that our model is sensitive to known feature spaces. These findings motivate further exploration into the effects of different noise types and intensities.

Given the inevitability of noise in real-world scenarios, assessing OOD detection on noisy datasets highlights the practical advantages of maintaining distance in the feature space. To compare the impact of three common real-world noises—namely *impulse* noise, *Gaussian* noise, and *Poisson* noise—we present the corresponding results for different noise intensities in Fig 4.13. Recognizing that each noise type operates on a distinct intensity scale, we select three increasing unit intensities for varied noises.

For impulse noise, we introduce one unit of intensity by setting 10% of the test set to zeros. Gaussian noise sees an increase in unit intensity set at 0.1 variance of the test set. Meanwhile, Poisson noise experiences an increase in unit intensity set at an expected value and variance of 1000mm. As depicted in Fig4.13, our model consistently exhibits the highest AUROC in the presence of Gaussian and impulse noise, approaching the performance of the MC-Dropout method in Poisson noise. These results underscore the robustness of our model in preserving feature space distances under diverse noise conditions.

4.6.3 Qualitative Results

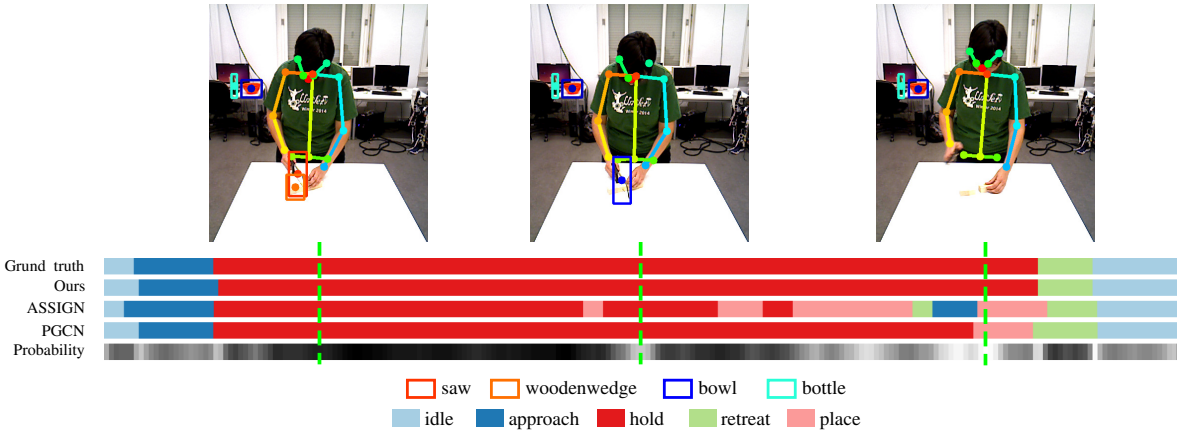


Figure 4.14: Qualitative results for action recognition and segmentation of the left hand in an example of *sawing* a wooden wedge from the Bimnual Actions dataset [Kre+21]. The probability, generated by our UQ-TFGCN model, is visually represented in a grayscale bar, and brighter grayscale values correspond to lower predicted probabilities.

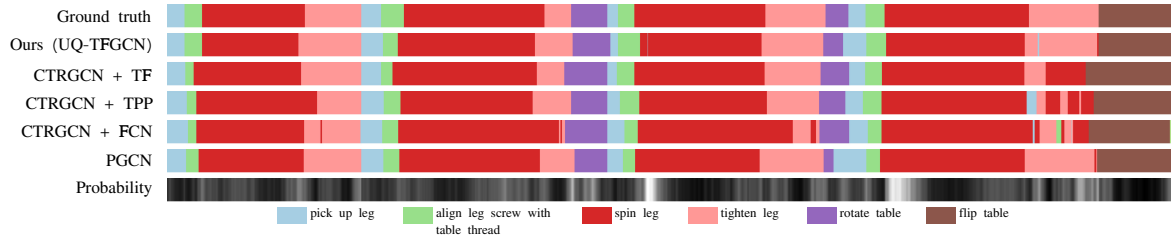


Figure 4.15: Qualitative uncertainty estimation and activity segmentation results of *assembly side table* example from the IKEA Assembly dataset [Ben+20]. Distinct actions are distinguished using various colors. A brighter grayscale value indicates a lower predicted probability.

We present detailed outputs, including the probabilities generated by our model and comparable methods, using examples from the Bimanual Action dataset [Kre+21]. Fig 4.14 illustrates an example involving left-hand actions during the *sawing* activity. A notable achievement of our model is its precise predictions of the start and end of segments, correctly identifying the frame indices for the *retreat* segment in the example. This underscores our model's significant improvement in preventing both shift- and over-segmentation. Additionally, our model outperforms other methods in action recognition accuracy. In this example, both ASSIGN [Mor+21] and PGCN [XB22b] incorrectly predict a *place* action, while our model successfully predicts a *hold* action consistent with the ground truth.

Another impressive achievement is that, in addition to predicting the action label, our model also outputs the predicted probability. As depicted at the bottom of Fig 4.14, this probability reflects the similarity between the current feature and known features from the training set, enhancing the interpretability of the model's predictions. Instances of incorrect predictions often occur at the junction of two sub-actions, where the probability is low, as the features at such moments deviate from

Table 4.23: Ablation study of the online sampling duration and gap. Each duration is combined with four different sampling gaps. The model is trained on our TUM-HOI data and evaluated on the subject 1 testset.

Duration				**Gap**				**Evaluation metrics on the TUM-HOI dataset*** (%)					
30	60	90	120	1	10	20	30	Top 1	F1 macro	F1@10	F1@25	F1@50	time (ms) ↓
×				×				72.97	73.58	82.42	79.79	**72.81**	18.60
×					×			72.72	73.37	82.27	79.21	72.15	18.68
×						×		72.51	73.23	82.38	79.01	71.83	20.36
×							×	72.57	73.20	82.68	79.59	71.71	19.59
	×			×				74.74	74.93	83.53	80.85	72.41	**17.46**
	×				×			74.64	74.82	83.46	80.64	71.99	17.83
	×					×		74.68	74.85	83.44	80.90	71.99	18.31
	×						×	74.53	74.79	83.70	81.14	72.33	20.75
		×		×				74.55	74.96	84.12	81.26	71.25	18.37
		×			×			74.56	75.00	84.36	81.45	71.16	18.65
		×				×		74.47	74.96	84.40	81.63	71.48	19.54
		×					×	74.58	75.07	**84.74**	**81.74**	71.90	20.11
			×	×				74.67	74.95	84.19	81.07	70.32	18.08
			×		×			74.71	75.01	84.24	81.14	70.28	18.19
			×			×		**74.85**	75.18	84.30	81.20	70.74	19.17
			×				×	74.82	**75.19**	84.35	81.55	70.92	18.99

* The best results cross all configurations are in **bold**; The best results for each gap setup with the same duration are underlined.

the distribution centers of both actions. Predictions with low probabilities in a continuous action indicate potential noise in the input, such as incorrect object detection. In this example, we select three representative frames within a continuous *hold* action, showcasing predictions with high (black), medium (gray), and low (white) probabilities. Frames with high probability correspond to precise object location and correct labels, while frames with medium and low probabilities indicate either mislabeled or missing objects.

