Hybrid architecture for human action recognition using skeleton data

by

Muhammad Salik Nadeem

A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Masters of Science in **Computer Science**

Faculty of Science
Ontario Tech University
Oshawa, Ontario, Canada
December, 2024

THESIS EXAMINATION INFORMATION

Submitted by: **Muhammad Salik Nadeem**

Master of Science in **Computer Science**

Thesis Title: Hybrid architecture for human action recognition using skeleton data

An oral defense of this thesis took place on December 11th, 2024 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	**Dr. Mehran Ebrahimi**
Research Supervisor	**Dr. Faisal Qureshi**
Examining Committee Member	**Dr. Amirali Salehi-Abari**
Thesis Examiner	**Dr. Andrew Hogue**

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

In this work, we propose a deep learning architecture, incorporating a Graph Convolutional Network (GCN) backbone combined with a partitioning transformer, that achieves results comparable to the state-of-the-art methods in skeleton based multi-person, multi-view human action recognition. By leveraging attention-based GCN, the model captures context-dependent intrinsic topology while enhancing discriminative information. Furthermore, utilizing transformers, we harness their ability to aggregate long-range temporal information, allowing us to learn complex actions by attending to both short-term and long-term temporal windows. This is achieved through our partitioning strategy, which efficiently captures the relationships between neighboring and distant joints, enabling a comprehensive understanding of human movement dynamics. In this work, we also introduce a Cosine-based noise as a new data augmentation strategy for joints across time. This helps improve our model, Hybrid-Graformer, achieve accuracy comparable to the state-of-the-art across various skeleton-based action recognition benchmarks.

Keywords: Graph Convolutional Networks; Transformers; Action Recognition; Skeleton-based Action Recognition

Statement of Contributions

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am sole source of the creative works and/or inventive knowledge described in this thesis.

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize the Ontario Tech University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize the Ontario Tech University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Muhammad Salik Nadeem

Acknowledgements

First and foremost, I am thankful to God Almighty for the opportunities and blessings that He has and continues to bestow upon me.

I would like to express my gratitude to Dr. Faisal Qureshi for giving me the opportunity to pursue my Master's degree with VCLab at Ontario Tech University. His support and insightful guidance throughout this journey have been invaluable. It would not have been possible to navigate the complexities of my research without the environment that Dr. Qureshi fostered. I am grateful for his contribution to my intellectual and academic growth.

Next, I extend my deepest thanks to my parents, Nadeem Afzal and Fauzia Nadeem, whose sacrifices and encouragement I can never fully repay. My mother always believed in me and instilled the confidence to chase my dreams while my father ensured I had access to the best opportunities and resources. Their love and unwavering support have profoundly shaped my life and I am eternally grateful to God for blessing me with the best parents. Another special mention here goes to my aunt, Shaista Vine, who has been like a second mother for me. Her contributions and sacrifices for me and my family are immeasurable.

I am especially grateful to my loving wife Anika Javed, who has been my steadfast foundation ever since she became a part of my life. I can never thank her enough for motivating me to spend countless hours and nights working on my research while she took care of everything else. Her presence in my life has been the shinning light that has kept me going and allowed me to complete this program.

Lastly, I want to express my heartfelt gratitude to my friend Tony Joseph, whom I met through VCLab. His sincere friendship, support, and guidance have been instrumental in keeping me motivated and pursuing my research. I am genuinely thankful for finding a great friend through this program.

Contents

List of Tables

List of Figures

List of Equations

List of Notations

Symbol	Description
Scalars	
a, b, c	Scalars (lowercase)
α, β, γ	Scalars (Greek letters)
n	Number of training samples
Vectors	
$\mathbf{v}, \mathbf{w}, \mathbf{x}$	Vectors (bold lowercase)
$\mathbf{1}, \mathbf{0}$	Ones vector, zeros vector
Matrices	
$\mathbf{X}, \mathbf{W}, \mathbf{A}$	Matrices (bold uppercase)
$\mathbf{I}_n$	Identity matrix of size $n \times n$
$\mathbf{1}_{n \times m}$	All ones matrix of size $n \times m$
$\mathbf{X}_{i,j}$	Element in i^{th} row and j^{th} column of $\mathbf{X}$
$\mathbf{X}_{:,j}$	j^{th} column of matrix $\mathbf{X}$
$\mathbf{X}_{i,:}$	i^{th} row of matrix $\mathbf{X}$
Sets	
$\mathbb{R}, \mathbb{N}, \mathbb{Z}$	Real numbers, natural numbers, integers
$\mathbb{R}^{n \times m}$	Set of real-valued $n \times m$ matrices
$\mathbb{R}^n$	Set of real-valued n-dimensional vectors
Linear Algebra	
$\mathbf{X}^T$	Transpose of matrix $\mathbf{X}$
$\mathbf{X} \circ \mathbf{Y}$	Hadamard product of matrices $\mathbf{X}$ and $\mathbf{Y}$
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$

Continued on next page

Symbol	Description
Graphical Notations	
$\mathcal{G}$	Graph (nodes and edges)
$\mathbf{A}_{ij}$	Adjacency matrix element representing connection between nodes i and j
Probability and Statistics	
$p(x)$	Probability density function of x
$\mathbb{E}[X]$	Expectation of a random variable X
$\mathbb{P}(A)$	Probability of event A
σ^2	Variance of a random variable
Deep Learning	
$\mathbf{W}, \mathbf{b}$	Weights and bias terms
$f(\cdot)$	Activation function
L	Loss function
$\mathbf{y}_i, \hat{\mathbf{y}}_i$	Ground truth and predicted labels
θ	Parameters of the model

Chapter 1

Introduction

Humans can effortlessly interpret actions in videos, even in the face of challenges such as blurriness and occlusion. However, developing action recognition models that can replicate this level of understanding has remained a central challenge in computer vision [2, 49, 55, 111, 138], particularly with the exponential growth of video content creation driven by social media. In light of this growing trend, human action recognition models hold tremendous value and find applications in numerous important areas, including video indexing, surveillance, intelligent agents, and more.

Videos represent 3D actions as 2D projections of a 3D world, accurate action recognition hinges on a robust model and comprehensive feature representation of human actions. Incomplete or inadequate representations can lead to incorrect predictions. To effectively understand human actions, a model must not only capture the representation of the action itself but also account for varying camera conditions (whether stationary or moving), recognize different body shapes and sizes, and generalize across individuals performing the same actions.

Given the complexity of the problem, numerous approaches have been proposed to tackle this challenge, accompanied by the release of various benchmark datasets to support the research community. In the following chapters, we discuss these approaches and

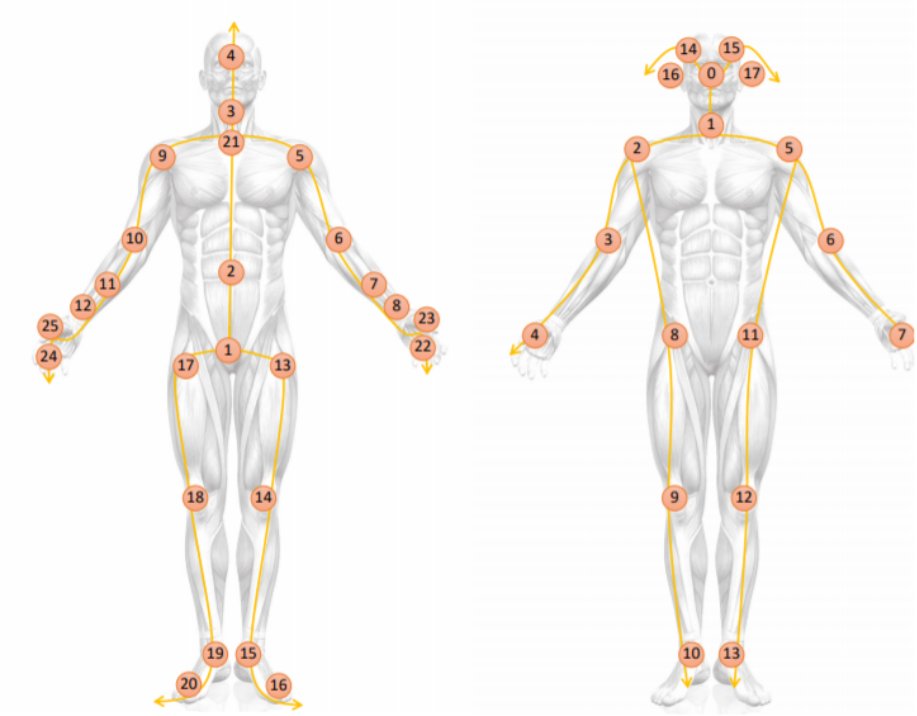

Figure 1.1: Illustrations of human skeleton graphs from two datasets. NTU RGBD dataset (left) and Skeleton-Kinetics dataset (right). Figure adapted from [99].

the datasets used to advance action recognition research.

1.1 Human Action Recognition

Since the early 2010s, deep learning-based approaches have achieved unprecedented success in many long-standing computer vision tasks, such as human pose estimation, object detection, action recognition, action prediction, and action interpolation. The availability of large-scale labeled datasets, alongside advancements in computational resources, particularly the development of faster and more efficient GPUs, has significantly contributed to the increased accuracy and robustness of machine learning models. In particular, computer vision has greatly benefited from the application of Convolutional Neural Networks (CNNs), which, when trained on vast amounts of labeled data, have produced results comparable to human-level accuracy across many challenging problems. Human action

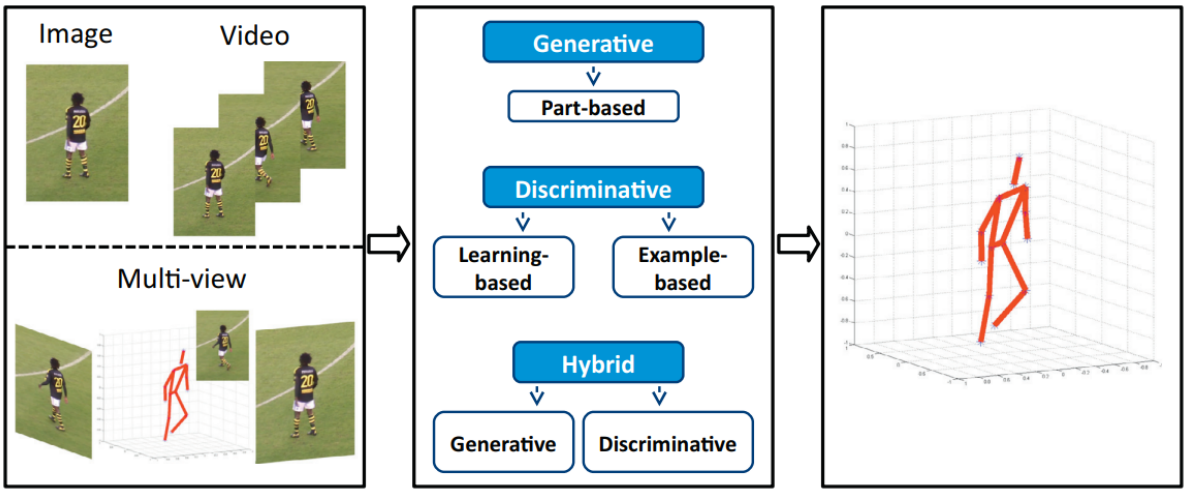

Figure 1.2: Taxonomy of 3D pose estimation methods. These methods, applied to images or videos in monocular or multi-view setups, are categorized into generative (including part-based models), discriminative (divided into learning-based and example-based), and hybrid approaches, which combine aspects of both generative and discriminative methods. Figure adapted from [93].

recognition is one such domain that has seen substantial progress. Various approaches to this problem utilize different input modalities, including RGB image sequences, depth maps, 2D or 3D skeleton joints, bones and body parts, dynamic images, and even infrared camera frames.

In this work, we focus on 3D skeleton data, as shown in Figure 1.1, as the input modality for human action recognition, which offers numerous advantages. First, 3D skeleton data provides a compact representation of human motion, significantly reducing dimensionality compared to RGB or depth images. It preserves essential information about body pose and movement. This compactness facilitates more efficient model training and reduces the risk of overfitting. Additionally, 3D skeleton data captures the full range of motion in a detailed and expressive manner, allowing even subtle movements to be accurately represented. Another advantage is that 3D skeleton-based models tend to be view invariant, as they focus on the relative positions of joints rather than appearance

Figure 1.3: Examples of action video frames utilized in computer vision research: a) single person's action; b) human interaction in surveillance footage; c) entertainment videos; d) autonomous driving applications; e) human-robot interactions; f) health-care applications. Figure adapted from [55].

features, making them more robust to changes in camera angles and background clutter. Furthermore, skeleton data is less sensitive to variations in lighting or occlusions, which are common challenges when working with RGB or depth data. These properties make 3D skeleton data an ideal choice for human action recognition, particularly in environments where generalization across subjects and viewpoints is critical.

Moreover, in scenarios where 3D skeleton data is not readily available, the development of highly efficient, real-time 3D pose estimation algorithms [93, 152] provides an effective alternative, as shown in Fig 1.2. These algorithms can accurately extract 3D skeletal data from any RGB video sequence, enabling the use of 3D skeleton-based models without the need for dedicated motion capture systems. This advancement ensures that researchers and practitioners can easily obtain high-quality 3D skeleton data from standard RGB videos, further broadening the applicability of skeleton-based action recognition methods.

1.2 Applications

Human action recognition algorithms have many applications, including surveillance, video retrieval, entertainment, human-robot interaction, healthcare, and autonomous driving as shown in Figure 1.3. The proliferation of video data, coupled with advances

in computational power and sophisticated algorithms, has catalyzed remarkable progress in these fields. The evolution of deep learning and artificial intelligence (AI) has played a pivotal role in driving these innovations.

Visual surveillance, where security concerns have become increasingly relevant in the contemporary life, can be automated using action recognition algorithms. Surveillance systems, often designed to monitor permissible and prohibited behaviors, benefit from action recognition algorithms. By integrating these algorithms with a network of cameras, such systems can enhance the detection and prevention of criminal activities. The mere presence of cameras, supported by action recognition, also contributes to a heightened sense of security.

Beyond surveillance, video retrieval faces growing challenges with the rapid increase in online video content. Traditional retrieval methods rely heavily on textual data like tags, titles, and descriptions, which are often inaccurate or irrelevant. In contrast, analyzing human actions within videos offers a more reliable alternative. This method focuses on ranking videos based on the relevance of detected actions, offering a more effective means of retrieval compared to simple classification tasks.

In the realm of entertainment, the gaming industry has witnessed a surge in popularity, particularly with games incorporating full-body interaction. Games such as dance and sports simulations rely on affordable RGB-D sensors that capture both color and depth information. This depth data provides crucial structural information, enabling precise action recognition by reducing motion variation within classes and filtering out background noise, leading to a more immersive gaming experience.

The integration of human action recognition is also transforming human-robot interaction. Effective communication between humans and robots is essential, whether in domestic or industrial settings. For instance, tasks such as "passing a cup of water" or "assembling an object" require visual communication for robots to interpret and respond accurately. Action recognition algorithms enhance this interaction, ensuring that robots

understand human actions effectively.