Another qualitative result of a complex *assembly side table* task from the IKEA Assembly dataset [Ben+20] is demonstrated in Fig. 4.15. It can be seen that the proposed temporal fusion (TF) decoder has a better performance in preventing shift- and over-segmentation compared with Fast-FCN [Wu+19a] and *temporal pyramid pooling* (TPP) [XB22b] decoders.

4.7 Real-Time System for Understanding of Human-Object Interaction

As demonstrated in Section 3.7, the real-time system has an object detector and human pose estimator. Here, we employ YOLO [Red+16] as an object detector and OpenPose [Cao+19] as the pose estimator. However, the performance of the YOLO is unstable in a varying environment. Therefore, an Apriltag [Ols11] marker is attached on each object to enhance the unique features. The results are demonstrated on the Figure 4.16

Table 4.24: Cross validation on different testset of the TUM-HOI dataset.

Subject	Top 1 (%)	F1 micro (%)	F1@10 (%)	F1@25 (%)	F1@50 (%)
1	77.94	77.84	89.61	86.92	76.16
2	78.67	79.83	92.67	89.59	83.06
3	76.22	76.10	87.57	84.66	73.71
4	69.93	69.80	87.07	83.51	70.49
5	80.76	81.05	91.64	89.80	81.98
6	75.40	76.55	88.97	84.85	74.83
	76.49 ± 6.80	76.86 ± 7.20	89.59 ± 4.04	86.55 ± 4.87	76.70 ± 8.93

Considering the necessity of employing a sliding window to sample captured inputs for diverse application requirements in the implemented real-time system, this study delves into the analysis of the online sampling duration and sampling gap's impact. As presented in Table 4.23, longer video sequences contribute to enhanced model performance, indicating that increased temporal information is provided by extending the video duration. However, beyond 60 frames, the performance improvement diminishes, particularly in terms of F1@10 and F1@25, showing only marginal increases.

Furthermore, compared to the results obtained from complete video sequences with a top-1 accuracy of 77.94%, the performance of video clips is notably reduced. This reduction is attributed to the duplicate counts of misclassifications during the evaluation of results from video clips.

In addition, the impact of sampling gaps is found to be minor, irrespective of the length of video sequences. Regarding runtime, the video duration exhibits no significant influence on the model's running speed when the input is within 120 frames. In conclusion, based on the findings from the TUM-HOI dataset, this study recommends a video duration of 120 frames and a sampling gap of 20 frames as an optimal combination, yielding the best performance in terms of top-1 accuracy.

In addition to the ablation study conducted on the Subject 1 test set, a cross-validation experiment encompassing all subjects was performed on our TUM-HOI dataset. As illustrated in Table 4.24, the mean values of the cross-validation results are not significantly different from the outcomes of the previous experiments focused on Subject 1. However, the relatively large standard deviations can be attributed to the challenges posed by the Subject 4 test set.

Subject 4 presents challenges due to being smaller in stature compared to other subjects, potentially leading to model confusion. Additionally, the subject performs real actions in a real kitchen during the sitting part, introducing complexity through subtle subconscious actions.

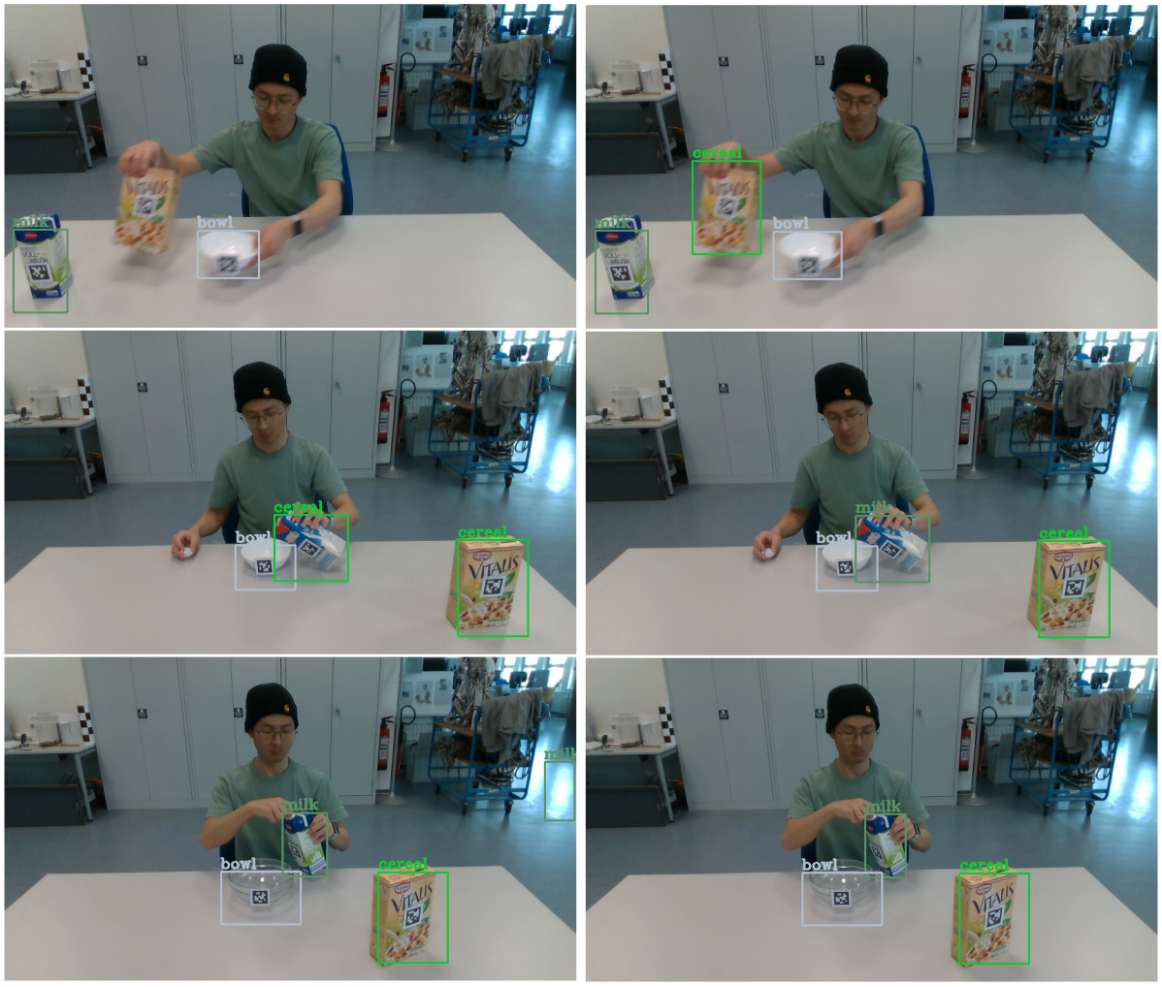


Figure 4.16: Comparison of the results of objects information from YOLO and YOLO + Apriltag. The left column shows the results from the YOLO and the right column is results of combining YOLO and Apriltag tracker. Each row presents a typical error type of object detection (left) and the results after correcting by marker features (right). Row 1: miss detection. Row 2: misclassification. Row 3: wrong detection of non-existing objects.

Chapter 5

Summary

This work introduces several novel methods for human action recognition, human-object interaction (HOI) detection and segmentation, event detection, and uncertainty quantification, which provides a foundation for applications in healthcare, human-robot collaboration, and beyond.

5.1 Conclusion of Proposed Methods

Firstly, the graph representation of action-related subjects and objects serves as a key mechanism to reduce noise and simplify the action recognition process.

Hybrid Attention: the introduction of a Hybrid Attention-based Graph Neural Network (HA-GCN) stands as a significant improvement in skeleton-based human action recognition. This novel approach combines a newly designed graph structure with a hybrid attention mechanism to efficiently extract and merge attention cues from different input streams. The hybrid attention layer includes two branches: *relative distance* and *relative angle* attention mechanisms. These branches work together within the spatial layer to seamlessly fuse attention cues using trainable parameters, thereby enriching the network's ability to recognize complex actions. Experiments on two large-scale datasets confirms the effectiveness of the hybrid attention model in enhancing the performance of multi-stream skeletal action recognition. Additionally,enhancements to the initial adjacency matrix, especially the incorporation of connections between heads, hands, and feet, help capture significant dynamic interactions between body-parts, thus strengthening the action recognition process.

However, the graph neural network requires structured data, necessitating a fixed number of nodes within the network architecture. Therefore, the introduction of new nodes requires the network to be retrained, representing the limitations of the current framework.

Pyramid Graph Convolutional Network: in addition to pure-human action recognition, a novel Pyramid Graph Convolutional Network (PGCN) is introduced for understanding human-object interaction relation sequences through action recognition and segmentation. This method integrates a spatial attention graph convolutional encoder and a temporal pyramid pooling decoder, forming a symbiotic relationship where each component complements the other. The spatial attention mechanism provides the decoder with high-level spatial relationships between graph nodes (human and objects), while the temporal pyramid pooling decoder helps upsample these

spatial features to the original temporal scale and subsequently predict frame labels. Experiments on two HOI datasets with different input formats (2D and 3D) demonstrate that PGCN has a general capability that can be applied to other structurally represented domains.

As can be seen from the introduced networks, the attention mechanism has been proven to be beneficial to the update of node relationships in a graph representation. However, different attention mechanisms have varying effects on different input types, e.g., joint position and bone connection. Therefore, achieving optimal results requires iterative exploration over multiple trials to find the most effective attention mechanism for a given task.

In the context of action segmentation, the importance of the decoder exceeds that of the encoder, since the encoder only provides compressed features, while the decoder is responsible for upsampling these features back to the original time scale. This observation highlights the critical role of the decoder in reconstructing temporally coherent action sequences from compressed feature representations.

Temporal Fusion Graph Convolutional Network: most of shift- and over- segmentation errors are caused by the decoder, which emphasizes the need for enhanced feature upsampling and improved action class detection. In doing so, a novel decoder equipped with a temporal fusion module is introduced, aiming to mitigate such errors and improve the accuracy of action segmentation. The integration of the new decoder with the temporal fusion module requires increased parameters and computational requirements. Despite this disadvantage, the inference process can still segment actions online even on standard desktop hardware, which confirms the utility of the proposed framework.

The proposed Temporal Fusion Graph Convolutional Network is evaluated on two public challenging human-object interaction datasets. The experimental results demonstrate that introducing the condensed features into the final upsampled feature map through residual connections can significantly improve the accuracy. In other words, the upsampling process inherently introduces uncertainty into the predictions. Exploiting residual connections effectively mitigates this uncertainty, thereby improving the overall accuracy performance of the network.

Spectral Normalization Residual connection: nevertheless, the network with residual connections still cannot differentiate between unknown and known data, resulting in a situation where different inputs are mapped to the same space. Therefore, a *spectral normalized* residual connection is introduced to preserve input distance in feature space and to recognize the novelty of input. The analysis of parameter quantities offers convincing evidence in favor of the efficiency of the proposed SN-res method. While spectral normalization helps maintain meaningful isometric properties, it does exhibit a trade-off in accuracy. In the context of safe collaboration between humans and robots, the prediction of uncertainty is more important than accuracy. Therefore, efforts to effectively measure uncertainty levels are critical to foster trust and facilitate seamless interactions between human and robotic agents. Results obtained from public Human-Object Interaction (HOI) datasets, featuring two distinct data formats (2D and 3D), demonstrate the general capability of our model. This versatility suggests potential applications in other domains represented with similar structural frameworks.

Gradual Online Dictionary Learning: In addition to daily action recognition,

a novel event detection method is addressed for an emergency situation, notably fall-down incidents. The proposed approach leverages a robust latent action unit extraction technique called Gradual Online Dictionary Learning (GODL). It outperforms existing dictionary learning methods and end-to-end deep learning methods in terms of robustness and average accuracy. Unlike other spatial-temporal methods, the proposed approach exhibits the unique capability to determine the specific phase of the fall-down action in progress. Therefore, it can not only detect fall-down activity but also predict and prevent it. This ability is especially useful in healthcare scenarios where nurses cannot supervise elderly patients all the time. Since the method has the ability to extract action unit, it can also be utilized in industry areas for robot learning from demonstration. An action unit can be seen as an action primitive and mapped to corresponding robot actions.

Real-time system: the final contribution of this research is the development of a novel real-time system designed to understand human-object interaction, thereby laying the foundation for effective human-machine collaboration. Most currently available HOI detection methods usually process the entire collected video or skeleton sequences offline. To tackle this problem, this research introduces a real-time system that receives RGB frames and depth frames from an RGB-D camera as input and outputs frame-wise action labels for left and right hands in real-time. This system enables robots to collaborate with humans during a task, rather than afterward. An improved YOLO method combined with markers is developed to enhance the multi-objects tracking algorithm.

In summary, this research represents significant contributions in the field of human action recognition, providing novel methods that address complexities inherent in multi-stream input, spatio-temporal information extraction, and real-time processing. The effectiveness of the proposed methods is confirmed by their enhanced performance on different datasets, emphasizing their potential applicability in other fields characterized by structural representation.

5.2 Future Work

Currently, our focus is primarily directed towards the relatively large movements of human body skeletons. A good extension of this work involves combining hand skeletal points and eye movements, as they have valuable insights into the recognition and comprehension of fine motions.