Finally, autonomous driving has become another critical domain for action prediction algorithms. These algorithms allow vehicles to anticipate pedestrian movements and future actions, which is vital for collision avoidance. By analyzing motion characteristics early in an action sequence, autonomous vehicles can predict actions without needing to observe the entire sequence, thereby improving safety on the roads.

1.3 Research Challenges

Despite significant advancements in human action recognition, the state-of-the-art algorithms still face difficulties in accurately classifying all actions. While deep learning approaches can mitigate many of these challenges given enough data, it is impractical to have datasets that encompass every possible variation in human actions. This has driven researchers to develop generalized models that can learn complex human body structures and movements, while also capturing interactions across time, varying angles, and different environments.

One of the major challenges is intra-class variation as shown in Figure 1.4, where the same action can be performed differently by different individuals. For instance, a common action like "reading" may look different for each person—some might have a different pose, some might sit while reading, some might stand. Moreover, variation in viewing angles exacerbates the problem, as the same action appears distinct when observed from multiple angles. Real-world videos are not recorded under consistent conditions, meaning actions need to be recognized regardless of the camera position. Additionally, pose variations between individuals performing the same action can introduce both subtle and pronounced differences, further complicating classification tasks.

Another significant challenge in human action recognition is inter-class variation, as illustrated in Figure 1.5. Actions from different categories can often appear strikingly

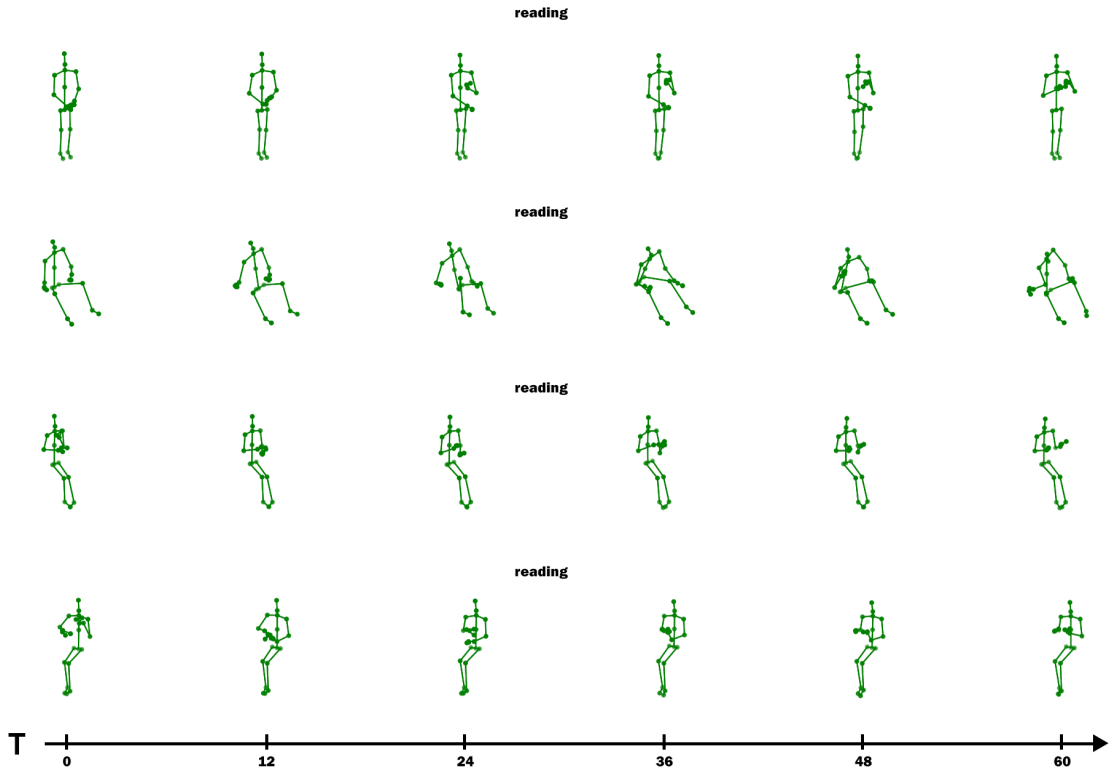

Figure 1.4: Intra-class variations: Examples of actions which look very different but belong to the same class.

similar, making differentiation difficult. For instance, distinguishing between "wiping face" and "sneezing/coughing" poses a challenge due to their shared motion patterns. Similarly, in human-object interaction tasks, actions such as "drinking" and "eating" involve comparable poses and gestures, yet they represent distinct interactions belonging to separate classes. This overlap can confuse models, particularly when relying solely on skeletal data.

Moreover, models also struggle with cluttered backgrounds and camera motion. While action recognition systems work well in controlled settings with static backgrounds, fixed cameras, and few obstructions, real-world scenarios are more challenging. Videos often include moving cameras, busy environments, and multiple subjects, making accurate action recognition more difficult.

Another critical challenge is the lack of annotated data. Given the extensive range

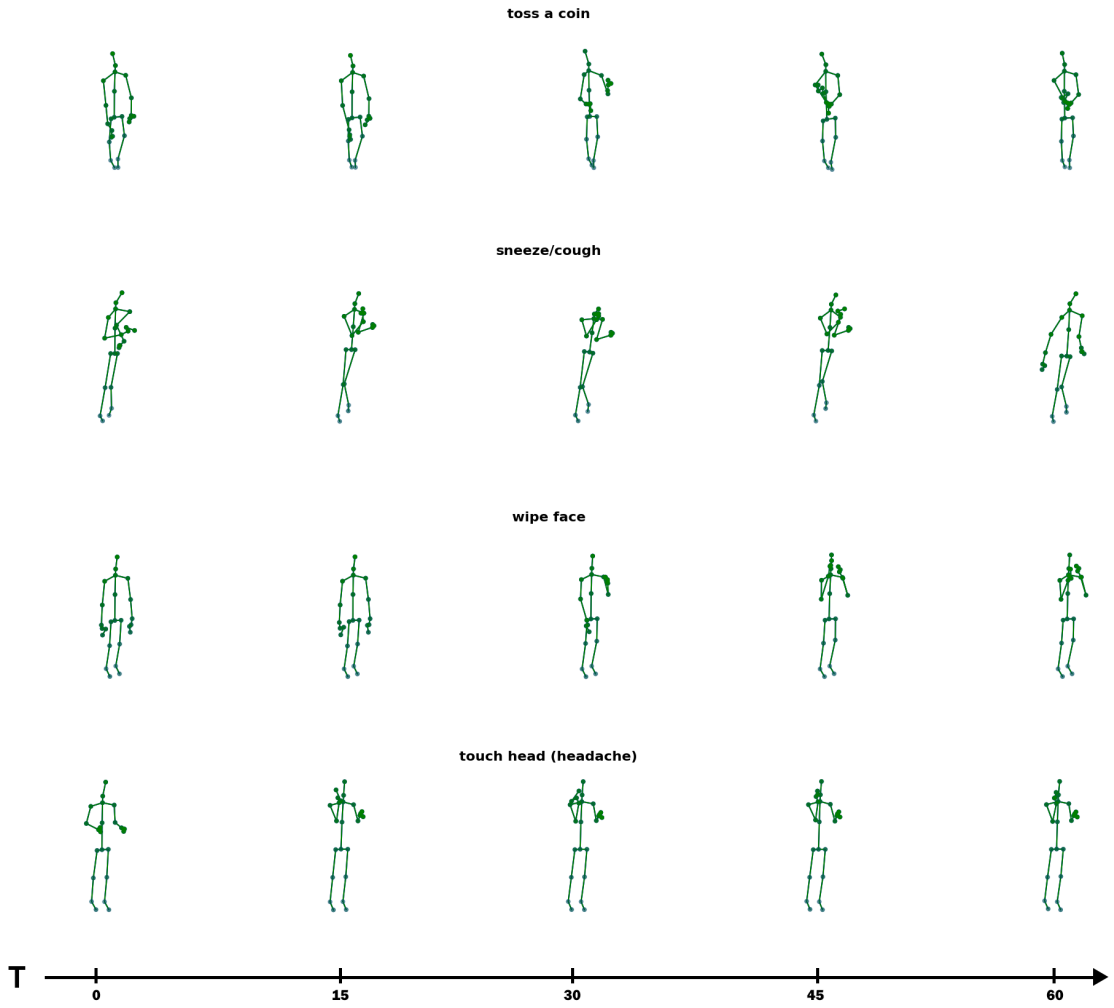

Figure 1.5: Inter-class variations: Examples of action classes which look very similar but belong to different classes.

of possible action classes and the inherent complexity of human motion, deep learning models require large volumes of labeled data to perform effectively. However, obtaining and accurately annotating such vast datasets can be both time-consuming and challenging. As a result, leveraging both labeled and unlabeled data is essential for improving model performance.

Additionally, uneven distribution of distinguishing features across video frames further complicates action recognition. Not all frames provide equal amounts of useful information; some frames are significantly more discriminative than others. The distribution

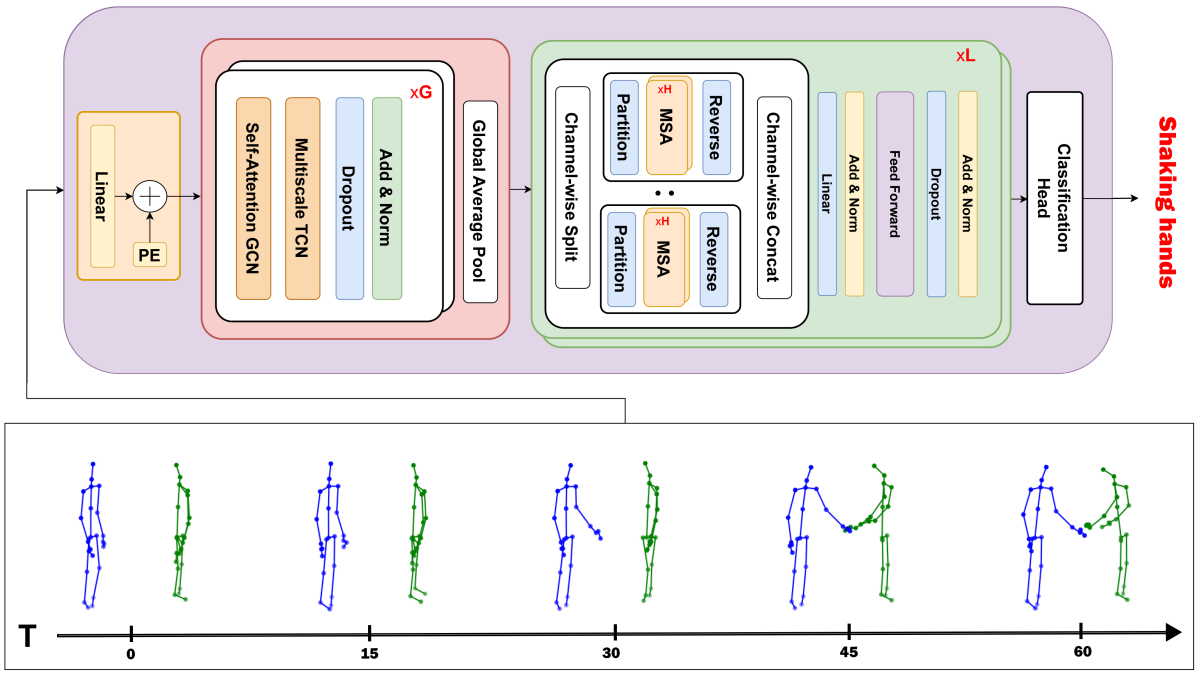

Figure 1.6: Overview of Hybrid-Graformer model: 3D skeleton joints are taken as input and the output is the class prediction label.

of discriminative motion can generally be categorized into four types. First, there are actions where discriminative motion is distributed throughout the entire action without any object interaction, such as "standing up" or "shaking hands." Second, certain actions exhibit discriminative motion only in short bursts, even without object interaction, like "sneezing" or "nodding." Third, some actions involve continuous motion with object interaction, such as "taking off a jacket." Lastly, certain actions have discriminative motion that occurs in brief moments but involves object interaction, like "eating" or "drinking." This uneven distribution of discriminative information necessitates that models learn to focus on the right frames to accurately classify actions.

Addressing these challenges is crucial for researchers seeking to develop more accurate and generalizable action recognition models. By refining methods to account for inter-class and intra-class variation, enhancing robustness against environmental complexities, and optimizing frame attention strategies, the field of human action recognition can

advance significantly. Ultimately, overcoming these obstacles will lead to models that perform effectively in diverse and dynamic conditions, opening up new possibilities for applications in areas such as surveillance, human-computer interaction, and intelligent systems.

1.4 Contributions

In this work, we introduce a novel hybrid architecture, illustrated in Figure 1.6, that combines a GCN backbone with a partitioning transformer for action classification using skeleton data. The main contribution in the architecture is the configuration we have used to construct the hybrid model which we describe in the following chapters. Our final model is trained on multiple modalities, including joints and bones, and we employ an ensemble approach to present our best results. The key contributions of this work are as follows:

1. We propose a novel configuration for a Hybrid-Graformer model for skeleton-based action recognition, combining a GCN backbone with a transformer.

2. We introduce a novel data augmentation technique called ‘Cosine-based noise generation,‘ which outperforms traditional random noise methods.

3. We employ a multi-modal ensemble model that processes joints, bones, position, and motion data.

4. We evaluated our Hybrid-Graformer model on the NTU-RGBD-60, NTU-RGBD-120, and NW-UCLA benchmark datasets, achieving results comparable to state-of-the-art methods [126, 153, 24].

1.5 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 reviews relevant literature, highlighting key approaches in the field and contrasting them with our proposed method. Chapter 3 covers background material required for understanding our approach. Chapter 4 discusses our proposed architecture, breaking down each component and the datasets used. Chapter 5 presents the results and analysis, while Chapter 6 concludes the thesis with a discussion of the implications and future directions.

Chapter 2

Related Works

This chapter covers prior work done in human action recognition. This problem has been tackled using a variety of approaches. We will also cover some of the earlier shallow approaches. The most recent successful ones can be divided into four groups depending on their core building block: Convolutional Neural Networks (CNN), Graph Neural Networks (GCN), Transformers and finally Hybrid Deep Neural Networks (DNN) with combination of multiple building blocks.

2.1 Shallow Approaches

Shallow methods, such as the use of Histogram of Oriented Gradients (HOG) or Scale-Invariant Feature Transform (SIFT) in computer vision tasks, depend significantly on domain knowledge to extract features relevant to the specific problem. These approaches typically employ algorithms, such as Support Vector Machines (SVMs) or decision trees, which can perform adequately on smaller datasets but are limited in their ability to generalize to high-dimensional data or more complex tasks.