Expanding upon the trajectory of the ongoing research, a promising area for future investigation lies in minimizing errors in the action segmentation process. As aforementioned, discrepancies within the action segmentation process are mainly caused by the decoder network. Therefore, an important goal of future work is to design and implement novel decoder architectures. One promising approach is to provide the decoder with multiple potential anchors, each adapting to different action lengths during the decoding stage. Training the network under the constraints of non-overlapping and gap-less anchor connections is expected to correct segmentation errors.

The of attention mechanisms has yielded clear benefits in updating relationships between nodes. However, the effectiveness of different attention mechanisms varies

depending on the input data relationships. Hence, well-designed attention algorithms has the potential to manage dynamic relationships within a network. Moreover, manual predefinition of the number of nodes limits the scope of the attention mechanism to existing nodes. A proficient attention algorithm should possess the flexibility to accommodate varying numbers of graph nodes, thereby ensuring the design of a multi-functional attention mechanism.

Furthermore, the impact of object detection and human pose estimation results on action recognition outcomes underscores the importance of refining these algorithms. Notably, spectral normalization has been observed to compromise the nonlinear mapping capabilities of the proposed model. Therefore, future efforts can provide insights into the neural network mapping process and explore strategies to mitigate this deleterious effect.

The applicability of this work can be extended to the areas of human-robot collaboration and learning from demonstration. Since this work extracts human action unit, an intelligent robot can learn skills by projecting human actions into robot action space. Therefore, developing well-designed projection capabilities is a critical next step in promoting learning from demonstrations. Exploring the identification and tracking of multi-person participation behaviors is another important task to be explored. This task requires systems that can identify and monitor different individuals and their associated behaviors. Furthermore, in the context of human-robot collaboration, developing vision-based decision-making strategies becomes an important frontier. This requires predicting potential subsequent actions by observing ongoing actions and detecting novelty, allowing the robot to autonomously decide whether to perform the next task or seek human intervention. Additionally, we are planning to enhance the capabilities of our models through active learning based on novelty detection results, thus continuing the trajectory of innovation and progress in this field.

Appendix A

Appendix 1

Considering a graph convolutional layer $g(\boldsymbol{x})$ with residual connections: $g(\boldsymbol{x}) = r(\boldsymbol{x}) + m(\boldsymbol{x})$ where m represents main stream and composed of several hidden layers, r is residual branch and $\boldsymbol{x}$ is input. Assume that the upper *Lipschitz* boundaries of main and residual streams are initially defined by the normalization and activation functions $\forall \boldsymbol{x}$, denoted as β_m and β_r respectively. The lower boundaries for both streams are defined as the extremum 0. Note that in practice, the feature distance is unequal to 0 due to non-zero weights and biases in the convolution kernels. Simplify the processing functions to be $\boldsymbol{g}_{1,2}$, $\boldsymbol{r}_{1,2}$ and $\boldsymbol{m}_{1,2}$ where 1 and 2 mean with input $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ respectively, we get:

$$0 \leq \frac{||\boldsymbol{m}_2 - \boldsymbol{m}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} \leq \beta_m, \tag{A.1}$$

$$0 \leq \frac{||\boldsymbol{r}_2 - \boldsymbol{r}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} \leq \beta_r. \tag{A.2}$$

where we simplify the symbol of *Lipschitz* norm as $||\cdot||$.

Additional residual connections shift the range to a higher values as following:

$$\begin{aligned} ||\boldsymbol{r}_2 - \boldsymbol{r}_1|| =& ||\boldsymbol{g}_2 - \boldsymbol{m}_2 - (\boldsymbol{g}_1 - \boldsymbol{m}_1)|| \\ =& ||\boldsymbol{g}_2 - \boldsymbol{g}_1 + (\boldsymbol{m}_1 - \boldsymbol{m}_2)|| \\ \leq& ||\boldsymbol{g}_1 - \boldsymbol{g}_2|| + ||\boldsymbol{m}_1 - \boldsymbol{m}_2|| \\ \leq& ||\boldsymbol{g}_2 - \boldsymbol{g}_1|| + \beta_m ||\boldsymbol{x}_2 - \boldsymbol{x}_1||, \end{aligned} \tag{A.3}$$

where the last line follows by the bound assumptions $\forall \boldsymbol{x}$, we get the lower bound range of $g(\boldsymbol{x})$:

$$||\boldsymbol{r}_2 - \boldsymbol{r}_1|| - \beta_m ||\boldsymbol{x}_2 - \boldsymbol{x}_1|| \leq ||\boldsymbol{g}_2 - \boldsymbol{g}_1|| \tag{A.4}$$

$$\begin{aligned} -\beta_m \leq \frac{||\boldsymbol{r}_2 - \boldsymbol{r}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} - \beta_m \leq (\beta_r - \beta_m) \\ \leq \frac{||\boldsymbol{g}_2 - \boldsymbol{g}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||}. \end{aligned} \tag{A.5}$$

The upper bound can be easily obtain by:

$$\begin{aligned} \frac{||\boldsymbol{g}_2 - \boldsymbol{g}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} \leq& \frac{||\boldsymbol{r}_2 - \boldsymbol{r}_1|| + ||\boldsymbol{m}_2 - \boldsymbol{m}_1||}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} \\ \leq& (\beta_r + \beta_m). \end{aligned} \tag{A.6}$$

From Eq. A.5, it can be seen that the upper bound of residual stream shifts the feature space distance to a higher range, when $\beta_r > \beta_m$. In fact, the main stream produces a fine feature maps through several cascaded layers, while the residual outputs coarse features, which means that the *Lipschitz* upper bound of the feature space distance in the residual connection is larger than that of the main stream, i.e., $\beta_r > \beta_m$. Note that when $\beta_r \leq \beta_m$, the feature space distance automatically satisfies the constraint, since $-\beta_m \leq \beta_r - \beta_m \leq 0$ and $0 \leq ||g_2 - g_1||$. Hence, constraining the *Lipschitz* upper bound of residual connections is crucial to preserve distance in the representation space.

Bibliography

[AC17] Alex Kendall, V. B. and Cipolla, R. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2017, pp. 57.1–57.12. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.57. URL: https://dx.doi.org/10.5244/C.31.57.

[And+18] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. "Bottom-up and top-down attention for image captioning and visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6077–6086.

[Ant+19] Antonik, P., Marsal, N., Brunner, D., and Rontani, D. "Human action recognition with a large-scale brain-inspired photonic computer". In: *Nature Machine Intelligence* 1.11 (2019), pp. 530–537.

[Avo+19] Avola, D., Cascio, M., Cinque, L., Foresti, G. L., Massaroni, C., and Rodolà, E. "2-D skeleton-based action recognition via two-branch stacked LSTM-RNNs". In: *IEEE Transactions on Multimedia* 22.10 (2019), pp. 2481–2496.

[BWM17] Baradel, F., Wolf, C., and Mille, J. "Human action recognition: Pose-based attention draws focus to hands". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 604–613.

[Beh+19] Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. "Invertible residual networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 573–582.

[BDB18] Ben Tanfous, A., Drira, H., and Ben Amor, B. "Coding Kendall's shape trajectories for 3D action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2840–2849.

[Ben+20] Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., and Gould, S. "The IKEA ASM Dataset: Understanding People Assembling Furniture through Actions, Objects and Pose". In: (2020).

[BCV13] Bengio, Y., Courville, A., and Vincent, P. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[Blu+15] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[Cao+19] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[CZ17] Carreira, J. and Zisserman, A. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[CZG20] Charpentier, B., Zügner, D., and Günnemann, S. "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1356–1367.

[CZZ20] Chen, G., Zhang, C., and Zou, Y. "AFNet: Temporal Locality-aware Network with Dual Structure for Accurate and Fast Action Detection". In: *IEEE Transactions on Multimedia* (2020).

[Che+21] Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., and Hu, W. "Channel-wise topology refinement graph convolution for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13359–13368.

[Che+20] Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., and Lu, H. "Skeleton-based action recognition with shift graph convolutional network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 183–192.

[Chi+13] Chiang, C.-K., Su, T.-F., Yen, C., and Lai, S.-H. "Multi-attributed Dictionary Learning for Sparse Coding". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1137–1144.

[Cho17] Chollet, F. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[Dai+22] Dai, R., Das, S., Sharma, S., Minciullo, L., Garattoni, L., Bremond, F., and Francesca, G. "Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2022), pp. 2533–2550.

[DT05] Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[De +17] De Miguel, K., Brunete, A., Hernando, M., and Gambao, E. "Home camera-based fall detection system for the elderly". In: *Sensors* 17.12 (2017), p. 2864.

[Dev+19] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*. 2019, pp. 4171–4186.

[DWA19] Dreher, C. R., Wächter, M., and Asfour, T. “Learning object-action relations from bimanual human demonstration using graph networks”. In: *IEEE Robotics and Automation Letters* 5.1 (2019), pp. 187–194.

[Dua+21] Duan, H., Zhao, Y., Chen, K., Shao, D., Lin, D., and Dai, B. “Revisiting Skeleton-based Action Recognition”. In: *arXiv preprint arXiv:2104.13586* (2021).

[Fan+09] Fan, Q., Bobbitt, R., Zhai, Y., Yanagawa, A., Pankanti, S., and Hampapur, A. “Recognition of repetitive sequential human activity”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 943–950.

[FR13] Fathi, A. and Rehg, J. M. “Modeling Actions through State Changes”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2579–2586. DOI: 10.1109/CVPR.2013.333.

[FPZ16] Feichtenhofer, C., Pinz, A., and Zisserman, A. “Convolutional two-stream network fusion for video action recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1933–1941.

[Fel+09] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.

[Fer+15] Fernando, B., Gavves, E., Oramas, J. M., Ghodrati, A., and Tuytelaars, T. “Modeling video evolution for action recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5378–5387.

[Fer+17] Ferrari, C., Lisanti, G., Berretti, S., and Del Bimbo, A. “A Dictionary Learning-Based 3D Morphable Shape Model”. In: *IEEE Transactions on Multimedia* 19.12 (2017), pp. 2666–2679.

[FB81] Fischler, M. A. and Bolles, R. C. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[GG16] Gal, Y. and Ghahramani, Z. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[Gaw+23] Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. “A survey of uncertainty in deep neural networks”. In: *Artificial Intelligence Review* (2023), pp. 1–77.

[Guo+17] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. “On calibration of modern neural networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1321–1330.

[HBL15] Henaff, M., Bruna, J., and LeCun, Y. “Deep convolutional networks on graph-structured data”. In: *CoRR* (2015).

[HAT03] Horimoto, S., Arita, D., and Taniguchi, R.-i. “Real-time hand shape recognition for human interface”. In: *12th International Conference on Image Analysis and Processing, 2003. Proceedings.* IEEE. 2003, pp. 20–25.

[Hua+18] Huang, Z., Liu, Y., Fang, Y., and Horn, B. K. “Video-based fall detection for seniors with human pose estimation”. In: *2018 4th International Conference on Universal Village (UV)*. IEEE. 2018, pp. 1–4.

[HW21] Hüllermeier, E. and Waegeman, W. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110 (2021), pp. 457–506.

[Hus+13] Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. “Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations”. In: *Twenty-third international joint conference on artificial intelligence*. 2013.

[HGS19] Hussein, N., Gavves, E., and Smeulders, A. W. “Timeception for complex action recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 254–263.

[Iof17] Ioffe, S. “Batch renormalization: Towards reducing minibatch dependence in batch-normalized models”. In: (2017), pp. 1942–1950.

[Jal+12] Jalal, A., Uddin, M. Z., Kim, J. T., and Kim, T.-S. “Recognition of human home activities via depth silhouettes and R transformation for smart homes”. In: *Indoor and Built Environment* 21.1 (2012), pp. 184–190.

[JAT20] Jameel, T., Ali, R., and Toheed, I. “Ethics of Artificial Intelligence: Research Challenges and Potential Solutions”. In: *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. 2020, pp. 1–6. DOI: 10.1109/iCoMET48670.2020.9073911.

[KJM10] Karami, A.-B., Jeanpierre, L., and Mouaddib, A.-I. “Human-robot collaboration for a shared mission”. In: *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2010, pp. 155–156.

[Kay+17] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. “The kinetics human action video dataset”. In: *CoRR* (2017).

[Ke+17] Ke, Q., An, S., Bennamoun, M., Sohel, F., and Boussaid, F. “Skeletonnet: Mining deep part features for 3-d action recognition”. In: *IEEE signal processing letters* 24.6 (2017), pp. 731–735.

[KG17] Kendall, A. and Gal, Y. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems* 30 (2017).

[Kes+17] Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., and Bhowmik, A. "Intel RealSense Stereoscopic Depth Cameras". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 1–10.

[KJH21] Kim, T. S., Jones, J., and Hager, G. D. "Motion guided attention fusion to recognize interactions from videos". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13076–13086.

[KR17] Kim, T. S. and Reiter, A. "Interpretable 3d human action analysis with temporal convolutional networks". In: *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*. IEEE. 2017, pp. 1623–1631.

[KW17] Kipf, T. N. and Welling, M. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations (ICLR)*. 2017.

[KW12] Kong, S. and Wang, D. "A dictionary learning approach for classification: separating the particularity and the commonality". In: *European conference on computer vision*. Springer. 2012, pp. 186–199.

[Kon+18] Kong, X., Meng, Z., Meng, L., and Tomiyama, H. "A privacy protected fall detection IoT system for elderly persons using depth camera". In: *2018 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE. 2018, pp. 31–35.

[Kös+12] Köstinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H. "Large Scale Metric Learning from Equivalence Constraints". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[Kre+21] Krebs, F., Meixner, A., Patzer, I., and Asfour, T. "The KIT Bimanual Manipulation Dataset". In: *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2021, pp. 499–506.