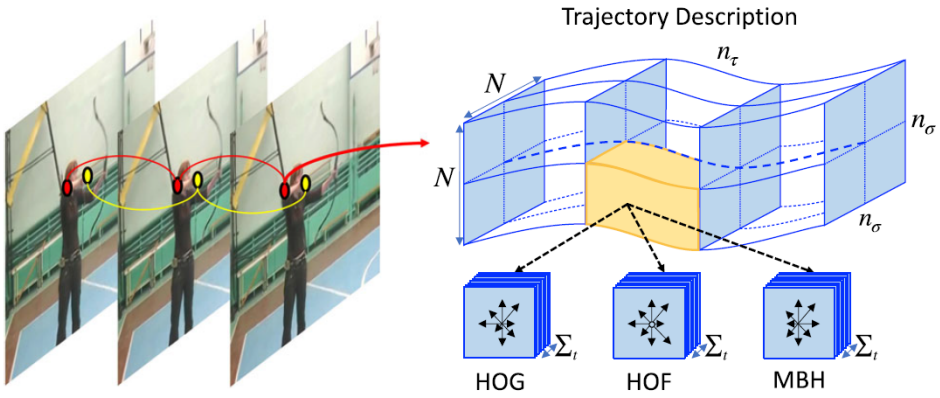

Figure 2.1: Point trajectories are tracked over frames, and are described by HOG, HOF and MBH features. Figure adapted from [124].

2.1.1 Action Representation

The primary challenge in action recognition is to effectively represent an action in a video given the variability in motion speed, camera angles, appearance, and pose. An action representation must be computationally efficient, effectively capture the characteristics of actions, and maximize the distinction between different actions to minimize classification errors. One of the key difficulties is handling large variations in appearance and pose within the same action category, which complicates recognition. The objective of action representation is to transform an action video into a feature vector that captures the most representative and discriminative information, thereby reducing variability and improving recognition accuracy. Action representations can be broadly divided into global and local features. This section focuses on traditional hand-crafted action representation methods, where the parameters are predefined by experts, as opposed to deep learning methods that automatically learn from data.

Human actions in videos generate space-time shapes within a 3D volume, capturing both spatial and dynamic information of the human body. Holistic representation methods are designed to capture the overall motion information of the entire human subject, providing rich data for action recognition. However, these methods are sensitive to

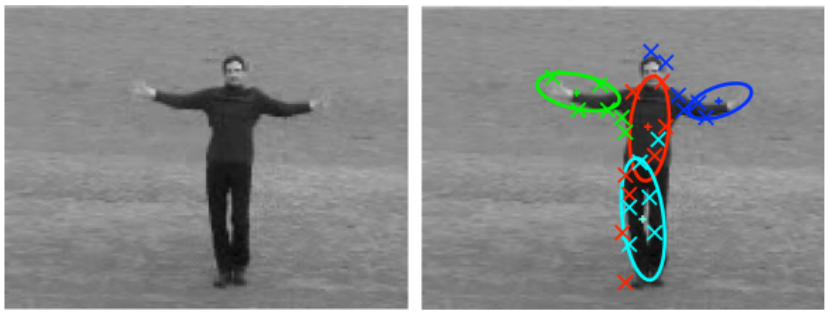

Figure 2.2: Example of body parts detected by the constellation model by Niebles and Fei-Fei [80].

noise, often introducing irrelevant information from both the subject and the background. Some techniques, like those developed by Gorelick et al. [37] and Blank et al. [8], use the Poisson equation to extract shape properties for action representation and classification. Alternately, motion information can be computed using optical flow algorithms, which analyze motion between consecutive frames. These methods have been used to describe features of human bodies and their parts.

Local representation methods focus on identifying regions with significant motion information, avoiding the noise issues of holistic representations. Methods like space-time interest points (STIPs) [57, 58] and motion trajectories are robust against variations in translation and appearance. STIPs detect motion changes in both spatial and temporal dimensions, while other methods use local motion information to generate feature vectors for action recognition. However, STIPs only capture short-term information, making it necessary to track interest points over time to capture long-duration motion. Feature trajectories are one way to achieve this, using techniques like Harris3D interest points with a Kanade-Lucas-Tomasi (KLT) tracker or matching Scale-Invariant Feature Transform (SIFT) points across frames. These trajectories can be described using various features, such as Histogram of Oriented Gradients (HOG) [18], Histogram of Optical Flow (HOF) [60], and Motion Boundary Histogram (MBH) [123], to represent complex human activities and reduce the effects of camera motion as shown in Figure 2.1.

2.1.2 Action Classifiers

Human action classifiers employ various strategies for categorizing actions in videos based on the computed representations. One of the most straightforward methods is direct classification, where the entire action video is summarized into a feature vector and classified into predefined categories. Common techniques include support vector machines (SVMs) [94], k-nearest neighbors [7, 59], and bag-of-words models [61, 123]. However, bag-of-words models, while useful for basic text representation, fail to capture temporal or structural dependencies, which significantly limits their effectiveness in handling more complex and dynamic scenarios.

A more refined strategy is sequential modeling, where the temporal evolution of actions is captured by treating videos as sequences of frames. Techniques like Hidden Markov Models (HMMs) [26, 88] and Conditional Random Fields (CRFs) [107, 127] fall under this category. While these approaches are effective in temporal representation, they often struggle with background noise and challenging datasets, making their performance less robust in real-world scenarios.

Another category of methods is space-time approaches, which focus on spatiotemporal correlations between local features [60, 74]. These methods improve over direct classification by modeling the distribution of interest points across both space and time, thereby capturing more dynamic information about the action. Part-based approaches take a different angle by concentrating on the motion of specific body parts [28, 80, 129, 130]. These methods capture the geometric relationships between different body parts as shown in Figure 2.2, helping to distinguish between various actions by focusing on structured motion patterns. They are especially helpful in situations where distinguishing between actions depends on small differences in how body parts move.

Manifold learning approaches [127, 46] tackle the challenge of high-dimensional representations by reducing them to lower-dimensional spaces. These methods capture the nonlinear structures of actions by representing human silhouettes in a compact form,

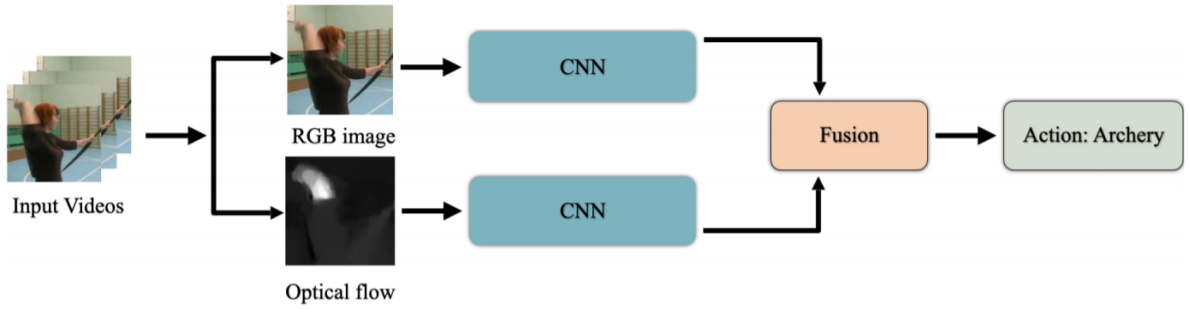

Figure 2.3: A simplified high-level structure of a two-stream CNN network. The two networks separately capture spatial and temporal information and are then fused together. Figure adapted from [111].

which can significantly enhance recognition performance in complex environments. Another important category is mid-level feature approaches [17, 69], where hierarchical models learn mid-level features from low-level representations. These techniques act as a bridge between raw features and high-level action recognition, though they often require additional annotations to perform optimally. Lastly, feature fusion approaches [72, 142] focus on combining multiple types of features for a more holistic action classification. By considering the inter-relationships between various features, these methods improve classification accuracy by leveraging complementary information from different aspects of the video data.

Each of these approaches has distinct strengths and limitations—some excel at capturing temporal dynamics, while others are better suited for handling spatial relationships or fusing multiple features. However, even the most sophisticated traditional methods pale in comparison to the capabilities of deep learning-based classifiers, which offer superior representation and classification ability for human action recognition tasks.

2.2 Deep Approaches

Deep learning methods have shown significant success in human action recognition tasks. Convolutional Neural Networks (CNNs) are widely used for extracting spatial features,

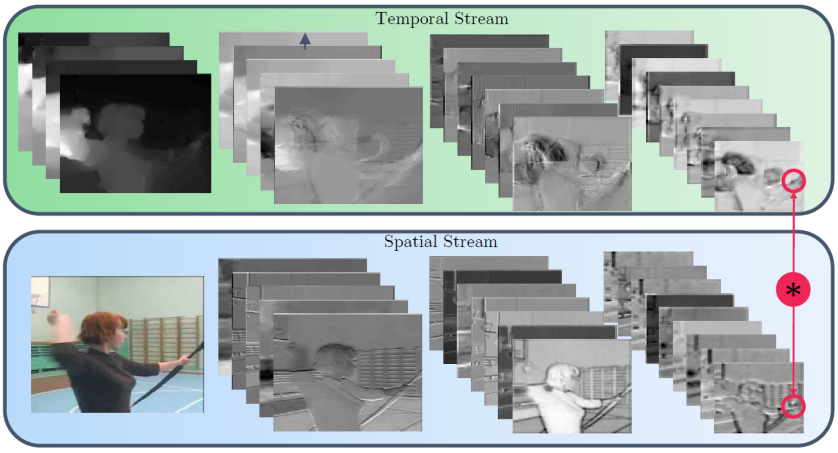

Figure 2.4: Example outputs of the first three convolutional layers from a two-stream ConvNet model [106]. The two networks separately capture spatial (appearance) and temporal information at a fine temporal scale. Figure adapted from [29].

with specialized variants such as Two-Stream CNNs capturing both spatial and temporal information, and 3D CNNs extending this to spatio-temporal representations. CNNs combined with Recurrent Neural Networks (CNN-RNN) enhance temporal sequence modeling, while dynamic image-based approaches encode motion information into single images. Graph Convolutional Networks (GCNs) are effective for modeling skeletal structures in non-Euclidean spaces, and Transformers leverage attention mechanisms for capturing long-range dependencies. Hybrid models combine these architectures to leverage their complementary strengths.

2.2.1 Two-Stream CNNs

The two-stream architecture [106] is inspired from the way human eyes process visual data as shown in Figure 2.3. There is a slower stream which focuses on spatial data and a faster stream which reads motion data. These models generally have two CNNs which learn spatial and temporal features from RGB and stacked optical flow, respectively. Several strategies have been proposed in [29, 30] for fusing the outputs of both streams together in order to make action classification.

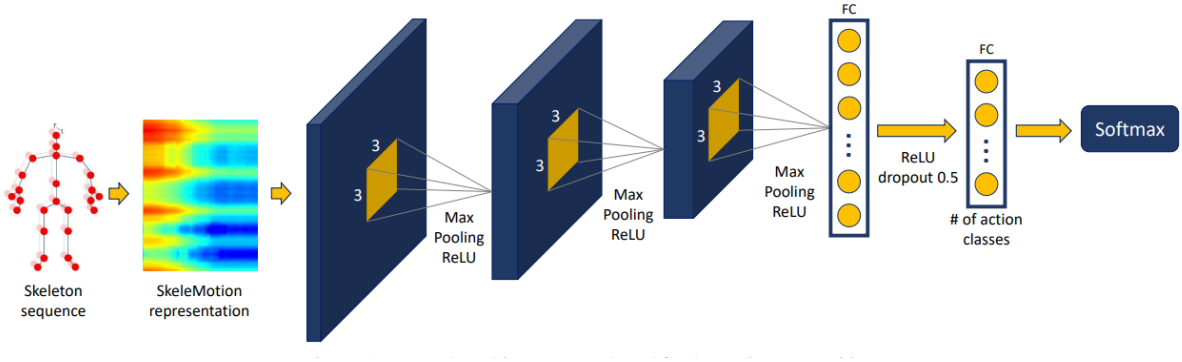

Figure 2.5: Skeleton sequences can be converted to 2D pseudo-images and then be fed to 2D CNNs for feature learning. Figure adapted from [9].

These architectures do a good job in capturing spatial and short-term temporal features but generally struggle with long-term temporal information which is critical for certain action classes. For example "making a phone call" and "salute" might have a similar motion in between but a key distinction is taking the phone out in the initial frames then taking the hand up to the ear with the phone. There are cases where having the model learn long-term temporal features helps in distinguishing otherwise similar looking actions. In order to get over this limitation, Wang et al. proposed a temporal segment network to take sparsely sampled frames from a video sequence during training and classification scores of the sampled frames are aggregated to a final one in testing [106]. However, since these frames are processed independently, there is no long-term temporal relationship captured in the representations as shown in Figure 2.4.

For most of the networks in these two stream approaches, optical flow has been dominantly used for capturing temporal motion representations which is computationally expensive. [31] use the same RGB stream as input for the temporal stream by reducing the resolution and increasing the frame rate as compared to the spatial network which effectively results in the CNN capturing motion patterns instead of spatial structure. [146, 156] use motion vectors that are extracted from RGB video instead of optical flows which significantly reduces the computational overhead for calculating optical flows.

2.2.2 3D CNN

3D CNNs first introduced in [45] are a direct extension of 2D CNNs by convolving and pooling in the temporal dimension as well as shown in Figure 2.5. Tran et al. [116] introduced the C3D architecture which extracts spatial and temporal representations. Carreira and Zisserman [11] combined both two-stream and 3D CNNs together to create a new architecture called I3D which outperformed both 3D CNNs and previous two-stream networks by using the inflation of 2D kernels pretrained on ImageNet to 3D. Yet the parameters of the spatial and temporal filters continue to increase which results in an increase in the complexity and memory usage in 3D CNNs. Different methods, however, were applied to reduce these problems. For instance, a 2D spatial and 1D temporal convolution can be used instead of a 3D convolution kernel [109, 86, 137, 117].

Li et al. [65] proposed a CNN-based framework for action recognition, treating skeleton sequences as 3D tensors akin to image data. They also introduced a skeleton transformer to optimize joint ordering. Tasnim et al. [114] introduced a Deep Convolutional Neural Network (DCNN) model designed to train feature vectors derived from joint coordinates along the x, y, and z axes. Each frame's joints, indexed by j, are represented as $f_i(x_{ij}, y_{ij}, z_{ij})$, where i denotes the frame number. It must be kept in mind, however, that only short temporal information can be encoded as every 3D convolution generally covers small temporal windows instead of the entire video. The 3D CNN approach while very effective, suffers from the same limitation as two-stream networks where they fail to capture long-term temporal relations.

Kim and Reiter [52] adopted Temporal Convolutional Networks (TCN) for human action recognition, presenting the Res-TCN model, which explicitly learns the spatio-temporal representation of skeleton data. The model accepts temporally concatenated frame-wise skeleton features spanning the entire sequence as input. Skip connections and 1D convolution filters are employed to glean spatial and temporal dependencies from the input data. Duan et al. [24] devised a PoseConv3D model utilizing 3D-CNNs to capture

the spatio-temporal dynamics of skeleton sequences. This model operates on 3D heatmap volumes as input, where pseudo heatmaps for joints and limbs are generated, serving as effective inputs for 3D-CNNs.

While CNN-based methods emphasize spatial and temporal joint relationships, they often rely on domain knowledge for joint ordering, highlighting the need for methodologies that can implicitly learn these relationships. Additionally, only short temporal information can be encoded as every 3D convolution generally covers small temporal windows instead of the entire video. The CNN based approach while very effective, fail to capture long-term temporal relations.