[KSH12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[KD18] Kucukyilmaz, A. and Demiris, Y. "Learning Shared Control by Demonstration for Personalized Wheelchair Assistance". In: *IEEE Transactions on Haptics* 11.3 (2018), pp. 431–442. DOI: 10.1109/TOH.2018.2804911.

[Kul+21] Kulak, T., Girgin, H., Odobez, J.-M., and Calinon, S. "Active Learning of Bayesian Probabilistic Movement Primitives". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2163–2170. DOI: 10.1109/LRA.2021.3060414.

[Kul+13] Kulis, B. et al. "Metric learning: A survey". In: *Foundations and Trends® in Machine Learning* 5.4 (2013), pp. 287–364.

[Kur+22] Kurillo, G., Hemingway, E., Cheng, M.-L., and Cheng, L. "Evaluating the accuracy of the azure kinect and kinect v2". In: *Sensors* 22.7 (2022), p. 2469.

[Lad+20] Ladjailia, A., Bouchrika, I., Merouani, H. F., Harrati, N., and Mahfouf, Z. "Human activity recognition via optical flow: decomposing activities into basic actions". In: *Neural Computing and Applications* 32 (2020), pp. 16387–16400.

[Lag+23] Lagamtzis, D., Schmidt, F., Seyler, J., Dang, T., and Schober, S. "Exploiting Spatio-Temporal Human-Object Relations Using Graph Neural Networks for Human Action Recognition and 3D Motion Forecasting". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 7832–7838.

[LPB17] Lakshminarayanan, B., Pritzel, A., and Blundell, C. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30 (2017).

[LP07] Laptev, I. and Pérez, P. "Retrieving actions in movies". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–8.

[LM+14] Le, T.-L., Morel, J., et al. "An analysis on human fall detection using skeleton from Microsoft Kinect". In: *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*. IEEE. 2014, pp. 484–489.

[Lea+17] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. "Temporal Convolutional Networks for Action Segmentation and Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1003–1012. DOI: 10.1109/CVPR.2017.113.

[LeC+98] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[Li+21a] Li, C., Xie, C., Zhang, B., Han, J., Zhen, X., and Chen, J. "Memory attention networks for skeleton-based action recognition". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[Li+17] Li, C., Zhong, Q., Xie, D., and Pu, S. "Skeleton-based action recognition with convolutional neural networks". In: *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2017, pp. 597–600.

[Li+18] Li, C., Zhong, Q., Xie, D., and Pu, S. "Co-Occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 786–792. ISBN: 9780999241127.

[Li+20a] Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., and Sebe, N. "Spatio-temporal attention networks for action recognition and detection". In: *IEEE Transactions on Multimedia* 22.11 (2020), pp. 2990–3001.

[Li+19] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. "Actional-structural graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3595–3603.

[Li+21b] Li, Q., Xu, J., Wang, J., Jing, Y., and Wang, X. “Uncertainty quantification enforced flash radiography reconstruction by two-level efficient MCMC”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 7184–7199.

[Li+20b] Li, Y., Wang, H., Dang, L. M., Nguyen, T. N., Han, D., Lee, A., Jang, I., and Moon, H. “A deep learning-based hybrid framework for object detection and recognition in autonomous driving”. In: *IEEE Access* 8 (2020), pp. 194228–194239.

[Li+16] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. “Gated graph sequence neural networks”. In: *4th International Conference on Learning Representations, (ICLR)*. 2016.

[Lia+15] Liao, S., Hu, Y., Zhu, X., and Li, S. Z. “Person re-identification by local maximal occurrence representation and metric learning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2197–2206.

[LLL18] Lie, W.-N., Le, A. T., and Lin, G.-H. “Human fall-down event detection based on 2D skeletons and deep learning approach”. In: *2018 International Workshop on Advanced Image Technology (IWAIT)*. IEEE. 2018, pp. 1–4.

[Lie+19] Lie, W.-N., Lin, G.-H., Shih, L.-S., Hsu, Y., Nguyen, T. H., and Nhu, Q. N. Q. “Fully Convolutional Network for 3D Human Skeleton Estimation from a Single View for Action Analysis”. In: *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2019, pp. 1–6.

[Lin+20] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. “Focal Loss for Dense Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 318–327. DOI: 10.1109/TPAMI.2018.2858826.

[Lin+23] Ling, Y., Ma, Z., Xie, B., Zhang, Q., and Weng, X. “SA-BiGCN: Bi-Stream Graph Convolution Networks With Spatial Attentions for the Eye Contact Detection in the Wild”. In: *IEEE Transactions on Intelligent Transportation Systems* (2023).

[Liu+17] Liu, C., Mao, J., Sha, F., and Yuille, A. “Attention correctness in neural image captioning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 2017.

[Liu+20a] Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. “Simple and principled uncertainty estimation with deterministic deep learning via distance awareness”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7498–7512.

[LLS09] Liu, J., Luo, J., and Shah, M. “Recognizing realistic actions from videos “in the wild””. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 1996–2003.

[LLC17] Liu, M., Liu, H., and Chen, C. “Enhanced skeleton visualization for view invariant human action recognition”. In: *Pattern Recognition* 68 (2017), pp. 346–362.

[Liu+20b] Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. “Disentangling and unifying graph convolutions for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 143–152.

[Low99] Lowe, D. G. “Object recognition from local scale-invariant features”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[Ma+04] Ma, L., Tan, T., Wang, Y., and Zhang, D. “Efficient iris recognition by characterizing key local variations”. In: *IEEE Transactions on Image processing* 13.6 (2004), pp. 739–750.

[Ma+14] Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., and Li, Y. “Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine”. In: *IEEE journal of biomedical and health informatics* 18.6 (2014), pp. 1915–1922.

[Mai+10] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. “Online Learning for Matrix Factorization and Sparse Coding”. In: *Journal of Machine Learning Research* 11.1 (2010).

[MG18] Malinin, A. and Gales, M. “Predictive uncertainty estimation via prior networks”. In: *Advances in neural information processing systems* 31 (2018).

[Mil+16] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. “MOT16: A benchmark for multi-object tracking”. In: *arXiv preprint arXiv:1603.00831* (2016).

[Min+18] Min, W., Yao, L., Lin, Z., and Liu, L. “Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle”. In: *IET Computer Vision* 12.8 (2018), pp. 1133–1140.

[Mir+12] Mirmahboub, B., Samavi, S., Karimi, N., and Shirani, S. “Automatic monocular system for human fall detection based on variations in silhouette area”. In: *IEEE transactions on biomedical engineering* 60.2 (2012), pp. 427–436.

[Mor+21] Morais, R., Le, V., Venkatesh, S., and Tran, T. “Learning asynchronous and sparse human-object interaction in videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16041–16050.

[NK10] Najmaei, N. and Kermani, M. R. “Applications of artificial intelligence in safe human–robot interactions”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.2 (2010), pp. 448–459.

[NAM12] Nghiem, A. T., Auvinet, E., and Meunier, J. “Head detection using Kinect camera and its application to fall detection”. In: *2012 11th international conference on information science, signal processing and their applications (ISSPA)*. IEEE. 2012, pp. 164–169.

[Nie95] Nielsen, M. “Surface reconstruction: GNCs and MFA”. In: *Proceedings of IEEE International Conference on Computer Vision*. IEEE. 1995, pp. 344–349.

[Nix+19] Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. “Measuring Calibration in Deep Learning.” In: *CVPR workshops*. Vol. 2. 7. 2019.

[Ols11] Olson, E. “AprilTag: A robust and flexible visual fiducial system”. In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 3400–3407.

[Ond+13] Ondobaka, S., Newman-Norlund, R. D., Lange, F. P. de, and Bekkering, H. “Action recognition depends on observer’s level of action control and social personality traits”. In: *PLoS One* 8.11 (2013), e81392.

[PP06] Pantic, M. and Patras, I. “Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36.2 (2006), pp. 433–449. DOI: 10.1109/TSMCB.2005.859075.

[PD+20] Parsa, B., Dariush, B., et al. “Spatio-temporal pyramid graph convolutions for human action recognition and postural assessment”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1080–1090.

[PCM20] Plizzari, C., Cannici, M., and Matteucci, M. “Spatial temporal transformer network for skeleton-based action recognition”. In: *arXiv preprint arXiv:2008.07404* (2020).

[Pli+22] Plizzari, C., Planamente, M., Goletto, G., Cannici, M., Gusso, E., Matteucci, M., and Caputo, B. “E2 (go) motion: Motion augmented event stream for egocentric action recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 19935–19947.

[Pöh+16] Pöhlmann, S. T., Harkness, E. F., Taylor, C. J., and Astley, S. M. “Evaluation of Kinect 3D sensor for healthcare imaging”. In: *Journal of medical and biological engineering* 36.6 (2016), pp. 857–870.

[Qi+18] Qi, J., Wang, Z., Lin, X., and Li, C. “Learning Complex Spatio-Temporal Configurations of Body Joints for Online Activity Recognition”. In: *IEEE Transactions on Human-Machine Systems* 48.6 (2018), pp. 637–647.

[Qia+22] Qiao, T., Men, Q., Li, F. W., Kubotani, Y., Morishima, S., and Shum, H. P. “Geometric Features Informed Multi-person Human-Object Interaction Recognition in Videos”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 474–491.

[Rab89] Rabiner, L. R. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[Rah+14] Rahmani, H., Mahmood, A., Huynh, D. Q., and Mian, A. “Real time action recognition using histograms of depth gradients and random decision forests”. In: *IEEE winter conference on applications of computer vision*. IEEE. 2014, pp. 626–633.

[RM16] Rahmani, H. and Mian, A. “3d action recognition from novel viewpoints”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1506–1515.

[RSS10] Ramirez, I., Sprechmann, P., and Sapiro, G. “Classification and clustering via dictionary learning with structured incoherence and shared features”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 3501–3508.

[RC90] Rangarajan, A. and Chellappa, R. “Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation”. In: *[1990] Proceedings. 10th International Conference on Pattern Recognition*. Vol. 2. IEEE. 1990, pp. 127–133.

[Red+16] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[Rei+22] Reily, B., Gao, P., Han, F., Wang, H., and Zhang, H. “Real-time recognition of team behaviors by multisensory graph-embedded robot learning”. In: *The International Journal of Robotics Research* 41.8 (2022), pp. 798–811.

[Ren+15] Ren, S., He, K., Girshick, R., and Sun, J. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).

[Rou+11] Rougier, C., Meunier, J., St-Arnaud, A., and Rousseau, J. “Robust video surveillance for fall detection based on human shape deformation”. In: *IEEE Transactions on circuits and systems for video Technology* 21.5 (2011), pp. 611–622.

[RHK18] Ruan, W., Huang, X., and Kwiatkowska, M. “Reachability Analysis of Deep Neural Networks with Provable Guarantees”. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI’18. Stockholm, Sweden: AAAI Press, 2018, pp. 2651–2659. ISBN: 9780999241127.

[San+18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[Sev+19] Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., and Black, M. J. "On the integration of optical flow and action recognition". In: *Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40*. Springer. 2019, pp. 281–297.

[Sha+16a] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. "Ntu rgb+ d: A large scale dataset for 3d human activity analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1010–1019.

[Sha+16b] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2016.

[Shi+19a] Shi, L., Zhang, Y., Cheng, J., and Lu, H. "Skeleton-based action recognition with directed graph neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7912–7921.

[Shi+19b] Shi, L., Zhang, Y., Cheng, J., and Lu, H. "Two-stream adaptive graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12026–12035.

[SWC16] Shou, Z., Wang, D., and Chang, S.-F. "Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1049–1058. DOI: 10.1109/CVPR.2016.119.

[Si+19] Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. "An attention enhanced graph convolutional lstm network for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1227–1236.

[SS14] Stone, E. E. and Skubic, M. "Fall detection in homes of older adults using the Microsoft Kinect". In: *IEEE journal of biomedical and health informatics* 19.1 (2014), pp. 290–301.

[Su+23] Su, S., Li, Y., He, S., Han, S., Feng, C., Ding, C., and Miao, F. "Uncertainty quantification of collaborative detection for self-driving". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 5588–5594.

[TFK12] Tang, K., Fei-Fei, L., and Koller, D. "Learning latent temporal structure for complex event detection". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 1250–1257.

[Tan+18] Tang, Y., Tian, Y., Lu, J., Li, P., and Zhou, J. "Deep progressive reinforcement learning for skeleton-based action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5323–5332.

[TP13] Tra, K. and Pham, T. V. “Human fall detection based on adaptive background mixture model and HMM”. In: *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*. IEEE. 2013, pp. 95–100.

[Tra+23] Tran, H., Le, V., Venkatesh, S., and Tran, T. “Persistent-Transient Duality: A Multi-mechanism Approach for Modeling Human-Object Interaction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9858–9867.

[Trö+21] Tröbinger, M., Costinescu, A., Xing, H., Elsner, J., Hu, T., Naceri, A., Figueredo, L., Jensen, E., Burschka, D., and Haddadin, S. “A Dual Doctor-Patient Twin Paradigm for Transparent Remote Examination, Diagnosis, and Rehabilitation”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 2933–2940. DOI: 10.1109/IROS51168.2021.9636626.

[TH19] Tsai, T. and Hsu, C. “Implementation of Fall Detection System Based on 3D Skeleton for Deep Learning Technique”. In: *IEEE Access* 7 (2019), pp. 153049–153059. DOI: 10.1109/ACCESS.2019.2947518.

[Tur+08] Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. “Machine recognition of human activities: A survey”. In: *IEEE Transactions on Circuits and Systems for Video technology* 18.11 (2008), pp. 1473–1488.

[Van+20a] Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. “Uncertainty estimation using a single deep deterministic neural network”. In: *International conference on machine learning*. PMLR. 2020, pp. 9690–9700.