2.2.3 CNN Plus RNN

RNNs are a good alternative for CNNs when dealing with sequential data as they have hidden states which take information from previous states as well. However, vanilla RNN suffers from the vanishing gradient problem which makes it unsuitable for longer sequences. Hence, most RNN-based solutions use a gated model such as Long-Short Term Memory (LSTM) which do not suffer from vanishing gradients and effectively capture long term temporal relationships between data. Most LSTM-based approaches for action recognition use a cascade of CNN and LSTM. The CNN effectively captures spatial features while the LSTM captures the temporal dependancy between frames. Donahue et al. [20] suggested the Long-Term Recurrent Convolutional Network (LRCN). They essentially proposed to stack CNNs and LSTMs wherein the LSTM combines the frame-level characteristics extracted by 2D CNNs to capture spatial and temporal relationship as shown in Figure 2.6. An LRCN with a soft-attention model was proposed by Sharma et al. [96] in order to assign higher weights to important frames for better spatiotemporal learning.

Despite the LSTM's ability to capture temporal dependence, it is not capable to understand the intuitive high-level spatial and temporal relationship. Even though Jain

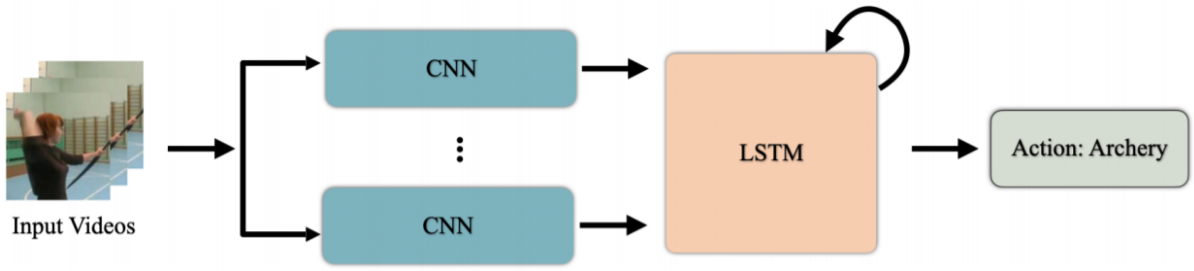

Figure 2.6: A simplified high-level structure of a CNN + RNN network. Figure adapted from [111].

et al. combined the spatiotemporal graphs and an RNN in [44] to obtain the spatio-temporal-structural information, it fails to model the motion-dynamics between the spatial correlation and frames simultaneously by directly applying LSTM to video-based action recognition. Shi et al. [101] introduced the Convolutional-LSTM (ConvLSTM) which replaces 1D vector multiplications with 2D vectors and convolutional kernels, applying spatial encoding in LSTM cells. This allows an LSTM to encode both spatial and temporal features more effectively. Zhu et al. [154] implemented a cascaded 3D CNN with ConvLSTM using both RGB and depth modalities as input and fusing the results together to make predictions. This results in better spatiotemporal encodings as 3D CNN encodes short-term encodings and the LSTM encodes the long-term relationships.

Certain methods such as [73, 67] use a CNN+ConvLSTM encoder to learn a representation of action data before doing classification. These methods do a good job at learning short-term and long-term temporal relations, they also learn features that are invariant to views effectively as representations of similar action features are pushed closer together in the latent space feature vector. However, despite their strengths, these methods fail to capture global temporal relationships in longer action sequences and even for spatial relationships they sometimes fail to capture the nuances of a human body, not capitalizing on the constraints naturally present in the human body. They also are not robust against changes in views.

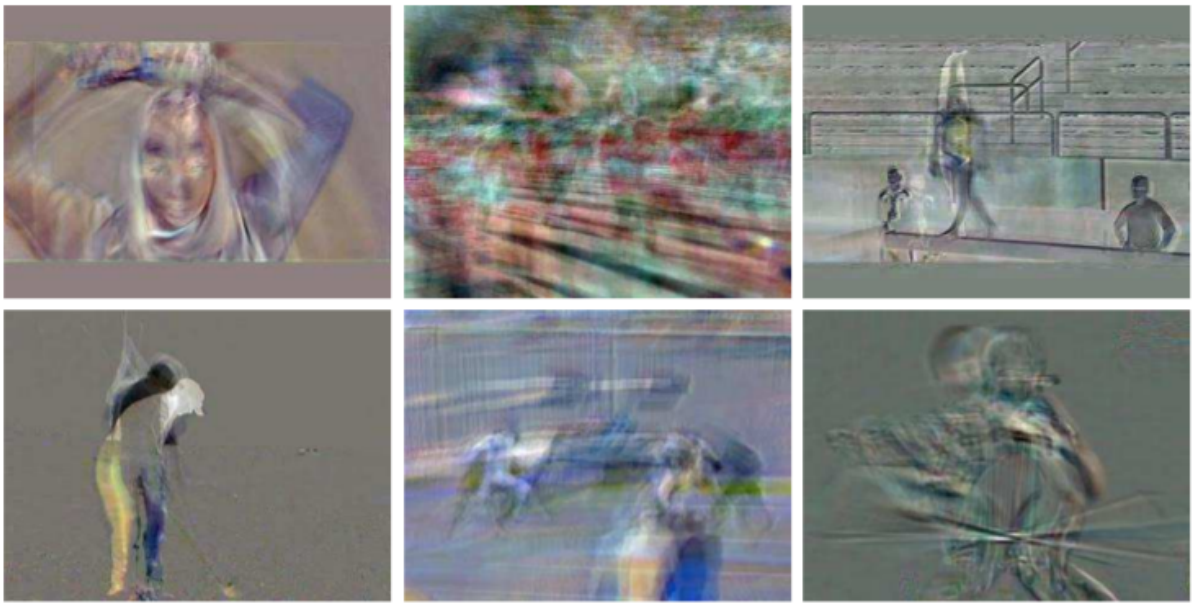

Figure 2.7: Dynamic images summarizing the actions and motions that happen in images in standard 2d image format. Figure adapted from [6].

2.2.4 Dynamic Images

The dynamic image-based (DI) approach attempts to encode spatiotemporal information by turning a video into one or multiple dynamic images. It then applies a CNN to do action classification. The approach proposed by [6] involves using rank pooling to transform a video into one or more dynamic images, as illustrated in Figure2.7, and then fine-tuning models pre-trained on ImageNet [91] using these images.

DI-based methods suffer from a similar limitation as two-stream networks and 3D CNN-based approaches where they only encode short-term temporal relationships. A rank pooling scheme that uses hierarchy to encode a video at multiple levels was introduced in [32] with the objective to enhance the ability of DIs to encode long-term temporal dependency. Through this approach, a video is divided into various overlapping video segments. Using rank pooling, each segment is encoded to produce a sequence of DIs. Rank pooling is applied again to multiple sub-sequences of the DI sequence. This practice aims to model high-order dynamics.

2.2.5 Graph-based approaches

Graphs are a common way to represent network structures across diverse fields like social networks, biology, knowledge systems, and information networks and have been used extensively in many models and algoriths [2, 12, 38, 135]. Graph learning methods typically extract key features from graphs using machine learning techniques. There are four main approaches: graph signal processing, matrix factorization, random walk, and deep learning.

Graph Signal Processing (GSP) extends traditional signal processing to graph structures, where data resides on nodes and edges define relationships [102]. GSP utilizes a graph shift operator (GSO), commonly represented by either the adjacency matrix A or the graph Laplacian $L = D - A$, where D is the degree matrix. The Laplacian matrix is particularly useful for capturing graph smoothness, a key property in GSP [81]. In GSP, signals on a graph are processed by transforming them into the graph spectral domain through the graph Fourier transform. This transform decomposes a signal x using the eigenvectors U of L, as $\hat{x} = U^T x$, which allows for filtering in the spectral domain [92]. Filters are often applied via polynomial approximations of the GSO, enabling efficient filtering without explicit eigen decomposition $H(L) = \sum_{k=0}^{K} h_k L^k$. This formulation underpins graph convolutional networks (GCNs), widely used in machine learning to leverage node relationships effectively [54]. GSP is powerful for analyzing data in complex, non-Euclidean domains, such as social and sensor networks, by enabling pattern extraction through graph-based filtering and spectral analysis.

Matrix factorization decomposes a matrix into lower-dimensional components, retaining key graph information like node relationships. In graph learning, it represents graph characteristics (e.g., vertex similarity) to produce vertex embeddings [155]. This method is effective for low-dimensional manifolds, preserving structural information while reducing dimensions. Graph Laplacian factorization preserves vertex similarity and supports transductive (training-only vertices) and inductive (new vertices) embeddings. Tech-

niques like Locality Preserving Projection (LPP) and its anchor-based variant AgLPP enhance this by capturing both local and global graph structures [41, 47]. Vertex proximity in a lower dimension, minimizing embedding error through methods like Singular Value Decomposition (SVD) and regularized Gaussian factorization [36, 3]. It's particularly suited for homogeneous graph data. Matrix factorization methods, though memory-intensive and limited in supervised tasks, are valuable for dimensionality reduction and structure preservation in graph learning.

Random walk-based methods are used for Network Representation Learning (NRL) by generating sequences of nodes while preserving relationships between them. These methods help in dimensionality reduction, particularly for networks with structural data. Graph-structured data encode important information in both graph structure and vertex attributes. Methods like `DeepWalk` [85] and `Node2Vec` [39] use random walks to generate sequences of nodes, treating vertices as words and using Word2Vec [78] for embedding. `Node2Vec` introduces a random walk strategy that balances breadth-first and depth-first sampling. The `LINE` [112] method, which maintains first- and second-order approximations, is another approach focused on large-scale networks. Incorporating vertex attributes (such as content or labels) enhances network representation. Models like `TADW` [140] and `MMDW` [119] use vertex information to improve embedding. `Struc2Vec` [33] focuses on structural similarity, while models like `Planetoid` [143] combine both network structure and vertex attributes. Heterogeneous networks consist of multiple vertex types and relationships. Methods like `HIN2Vec` [34] and `Metapath2Vec` [21] utilize meta paths or random walks to embed heterogeneous networks. These methods handle various relationships among vertices, improving embedding in social and knowledge networks. Time-varying networks evolve over time with new vertices and relationships. Models like `CTDNE` [79] and `HTNE` [157] incorporate temporal dynamics into embedding, capturing time-dependent changes in networks. `GraphSAGE` [40] generates embeddings for unseen vertices in dynamic networks by locally aggregating features from neighboring vertices.

Unlike methods that train embeddings for all vertices, it samples neighbors and updates vertex embeddings using various aggregators. However, it mainly focuses on local neighborhood information and does not directly capture higher-order proximity or community structure in graphs.

Graph-based CNNs or GCNs have seen great success in many tasks due to the expressive power of graphical representations. Human skeleton is naturally in the form of an acyclic graph as shown in Figure 1.1. Analyzing 3D skeletons with learning models have shown state-of-the-art performance in action recognition. CNNs and RNNs treat spatial features independently as vectors without taking into account how joints are connected together and form articulated movements. GCNs can make use of this information and learn features which are view-invariant and also capture both spatial and temporal dependencies. Hence, GCN-based action recognition has seen a lot of work [139, 103, 113, 150, 16, 13, 71, 64, 84, 42].

Graph Attention Networks (GATs) [122] incorporate attention mechanisms into graph neural networks to dynamically weight neighboring vertices. Gated Attention Networks (GAAN) [147] extend this with multi-head self-attention for vertex state updates. The Graph Attention Model (GAM) [63] is used for graph classification, adaptively selecting important vertices via an LSTM network. Attention Walks (AWs) [1] combine GNNs with random walks, using differentiable attention weights for co-occurrence matrix factorization.

Peng et al. [84] proposed an approach called ST-GCN (see Figure 2.8), that models graph sequences on the Riemann manifold using Poincare geometry features computed from spatial-temporal graph convolutional networks (GCNs). By training a Poincare model on multidimensional structural embeddings for each graph, they aimed to enhance feature learning by mixing dimensions for a more distinct representation. Yan et al. [139] developed Spatial-Temporal GCNs (ST-GCNs), that learn spatial and temporal data from skeleton data. Shi et al. [98] developed a two stream approach called two-stream

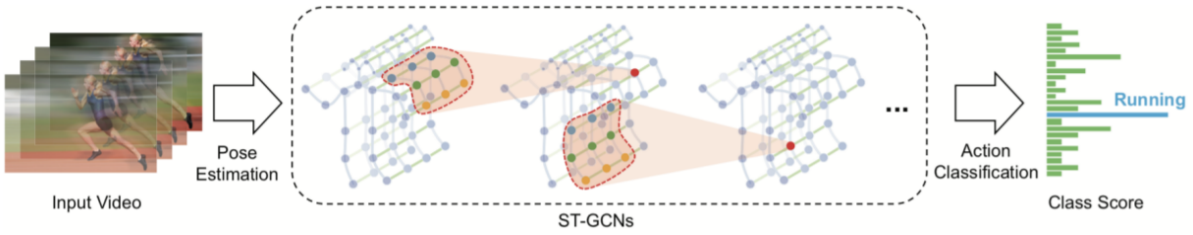

Figure 2.8: The joint dependency structure can naturally be represented via a graph structure, making them especially suitable for pose and action tasks. Figure adapted from [111].

Adaptive GCN (2s-AGCN) that learns the topology of the skeleton graph automatically by combining first order data (join coordinates) and second order data (length and direction of bones) in a two-stream fashion. In order to improve the discriminative power of GCNs attention models were also applied [104, 132]. Si et al. [104] combined GCNs with LSTMs which improves understanding the co-occurrence relationship between spatial and temporal domain.

Liu et al. [71] introduced G3D, a unified spatial-temporal graph convolution method aimed at effective feature learning. Employing a multi-scale aggregation scheme, G3D removes redundant dependencies between node features across different neighborhoods. Additionally, it incorporates graph edges in the "3D" spatio-temporal domain as skip connections to facilitate unobstructed information flow. Yang et al. [141] proposed PGCN-TCA to address several challenges encountered in previous GCN-based networks. This model integrates distant joint information, dynamic computation of adjacency matrices, and varying importance of frames and channels for action recognition, thus enhancing the overall performance. Chen et al. [13] introduced CTR-GC, a method that dynamically learns various topologies and efficiently aggregates joint features across multiple channels. This approach simultaneously learns shared topologies and channel-specific correlations, contributing to improved feature extraction and representation.

Lee et al. [64] introduced HD-GCN, featuring a hierarchically decomposed graph

(HD-Graph) and an attention-guided hierarchy aggregation (A-HA) module. HD-GCN aims to identify distant edges within hierarchy subsets and emphasize key hierarchy edge sets through attention-guided pooling and hierarchical edge convolution. Duan et al. [25] proposed DG-STGCN, leveraging learnable coefficient matrices for spatial modeling and dynamically diversified groups of graph convolutions and temporal convolutions for dynamic spatial-temporal modeling of skeleton motion. Hu et al. [42] introduced STGAT, designed to capture short-term dependencies using spatio-temporal modeling. STGAT reduces redundancy in local spatio-temporal features through the construction of local spatio-temporal graphs and dynamic relationship modeling. Chi et al. [16] proposed InfoGCN, with a learning framework combining an information bottleneck-based learning objective along with a classification loss. InfoGCN employs attention-based graph convolution to capture context-dependent topology and incorporates multi-modal representation for discriminative action classification. Wang et al. [128] introduced TCA-GCN, which dynamically learns spatial and temporal topologies and efficiently aggregates topological features across different temporal and channel dimensions for human activity recognition. TCA-GCN employs a temporal attention module and a channel aggregation module for feature learning.

GCNs are widely adopted in skeleton-based action recognition due to their powerful ability to model data topology, and these have produced the state-of-the-art results for human action recognition. Our approach most closely resembles [16, 13, 71] where we have a shared topology and an attention based spatial GCN model with a multiscale TCN module to capture temporal features. This multilayered GCN acts as the backbone for our architecture which helps in learning a dynamic topology for the skeleton data. The output of this GCN instead of average pooled with a classification head is instead split into key, quey and value and fed into a transformer model which then has an MLP layer for classification.

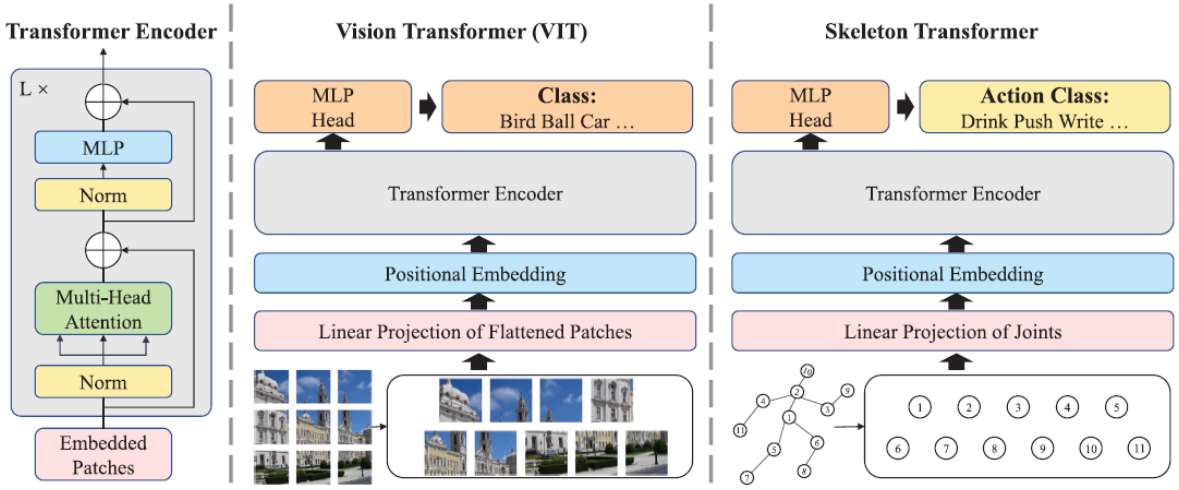

Figure 2.9: Illustrating the comparison between Vision Transformer and Skeleton Transformer in implementation. Figure adapted from [138].

2.2.6 Transformers

Action recognition employing transformers encompasses various approaches. The surge in popularity of transformers in Computer Vision can be largely attributed to the pivotal paper Vision Transformer (VIT) [23]. This seminal work adeptly adapts the encoder module from natural language processing for image encoding, marking a significant milestone in the field. Ranasinghe et al. [89] introduced the self-supervised video transformer, which aligns features across diverse perspectives. Approaches using skeletons for action recognition were adopted using similar approaches to VIT as shown in Figure 2.9. By leveraging 2D skeletal representations of short time sequences, Mazzia et al. [75] introduce Action Transformer (Ac-Former), a fully self-attention architecture for action recognition. Ac-Former employs an encoder derived from the standard transformer architecture, comprising alternating multi-head self-attention and feedforward blocks.

Akbari et al. [4] introduced the Video-Audio-Text-Transformer (VATT), an end-to-end model that learns multi-modal representations from unannotated raw video, audio, and text data by applying multimodal contrastive losses. Furthermore, the Motion-

Transformer [15] achieves temporal dependency comprehension via self-supervised pre-training focused on human actions. The Masked Feature Prediction (MaskedFeat) model [131], pre-trained on unannotated videos using MViT-L, excels in learning diverse visual representations. Similarly, the videomasked autoencoder (VideoMAE) [115], employing vanilla ViT, employs masking strategies for feature learning.

Sun et al. [110] propose MSST-RT, which enhances spatio-temporal modeling by using Spatio-temporal Relative Transformer (ST-RT) that utilizes a lightweight transformer with a relative mechanism, establishing relationships between distant joints in the spatial dimension and frames in the temporal dimension. Additionally, Multi-stream Spatial-Temporal Relative Transformer (MSST-RT) integrates multiple ST-RT pathways to extract spatio-temporal features from four skeleton sequences, enhancing behavior recognition performance. Lie et al. [126] introduced 3MFormer, which leverages three Hierarchical Token (HoT) branches [51], modeling hyper edges of varying orders (from 1 to r) through the creation of several multi-mode token adaptations within the 3Mformer framework. [82, 133] introduced the concept of partitioning in a transformer by embedding joint sets from the same partition into a unified token prior to the attention modules, relying solely on partition strategies for tokenization without incorporating partition-specific attention mechanisms. Although they include skeletal-temporal attention at the joint-group level, these models face limitations in recognizing localized actions due to the tokenization process, which results in the loss of physically similar skeletal information. In contrast, [19] introduces partition-specific skeletal-temporal attention modules, allowing it to effectively capture skeletal-temporal relationships at the joint-element level without the need for tokenization.

Our approach combines the feature output from a GCN backbone, which is then linearly projected into a transformer model. The Transformer is a partitioning style multi-head self attention (MHSA) transformer closely resembling [75, 23, 19]. GCN output feature is linearly projected with positional embeddings and fed into the transformer.

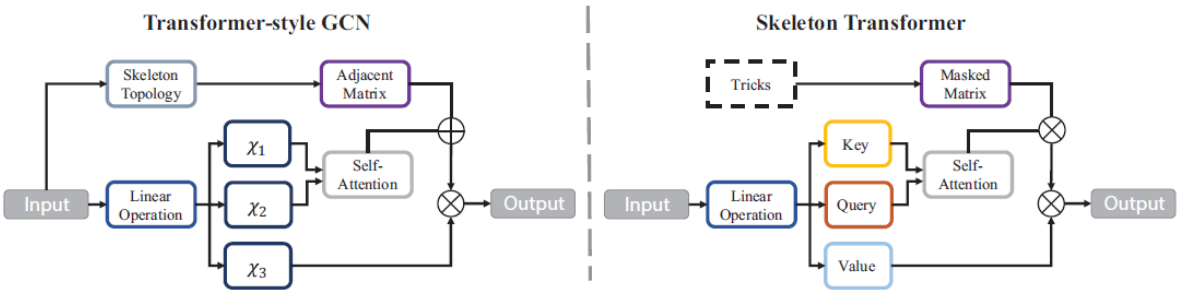

Figure 2.10: Illustrating the comparison between Transformer-style GCN and Skeleton Transformer in implementation. Figure adapted from [138].

This allows our transformer model to capture long-range temporal correlations over the learned spatial features of the 3D skeleton data, thereby enhancing its ability to learn from skeleton-based data.

2.2.7 Hybrid

Various deep learning techniques such as CNNs, RNNs, GCNs, Attention, and Transformers are invaluable in computer vision tasks. However, each approach possesses distinct strengths and weaknesses. To capitalize on their strengths and mitigate weaknesses, researchers have explored hybrid-DNN approaches, blending different architectures. These hybrid models have demonstrated remarkable performance in skeleton-based action recognition. As shown in Figure 2.10, more and more GCNs have started to adopt a Transformer style self-attention block and Transformers have been adapted to use graphical data such as skeletons as inputs.

Si et al. [105] proposed AGC-LSTM (Attention Enhanced Graph Convolutional LSTM Network), which combines discriminative features in spatial and temporal domains while exploring their co-occurrence relationship. AGC-LSTM utilizes an attention mechanism to learn high-level semantic representations efficiently. Shi et al. [100] introduced DSTA-Net (Decoupled Spatial-Temporal Attention Networks), employing pure attention modules without manual design of traversal rules or graph topologies. DSTA-Net utilizes

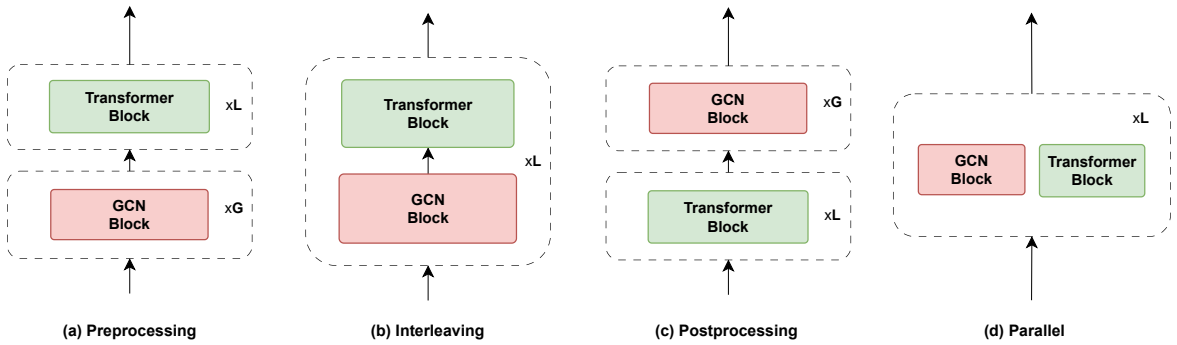

Figure 2.11: Types of GNN-as-Auxiliary-Modules with Transformer architecture.

spatial-temporal attention decoupling and position encoding to construct attention networks. Xiang et al. [136] utilized a large-scale language model to provide text descriptions for body parts' movements in actions. They proposed a multi-modal training scheme, dividing the skeleton into parts and encoding each part with descriptive text for action representation learning. Trivedi and Sarvadevabhatla [118] proposed PSUMNet (Part Stream Unified Modality Network), employing a combined modality part-based streaming approach. PSUMNet reduces parameter count while achieving superior performance across skeleton action recognition datasets. Zhou et al. [153] developed Hyperformer, a hybrid model incorporating bone connectivity into the Transformer via a graph distance embedding. Hyperformer introduces Hypergraph Self-Attention (HyperSA) to integrate higher-order relations into the model, narrowing the performance gap between Transformers and GCNs. Gao et al. [35] proposed FG-STForm, an end-to-end Focal and Global Spatial-Temporal Transformer network. FG-STForm effectively captures relations among local joints and global contextual information in both spatial and temporal dimensions, leveraging selective focal joints and dilated temporal convolutions. Meng et al. [76] introduced TAG, an optimized ST-GCN framework integrated with Transformer structure. TAG incorporates adaptive graph convolutional layers and attention mechanisms to enhance feature extraction and generalization capabilities.

Integrating GNNs with Transformers enhances both global and local relation modeling. Fig 2.11 shows four architectures that are commonly used: (i) GNN block before

Transformer block also called preprocessing, (ii) interleaving GNN and Transformer layers, (iii) postprocessing with GNN after the Transformer block and (iii) parallel GNN and Transformer blocks. For instance, GraphTrans [134] applies a Transformer atop a GNN layer to improve graph classification. Rong et al. [90] uses GTransformer modules with a dyMPN for molecular data, enhancing node and edge representations. GraphiT [77] employs a Graph Convolutional Kernel Network (GCKN) to encode sub-structures with kernel embeddings, enriching structural information. Mesh Graphormer [68], using the second architecture, combines graph residual blocks with multi-head self-attention to model local and global interactions in 3D mesh data. Graph-BERT [148] applies the parallel architecture, incorporating a graph residual term in each attention layer.

Improved positional embeddings (PE) from graphs enable Transformers to encode structural information without major architectural changes. Approaches include Laplacian eigenvectors for structural data [27] and SVD on the adjacency matrix [43], where top singular values are used as embeddings with random sign-flipping for augmentation. Heuristic PEs such as degree centrality in Graphormer [145] capture node importance. Graph-BERT [148] adds Weisfeiler-Lehman codes and hop-based embeddings, while AMR graph tasks use tree-structured distance embeddings [10]. Other methods, like those in [56], learn PEs directly from the Laplacian spectrum.

To further integrate graph structure, new approaches modify attention matrices based on graph-specific data [134]. One technique restricts nodes to attend only to neighbors, enhancing neighborhood representation in competitive models. GraphiT [77] extends adjacency matrices to kernel matrices, encoding various graph features and balancing highly connected nodes with degree matrices. Graphormer [145] introduces spatial biases with shortest path distances and edge-based biases, refining attention mechanisms for graph data. Gophormer [151] incorporates multi-hop proximity data, capturing richer structure. PLAN [50] tailors attention for tree-structured data, encoding parent-child relationships, which enhances propagation modeling in social media contexts. These techniques thus

customize attention matrices with graph-aware biases, enhancing performance on structurally complex tasks.

Our architecture falls into this category as we have a GCN-backbone with a Transformer encoder head followed by a classification layer. Unlike other models in this category, our approach is much simpler, as we use standard architectures with minimum modifications and computationally less expensive, by partitioning strategies using domain knowledge to breakdown the feature space into smaller problems reducing the overall calculations needed in training, while producing state of the art results.

Chapter 3

Background

This chapter aims to provide a brief overview of the concepts and algorithms used in this research which is required to understand future sections. We start by giving a brief overview of the problem and then discuss the various algorithms and techniques used in this research. We also discuss the dataset used in this paper. The problem that we are addressing in this research is human action recognition using 3D skeleton data, which involves the automatic identification of human actions from video or image sequence data. More specifically, we focus on cross-view multi-subject human action recognition that requires recognizing actions from multiple viewpoints or camera angles involving one or more subjects performing an action.

3.1 Human Action Recognition

Human Action Recognition is the process of classifying an action being performed by a subject or multiple subjects in a given image sequence or video. Recognition of human actions using 3D skeleton modality has garnered increased attention over the past few years with the development of efficient 3D pose estimation models which produce accurate 3D skeletons of humans using just video sequences. Understanding human action using just one modality in general is often insufficient as it generally fails to capture some or

all of the following: 3D range of motion, discriminate between diverse human body types and clothing, object interactions, different lighting conditions, varying camera angles, occlusions, illumination changes, etc.

3.2 Cross-view Human Action Recognition

Human Action Recognition becomes even more challenging when we consider different camera viewing angles. CNNs do a great job at learning spatial features in a given frame but these learned feature representations are not invariant to changes in the camera perspective for an action. This is especially important because in the wild, a video can be recorded from any angle and the trained model should be able to recognize different action classes. Skeleton based approaches capture exact motion in 2D or 3D space which is often key in determining the action being performed.