[Van+20b] Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. “Uncertainty estimation using a single deep deterministic neural network”. In: *International conference on machine learning*. PMLR. 2020, pp. 9690–9700.

[Vas+17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[VWB16] Veit, A., Wilber, M. J., and Belongie, S. “Residual networks behave like ensembles of relatively shallow networks”. In: *Advances in neural information processing systems* 29 (2016).

[Vel+18] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. “Graph Attention Networks”. In: *International Conference on Learning Representations* (2018).

[VAC14] Vemulapalli, R., Arrate, F., and Chellappa, R. “Human action recognition by representing 3d skeletons as points in a lie group”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 588–595.

[VT10] Venkatesha, S. and Turk, M. “Human activity recognition using local shape descriptors”. In: *2010 20th International Conference on Pattern Recognition*. IEEE. 2010, pp. 3704–3707.

[VSG13] Volkhardt, M., Schneemann, F., and Gross, H.-M. “Fallen Person Detection for Mobile Robots Using 3D Depth Data”. In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 2013, pp. 3573–3578.

[VM16] Vu, T. H. and Monga, V. “Learning a low-rank shared dictionary for object classification”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 4428–4432.

[WW17] Wang, H. and Wang, L. “Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 499–508.

[Wan+12] Wang, J., Liu, Z., Wu, Y., and Yuan, J. “Mining actionlet ensemble for action recognition with depth cameras”. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 1290–1297.

[Wan+16] Wang, P., Cao, Y., Shen, C., Liu, L., and Shen, H. T. “Temporal pyramid pooling-based convolutional neural network for action recognition”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2016), pp. 2613–2622.

[WM10] Wang, Y. and Mori, G. “Hidden part models for human action recognition: Probabilistic versus max margin”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.7 (2010), pp. 1310–1323.

[WSZ19] Wei, P., Sun, H., and Zheng, N. “Learning composite latent structures for 3D human action representation and recognition”. In: *IEEE Transactions on Multimedia* 21.9 (2019), pp. 2195–2208.

[Wen+19] Wen, Y.-H., Gao, L., Fu, H., Zhang, F.-L., and Xia, S. “Graph CNNs with motif and variable temporal block for skeleton-based action recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8989–8996.

[WSM14] Wilson, S., Srinivas, M., and Mohan, C. K. “Dictionary based action video classification with action bank”. In: *2014 19th International Conference on Digital Signal Processing*. IEEE. 2014, pp. 597–600.

[WB18] Wojke, N. and Bewley, A. “Deep Cosine Metric Learning for Person Reidentification”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 748–756. DOI: 10.1109/WACV.2018.00087.

[WBP17] Wojke, N., Bewley, A., and Paulus, D. “Simple online and realtime tracking with a deep association metric”. In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649.

[Wri+08] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. “Robust face recognition via sparse representation”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.2 (2008), pp. 210–227.

[Wu+19a] Wu, H., Zhang, J., Huang, K., Liang, K., and Yu, Y. “FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation.” In: *CoRR* abs/1903.11816 (2019).

[Wu+19b] Wu, J., Wang, K., Cheng, B., Li, R., Chen, C., and Zhou, T. "Skeleton Based Fall Detection with Convolutional Neural Network". In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE. 2019, pp. 5266–5271.

[XRV17] Xiao, H., Rasul, K., and Vollgraf, R. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

[XB22a] Xing, H. and Burschka, D. "Skeletal human action recognition using hybrid attention based graph convolutional network". In: *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE. 2022, pp. 3333–3340.

[XB22b] Xing, H. and Burschka, D. "Understanding Spatio-Temporal Relations in Human-Object Interaction using Pyramid Graph Convolutional Network". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 5195–5201.

[Xin+21] Xing, H., Xue, Y., Zhou, M., and Burschka, D. "Robust event detection based on spatio-temporal latent action unit using skeletal information". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 2941–2948.

[Xu+15] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.

[XZ18] Xu, T. and Zhou, Y. "Elders' fall detection based on biomechanical features using depth camera". In: *International journal of wavelets, multiresolution and information processing* 16.02 (2018), p. 1840005. ISSN: 0219-6913.

[YOI92] Yamato, J., Ohya, J., and Ishii, K. "Recognizing human action in time-sequential images using hidden Markov model." In: *CVPR*. Vol. 92. 1992, pp. 379–385.

[Yan+23] Yan, R., Xie, L., Shu, X., Zhang, L., and Tang, J. "Progressive Instance-Aware Feature Learning for Compositional Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[YXL18] Yan, S., Xiong, Y., and Lin, D. "Spatial temporal graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the AAAI conference on artificial intelligence*. 2018.

[YCL16] Yang, F., Choi, W., and Lin, Y. "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2129–2137.

[Yan+20] Yang, H., Antonante, P., Tzoumas, V., and Carlone, L. "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1127–1134.

[Yan+11] Yang, M., Zhang, L., Feng, X., and Zhang, D. “Fisher discrimination dictionary learning for sparse representation”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 543–550.

[YT12] Yang, X. and Tian, Y. L. “Eigenjoints-based action recognition using naive-bayes-nearest-neighbor”. In: *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE. 2012, pp. 14–19.

[Ye+21] Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., and Hoi, S. C. “Deep learning for person re-identification: A survey and outlook”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.6 (2021), pp. 2872–2893.

[ZI06] Zelnik-Manor, L. and Irani, M. “Statistical analysis of dynamic actions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.9 (2006), pp. 1530–1535.

[Zha+17] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. “View adaptive recurrent neural networks for high performance human action recognition from skeleton data”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2117–2126.

[Zha+16] Zhang, S., Li, Z., Wei, Z., and Wang, S. “An automatic human fall detection approach using RGBD cameras”. In: *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*. IEEE. 2016, pp. 781–784.

[ZWZ19] Zhang, Y., Wang, T., and Zhang, Y. “Tracking with the CAD Model of Object for Visual Servoing”. In: *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2019, pp. 1428–1432.

[Zhe+20] Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., Shen, J., Kehtarnavaz, N., and Shah, M. “Deep Learning-Based Human Pose Estimation: A Survey”. In: *arXiv preprint arXiv:2012.13392* (2020).

[Zhe+15] Zheng, L., Shen, L., Tian, L., Wang, S., Bu, J., and Tian, Q. “Person re-identification meets image search”. In: *arXiv preprint arXiv:1502.02171* (2015).

[Zhe+17] Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., and Tian, Q. “Person re-identification in the wild”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1367–1376.

[Zhe+21] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[Zhe+19] Zheng, Y., Zhang, D., Yang, L., and Zhou, Z. “Fall detection and recognition based on GCN and 2D Pose”. In: *2019 6th International Conference on Systems and Informatics (ICSAI)*. IEEE. 2019, pp. 558–562.

[ZPK16] Zhou, Q.-Y., Park, J., and Koltun, V. “Fast global registration”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 766–782.