3.3 Deep learning based approaches

Deep learning-based approaches [2, 49, 55, 111, 138] have consistently demonstrated state-of-the-art performance in the field of action recognition, owing to their ability to automatically extract high-level features from complex data. In the context of 3D skeleton-based action recognition, recent advancements can be classified into four categories, each distinguished by its foundational modeling technique. These categories include (1) Convolutional Neural Networks (CNNs), (2) Two-Stream Networks, (3) Transformer-based architectures, and (4) Graph Convolutional Networks (GCNs). Each of these approaches presents distinct advantages and limitations: CNNs excel at capturing spatial hierarchies, Two-Stream networks leverage complementary modalities for enhanced performance, Transformers are adept at modeling long-range dependencies through self-attention mechanisms, and GCNs are particularly effective at learning structural relationships between nodes in graph-like data such as skeletal joints.

CNNs are robust and versatile in capturing different types of features. 3D CNNs use a 3D array of image sequences including the image height and width and the consecutive temporal frames. Convolution on the time axis means we are embedding the temporal variations in the frames in our representations. We also have Temporal Convolution Networks (TCNs) which perform convolutions on the time domain with dilations to increase the receptive field and capture longer range dependencies in the data.

Two Stream approaches use a mechanism, which has been inspired by the human vision system. One stream focuses on capturing temporal or motion features and the other captures spatial or structural features. The spatial stream captures the structural representation at each time instance while the temporal stream tries to capture temporal variations in the joints. These are fused together to make the final prediction using various types of architectures and techniques.

Transformers have taken the center stage in most deep learning based approaches in various fields and computer vision is no different. Transformer based models perform on par with GCN model approaches and consistently show results comparable to state-of-the-art results. The most widely used baseline transformer model is the multi-headed self-attention transformer, with or without masking for causality. This transformer is versatile, robust and generalizes well on any input data distribution with a self-attention scheme which can capture global relationships in data points making it a viable model for most machine learning tasks. For optimal performance, these transformer models need a large amount of training data.

Graph Convolutional Network (GCN) based approaches are among the most popular in action recognition. GCNs perform convolutions on skeleton data over time to learn the movements performed by subjects for various actions. Using adjacency matrices to capture skeleton topology, these are well-suited to capture the graph-like nature of a human skeleton. These usually consist of a spatial block and a temporal block, which together capture both the spatial layout of the skeleton and the timing of the action being

performed. The key thing that GCNs struggle with is long term temporal modeling and are generally more sensitive to noise in the data than other models.

3.4 Algorithms

Several steps and algorithms involved in our approach require some prior understanding. These include Convolutional Neural Networks (CNNs), self-attention mechanisms, transformers, Graph Convolutional Networks (GCNs), and Temporal Convolutional Networks (TCNs).

3.4.1 Convolutional neural networks

Convolutional neural networks (CNNs) are a class of deep neural networks which are used to analyze visual features. Taking an image or a sequence of images as input a CNN can learn the spatial and temporal dependencies between features in images by applying relevant filters. The model's weights learn to identify features, starting from simple lines and edges at the initial layers and progressing to more complex abstractions as we move deeper into the network. A typical CNN consists of a few different types of layers such as convolution, fully connected, non-linear activation and pooling layers.

Each layer, except for the pooling layers, takes a 3D tensor as input, applies a function, and outputs a differentiable 3D tensor, allowing the weights to be learned through backpropagation. Each layer may or may not have parameters or additional hyperparameters. Larger CNNs also include skip connections to ensure the propagation of gradients throughout the network

3.4.2 Self-Attention

Self-attention is a core part of the transformer architecture. It is also known as scaled dot-product attention. When processing a particular element in the sequence, attention

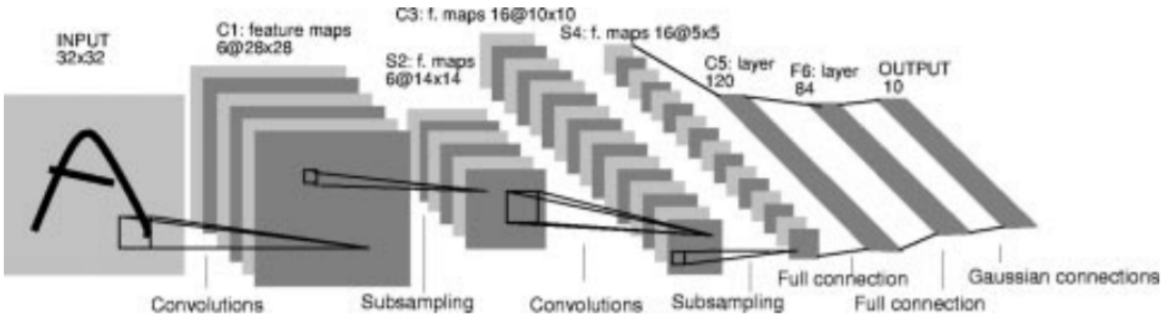

Figure 3.1: Architecture of LeNet-5, a CNN, here used for digits recognition. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical. Figure adapted from [62].

is used to weigh the importance of different parts of an input sequence. This mechanism allows the transformer to capture relationships and dependencies between tokens or data points in the input sequence, making it a powerful model which can attend to different parts of the data based on a global context.

In self-attention, a query, key, and value are associated with each element (token) in the input sequence. The idea is to compute a weighted sum of the values, where the weights are determined by the similarity between the query and the keys. This similarity score is obtained through a dot-product operation, followed by scaling and a softmax function. The output is the weighted sum of values, which represents the contextual information for the input token at hand. Given a sequence of input tokens $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ where $\mathbf{X} \in \mathbb{R}^{n \times d_k}$, which is of the form

$$\mathbf{X} = \begin{bmatrix} \ldots & \mathbf{x}_1 & \ldots \\ \ldots & \mathbf{x}_2 & \ldots \\ & \vdots & \\ \ldots & \mathbf{x}_n & \ldots \end{bmatrix}.$$

We calculate attention scores by spiting $\mathbf{X}$ into $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ and applying the attention function:

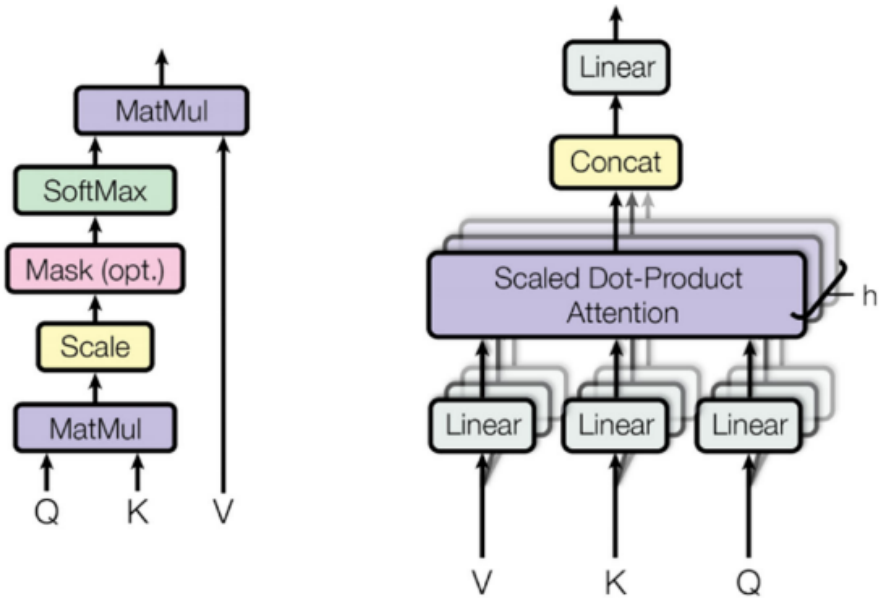

Figure 3.2: (Left) A self-attention block used in Transformers and modern GCNs. Given an input tensor such as a sequence of image features, the self-attention mechanism computes the key, query, and value vectors for each feature. It then calculates the attention scores, which are used to reweigh the value vectors. This process involves a single attention head. Finally, an output projection (W) is applied to produce output features with the same dimensions as the input features. (Right) Multiple self-attention blocks are stacked together in parallel to form a multi-head self-attention module which can attend to different aspects of the input features. Figure adapted from [121].

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{3.1}$$

here $\mathbf{Q} = \mathbf{X}\mathbf{W}_\mathbf{Q}, \mathbf{W}_\mathbf{Q} \in \mathbb{R}^{d_k \times n}$, $\mathbf{K} = \mathbf{X}\mathbf{W}_\mathbf{K}, \mathbf{W}_\mathbf{K} \in \mathbb{R}^{d_k \times n}$ and $\mathbf{V} = \mathbf{X}\mathbf{W}_\mathbf{V}, \mathbf{W}_\mathbf{V} \in \mathbb{R}^{d_v \times n}$, represent the query, key and value matrices. d_k is the dimension of the query and key vectors and d_v is the dimension of the value vectors. The division by $\sqrt{d_k}$ is used for scaling to prevent very small or very large values in the dot product, which can cause issues during training. After calculating attention scores, we apply the softmax function to obtain normalized weights. The final step involves computing the weighted sum of the value vectors by multiplying the softmax result with the value vectors. Finally the result is of the following dimension $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{n \times n}$.

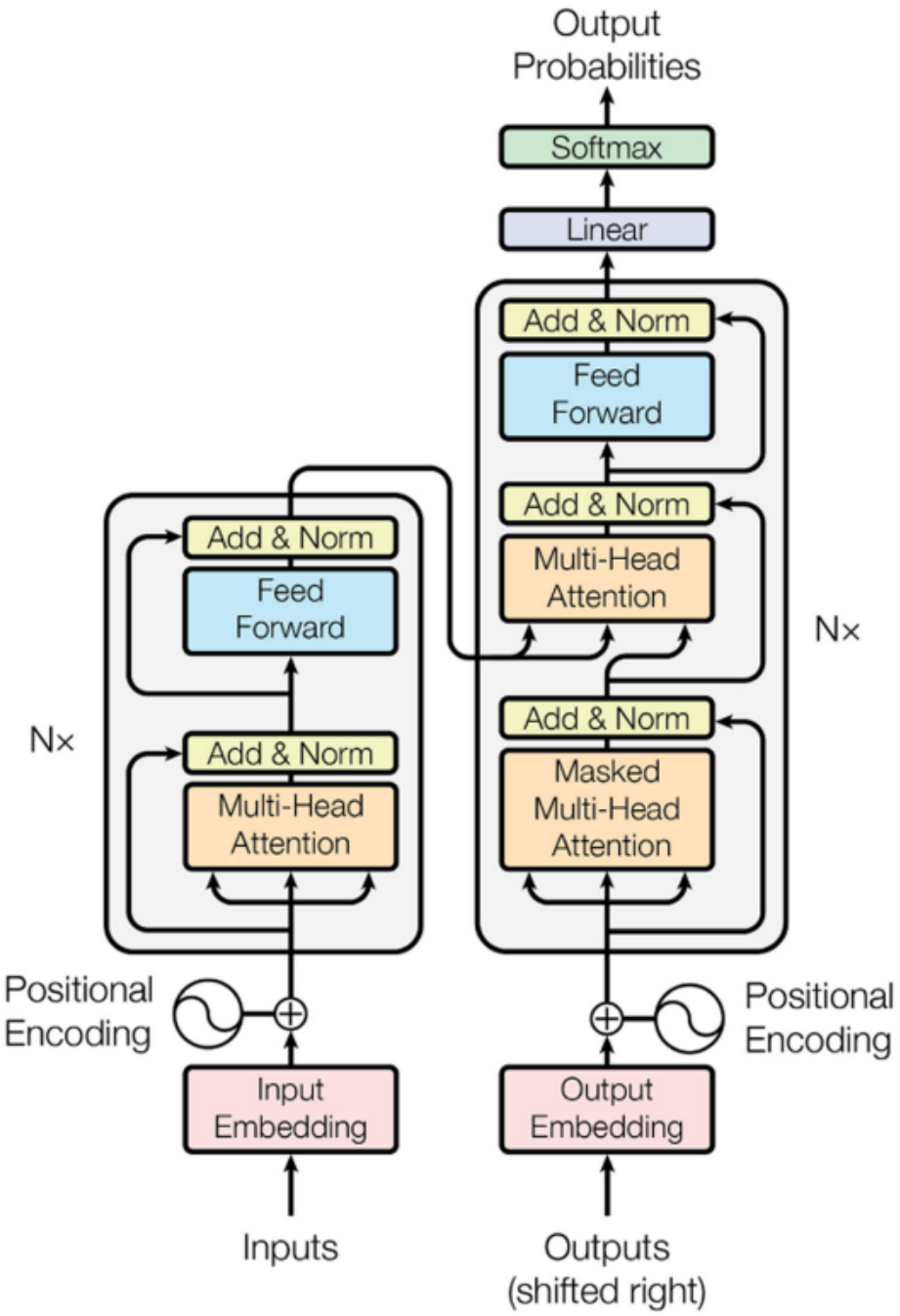

Figure 3.3: Architecture of the original Transformer model with an encoder and decoder style approach. Figure adapted from [121].

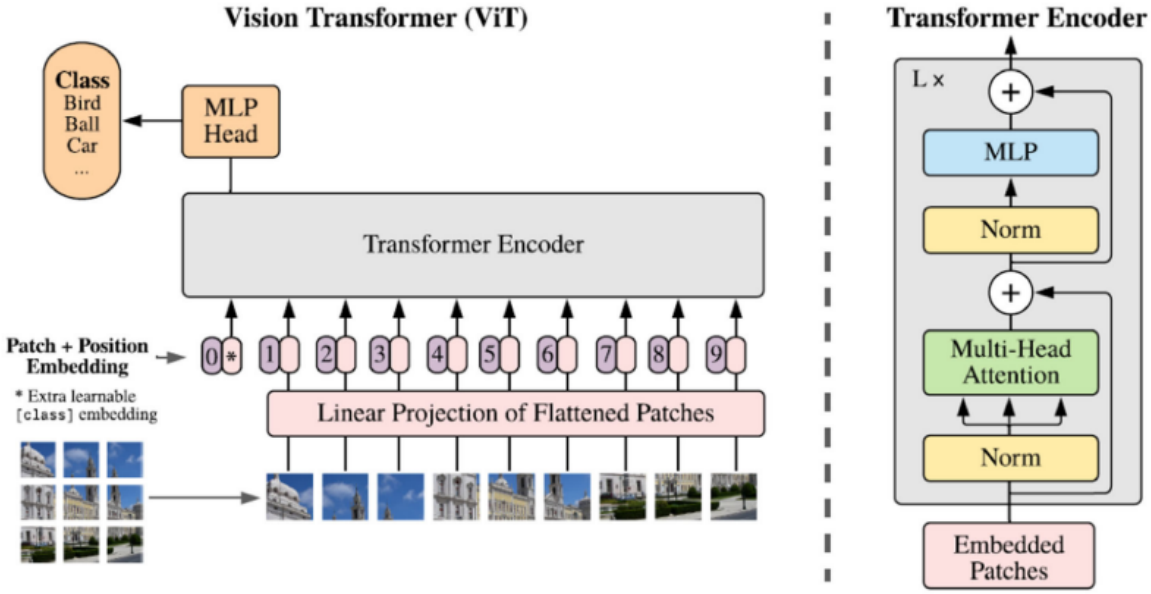

Figure 3.4: Architecture of the Transformer Encoder model which is used in Vision based problems. Figure adapted from [22].

3.4.3 Transformers

Transformers include, a positional embedding module, an attention block, addition and normalization blocks and finally output tokens. For the purpose of this research we are going to focus on the multi-head self-attention transformer model. In a multi-head self-attention mechanism, we apply self-attention multiple times in parallel, each time with different learned projection matrices for queries, keys, and values. The outputs of these parallel self-attention heads are then concatenated and linearly transformed to produce the final output. This can be represented as:

$$\text{MHSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_h)\mathbf{W^o}, \tag{3.2}$$

where $\mathcal{H}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$, for each head $\mathcal{H}_i$ we have learned weight matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d_k \times n}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_k \times n}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_v \times n}$ and $\mathbf{W^o} \in \mathbb{R}^{hn \times n}$ is a learned weight matrix used to linearly combine the output of different attention heads. The outputs of all heads are concatenated along a specified dimension (typically the last dimension) to

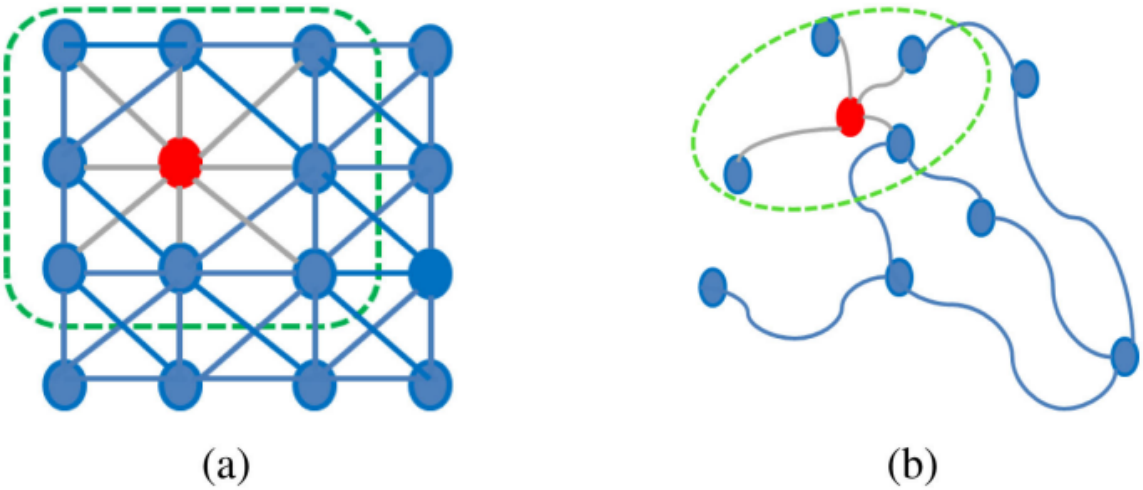

Figure 3.5: Depiction of two-dimensional (Euclidean) convolution in contrast to Graph convolution [26]. (a) The convolution operation within the Euclidean domain. (b) The convolution operation on a graph. Figure adapted from [2].

form a single tensor. In practice, we keep $d_k = d_v = n$.

This multi-head self-attention mechanism allows the model to focus on different parts of the input sequence in parallel, enabling it to capture various types of dependencies and relationships. This is a fundamental component of the transformer architecture, which has been highly successful in various natural language processing and computer vision tasks.

3.4.4 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are a class of neural network architectures specifically designed to perform learning tasks on data represented as graphs, leveraging the inherent structure and relationships within the graph. They have gained significant popularity in various applications, especially in the analysis of data with inherent graph structures such as social networks, recommendation systems, and even human action recognition using 3D skeleton data.

Human action recognition using 3D skeleton data involves understanding and classifying human movements. In this context, each frame of a video or motion capture

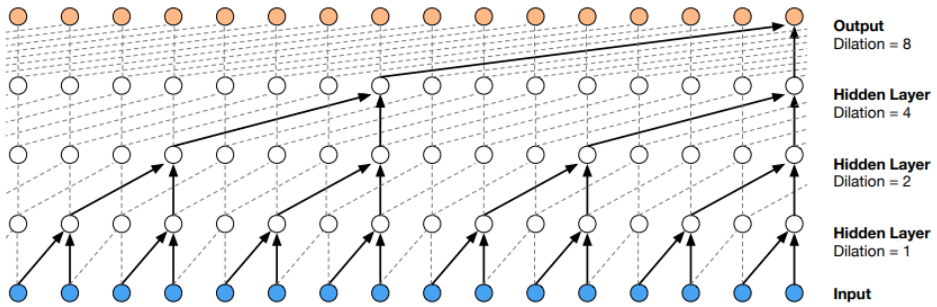

Figure 3.6: TCN: Visualization of a stack of dilated causal convolutional layers. Figure adapted from [120].

data can be represented as a graph, where the joints of the human body are nodes, and the edges represent the spatial relationships between these joints. GCNs can be used to effectively model these relationships and recognize complex human actions.

To represent the topology of 3D skeleton data, we use a graph $\mathcal{G}(N, E)$, where the joints constitute a set of N nodes and the bones are represented as edges E. GCNs typically use an adjacency matrix (often denoted as $\mathbf{A} \in \mathbb{R}^{n \times n}$) that encodes the relationships between nodes (joints). The adjacency matrix describes which joints are connected to each other and how strong these connections are. In the case of human action recognition, this matrix describes the joints, which are edges between pairs of nodes. The adjacency matrix can be computed from the 3D skeleton data. Each entry in the adjacency matrix represents the presence or absence of an edge between two joints. For example, if joint i is connected to joint j, $\mathbf{A}[i][j]$ is set to 1; otherwise, it is 0. The strength of the connection can be modeled by setting $\mathbf{A}[i][j]$ to a value other than 1 or 0, depending on the application.

The core of Graph Convolutional Networks (GCNs) lies in the graph convolution operation, which processes information at each node in the graph by aggregating information from its neighbors. Lets consider an input $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ where $\mathbf{X} \in \mathbb{R}^{n \times d}$. We consider a multi-layer GCN that follows this layer-wise propagation rule:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(l)}\mathbf{W}^{(l)}). \tag{3.3}$$

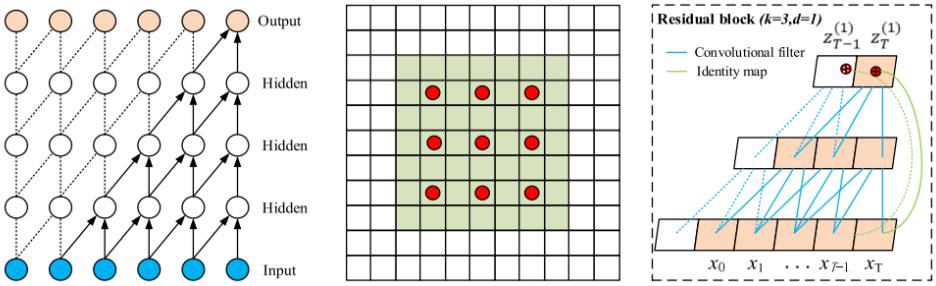

Figure 3.7: Architectural Elements in Multi-Scale TCN: (Left) Schematic of causal convolution: Ensures that the convolution operation respects the temporal order of data. (Middle) Dilated convolution: Introduces gaps between the kernel elements to exponentially increase the receptive field, enabling the model to capture long-range dependencies more effectively. (Right) Residual connection: Facilitates the training of deeper networks by allowing the input to bypass one or more layers, mitigating the vanishing gradient problem improving convergence rates and model performance. Figure adapted from [144].

In this equation, $\mathbf{H}^{(l+1)}$ represents the updated feature matrix at layer l, since $\mathbf{H}^{(0)} = \mathbf{X} \in \mathbb{R}^{n \times d}$. The function $\sigma(\cdot)$ denotes a non-linear activation, such as $\text{ReLU}(\cdot) = \max(0, \cdot)$. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ represents the adjacency matrix of the undirected graph $\mathcal{G}$, with added self-loops. $\mathbf{I}_n$ is the identity matrix, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times d}$ is the matrix of activations at the l-th layer and $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times n}$ is a trainable weight matrix specific to each layer.

In the context of human action recognition using 3D skeleton data, GCNs can be used to model the spatiotemporal relationships between joints over time, allowing the network to learn and recognize complex human actions based on the topological structure of the skeleton data. By aggregating information from neighboring joints in a graph-like fashion, GCNs have demonstrated their efficacy in capturing meaningful patterns in such data, enabling accurate action recognition.

3.4.5 Temporal Convolutional Networks

Temporal Convolutional Networks (TCNs) have emerged as a robust alternative to Recurrent Neural Networks (RNNs) for sequence modeling tasks. TCNs leverage the convolutional architecture, which is traditionally used in computer vision, to process sequential data, offering advantages in terms of parallelism, stable gradients, and flexible receptive

fields. Key Features of Temporal Convolutional Networks include causal convolutions, dilated convolution and residual connections. Unlike standard convolutions, TCNs use causal convolutions to ensure that there is no information leakage from future to past. This is crucial for time-series forecasting and other sequential tasks where the order of data points is significant. TCNs also employ dilated convolutions to increase the receptive field without losing resolution or increasing the computational cost significantly. This enables the model to capture long-range dependencies more effectively. Inspired by ResNet architectures, TCNs often include residual connections that help in training deeper networks by mitigating the vanishing gradient problem.

3.5 Summary

This chapter provided an overview of human action recognition, with a particular focus on cross-view action recognition. It discusses the significance of accurately identifying and classifying actions across different perspectives, highlighting the challenges posed by varying viewpoints.

The chapter also explores deep learning-based approaches that have transformed action recognition, allowing for improved model performance. Key algorithms are discussed, including Convolutional Neural Networks (CNNs), which are essential for feature extraction; self-attention mechanisms, which enhance understanding of data relationships; and transformers, known for their effectiveness in processing global sequential data.

Additionally, the chapter covers Graph Convolutional Networks (GCNs), which analyze structured data to capture relationships in human skeleton movements, and Temporal Convolutional Networks (TCNs), designed for modeling time-dependent patterns. Overall, this chapter sets the stage for understanding the algorithms and methodologies used in our work.

Chapter 4

Methodology

This work aims to leverage the strength of GCNs which learn graphical data such as the human skeleton efficiently, connected to a transformer, which learns global temporal patterns in the input. We apply our architecture to the complex problem of multi-person, multi-view human action classification using skeleton data. This chapter covers our model architecture, breaking down each component in detail.

4.1 Architecture

We introduce an architecture that can model multi-person, multi-view spatiotemporal interactions. Our model, shown in Figure 4.1 consists of a GCN backbone which takes in a linear projection of the input data and outputs an intermediary learned feature. The GCN backbone comprises of a self-attention GCN module and a multi-scale TCN module which learn infered topology of the input skeletons learning the relationships between different joints for differnet action classes. The output of our GCN backbone is consumed by a partitioning multi-head self-attention transformer model, which partitions the features into four smaller dimensional features, applying a partitioning strategy forcing the model to attend to different aspects of the action. These four categories are neighbouring joints over short intervals, neighbouring joints over large intervals, distant joints over short

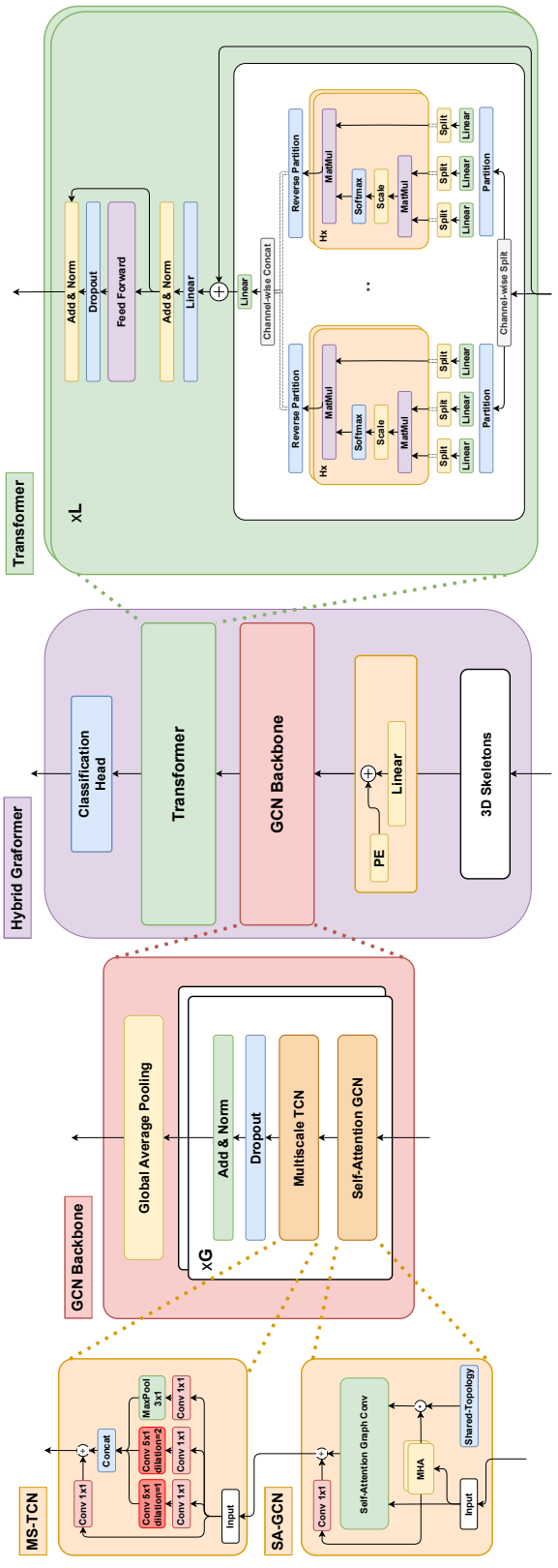

Figure 4.1: Overview of our architecture highlighting the key components. Our model contains a GCN backbone which learns inferred topology from 3D skeleton data and feeds into our partition style Transformer which learns discriminative features based on different partitions and finally the classification head gives the action class.

intervals and distant joints over large interval. We will discuss the breakup in more detail in a later subsection. Finally, the transformer output is passed to a pooling layer and a fully-connected classification layer which gives us the action class.

The human skeleton is conceptualized as a graph $\mathcal{G}(V, E)$, where the joints constitute a set of N vertices V and the bones are represented as edges E as shown in Figure 1.1. These edges can be denoted through an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where n is the total number of nodes and $\mathbf{A}_{i,j} = 1$ means there is a physical connection (bone) between joints i and j, otherwise we use $\mathbf{A}_{i,j} = 0$ to denote no connection between the joints.

Our datasets are of the form $\mathbf{X}_{dataset} \in \mathbb{R}^{M \times T \times N \times C}$, where M = number of skeletons, T = number of frames, N = number of joints, and C = number of channels, which in this case represents joint location in 3 dimension (x, y, z). For an action sequence, we consider all individual actors in a scene separately in dimension M so that our architecture can learn from all present humans performing an action, but for simplicity we will present our equations with $\mathbf{X}_{input} \in \mathbb{R}^{T \times N \times C}$ in the next sections.

4.1.1 Embedding Layer

The embedding block initially applies a linear transformation to the joint features, converting them into vectors of $D^{(0)}$ dimensions using learnable parameters, where $D^{(0)}$ is the base channels that we set based on experimental results. Additionally, positional embeddings (PE) are incorporated to convey positional information about the joints. Here, we utilize learnable PE, which remains consistent across different time frames. Using positional embeddings in this learnable format allows the model to learn it based on the input data but it restricts the training and test data to have the same number of frames.

$$\mathbf{H}_t^{(0)} = \text{Linear}(\mathbf{X}_t) + \text{PE} \tag{4.1}$$

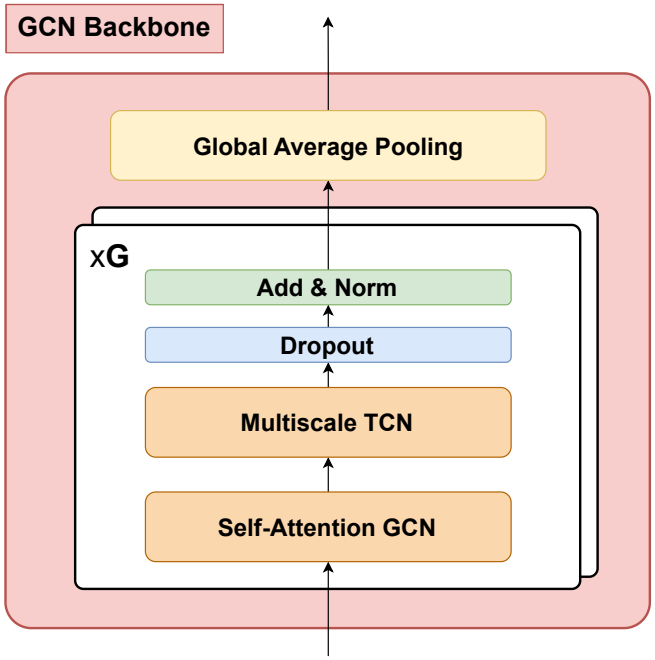

Figure 4.2: An overview of our GCN architecture with the key components.

where $\mathbf{H}_t^{(0)} \in \mathbb{R}^{N \times D^{(0)}}$ at time $T = t$, $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ which is projected to $\mathbb{R}^{N \times D^{(0)}}$ and $\text{PE} \in \mathbb{R}^{N \times D^{(0)}}$. This $\mathbf{H}^{(0)} \in \mathbb{R}^{T \times N \times D^{(0)}}$ is then sent to the GCN backbone.

4.1.2 GCN Backbone

Our GCN backbone is based on current GCN models which have produced state of the art results on major benchmarks. Most of these methods use the feature update rule of [53]. Our backbone, as shown in Figure 4.2, consists of two main modules, a spatial self-attention GCN (SA-GCN) block and a Multi-Scale Temporal Convolution Network (MS-TCN) block. The SA-GCN module takes inspiration from a vanilla GCN module which averages neighborhood vertex features and linearly transforms aggregate features as discussed in section 3.4.4.

SA-GCN, seen in Figure 4.3, employs self-attention on joint features to deduce intrinsic topology and leverages this topology as neighborhood vertex information. Self-

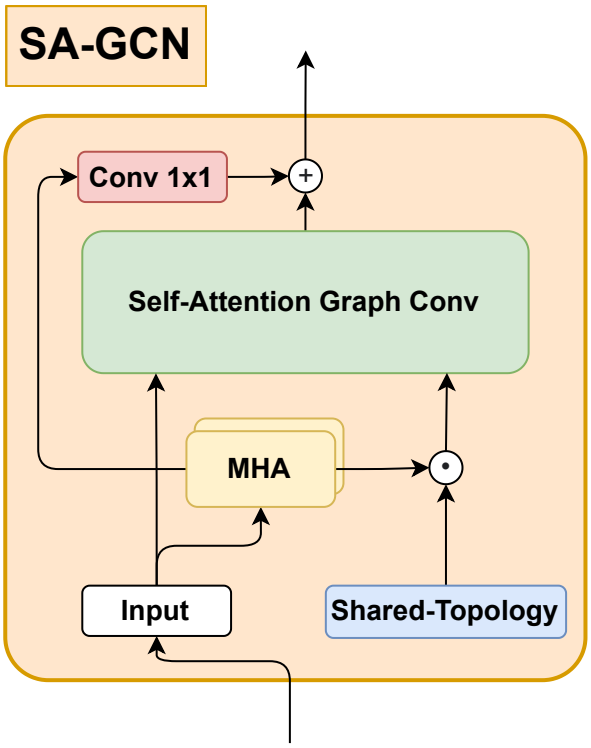

Figure 4.3: An overview of our SA-GCN architecture with the key components.

attention here is an attention mechanism that assesses connections between various body joints. By examining all potential joint relations, SA-GCN derives positive and constrained weights, termed self-attention maps, to quantify the intensity of these relationships. Similar to our discussion about self-attention in section 3.4.2, we apply the attention map to our input $\mathbf{H}_t^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$ at time $T = t$ at layer l as

$$\text{AttentionMap}\left(\mathbf{H}_t^{(l)}\right) = \text{softmax}\left(\frac{\mathbf{H}_t^{(l)}\mathbf{W}_Q^{(l)}\left(\mathbf{H}_t\mathbf{W}_K^{(l)}\right)^T}{\sqrt{D^{(l')}}}\right) \tag{4.2}$$

where we linearly project $\mathbf{H}_t^{(l)}$ to the queries and keys of $D^{(l')}$ dimension with learned matrices $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{D^{(l)} \times D^{(l')}}$ to get the self-attention map.

Furthermore, in addition to the self-attention map, SA-GCN is trained to learn a

shared topology $\tilde{\mathbf{A}}$ across time and instances, akin to prior works [13, 97]. Both the shared topology and self-attention map utilize multiple heads M to enable the model to simultaneously attend to various representation subspaces. This learned topology using shared adjacency matrix and self-attention, that we call intrinsic topology is calculated using $\tilde{\mathbf{A}}_m \odot \text{AttentionMap}_m(\mathbf{H}_t) \in \mathbb{R}^{N \times N}$, where $\odot$ indicates the broadcast element-wise product. For each head in $1 \leq m \leq M$, we do an element-wise product of the shared topology $\tilde{\mathbf{A}}_m \in \mathbb{R}^{N \times N}$ with the self-attention map $\text{AttentoinMap}_m(\mathbf{H}_t) \in \mathbb{R}^{N \times N}$ to obtain the learned intrinsic topology. We employ $D^{(l')} = D^{(l)}/8$ and $M = 3$ in this work. Since $\tilde{\mathbf{A}}_m \odot \text{SA}_m(\mathbf{H}_t)$ represents the neighborhood information for our GCN, the overall update rule is

$$\mathbf{H}_t^{(l+1)} = \sigma\left(\sum_{m=1}^{M}\left(\tilde{\mathbf{A}}_m^{(l)} \odot \text{AttentionMap}_m\left(\mathbf{H}_t^{(l)}\right)\right)\mathbf{H}_t^{(l)}\mathbf{W}_m^{(l)}\right), \tag{4.3}$$

where we add the results of all heads together which gives us our adjacency matrix with new learned relationships using self-attention between all joints at layer l, which is then used similar to a normal GCN update rule as discussed previously. We also add a 1×1 convolution layer as a residual connection.

For modeling the temporal features of the human skeleton, we incorporate the MS-TCN module [16, 71, 13]. The input to this module is the output from the previous SA-GCN layer which is $\mathbf{H}^{(l)} \in \mathbb{R}^{T \times N \times D^{(l)}}$. The module comprises of three convolution branches with different kernel sizes and dilation rates. The outputs from these branches are concatenated, and a residual connection with a 1×1 convolution is added to facilitate training. We extend standard temporal convolution layers by integrating multi-scale learning, enabling the model to effectively capture temporal dynamics across actions of different lengths. This module operates on the time dimension, aggregating causal temporal data, resulting in $\mathbf{H}^{(l+1)} \in \mathbb{R}^{T \times N \times D^{(l)}}$.

The GCN-backbone thus consists of multiple layers of SA-GCN, MS-TCN, dropout, and normalization functions. The backbone has G total layers followed by a global

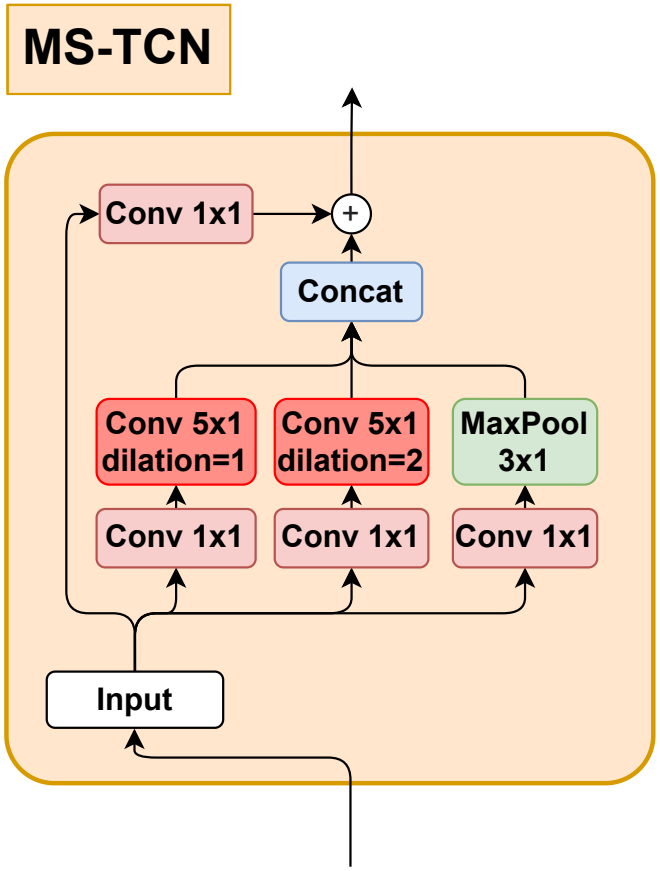

Figure 4.4: An overview of our MS-TCN architecture with the key components.

average pooling layer, which produces our GCN latent feature $\mathbf{X}_{\text{gcn}} \in \mathbb{R}^{T \times N \times D'}$.

4.1.3 Transformer

In this work, we use an architecture that captures the intricate dynamics of human actions through 3D skeleton data input over time. The core idea of our approach is to partition the skeletal joints and frames based on distinct types of joint-temporal relations and perform self-attention within each partition. Our partitioning strategy takes direct inspiration from [19] as shown in Figure 4.5. We categorize the key skeletal-temporal relations into four distinct types:

1. **Neighboring joints**: These split the skeleton joints into L total elements of size

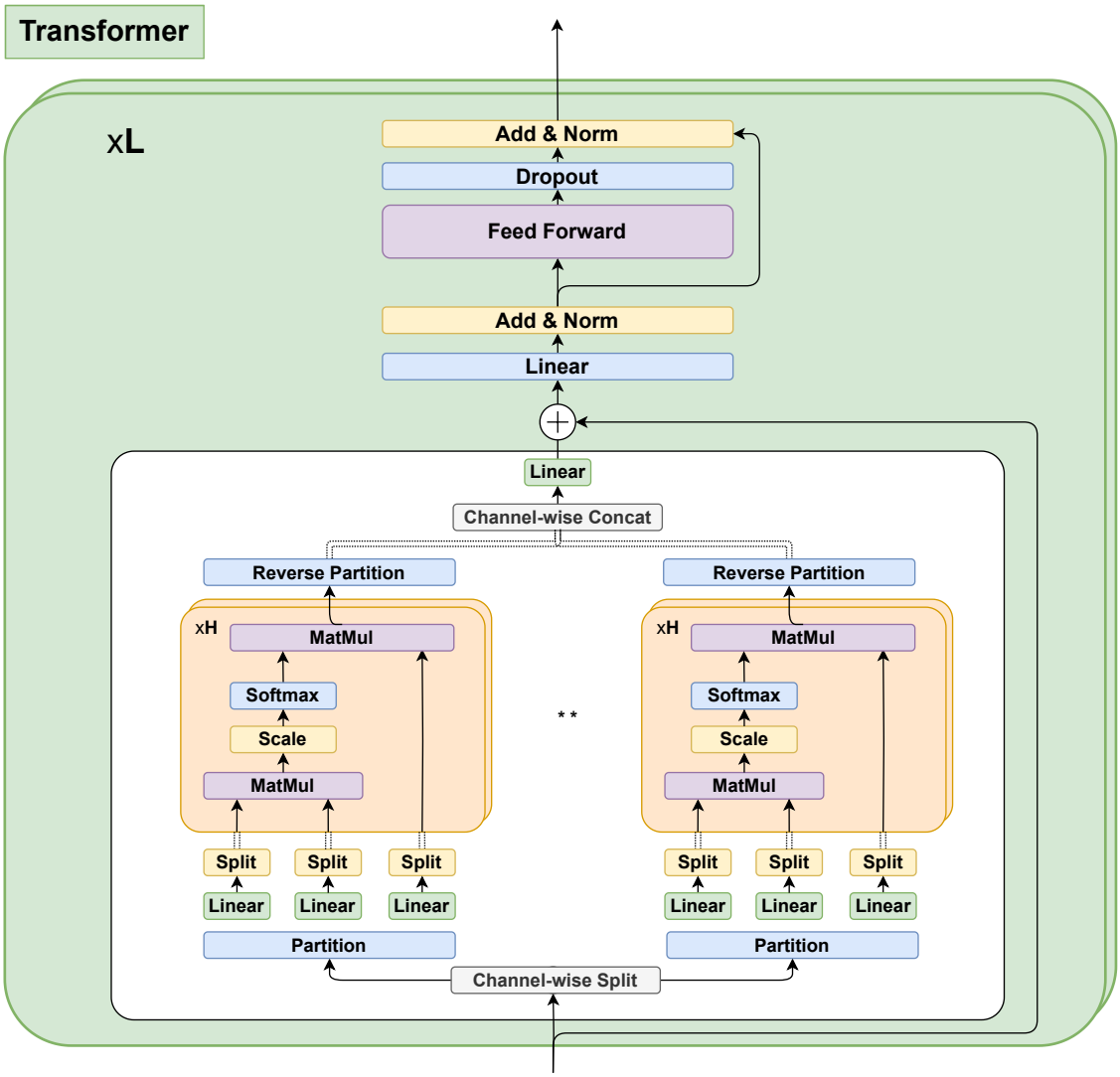

Figure 4.5: An overview of our Transformer architecture with the key components.

K each, where $K \times L = V$, the total number of joints. We extract these L elements of connected joints of size K each and stack them to create a Matrix of size $K \times L$.

2. **Distant Joints**: To create Distant joints, we transpose the neighboring joints to create a matrix of size $L \times K$. This moves the closer joints to interact with joints that are not in their immediate neighborhood.

3. **Neighboring frames**: To create a partition where neighboring frames are closer together we select N as the number of consecutive frames and M as the total elements where $N \times M = T$.

4. **Distant frames**: To create a partition with distant frames together we pick every Mth element starting from first and create N total elements, where $M \times N = T$.

By partitioning the skeletal data in this manner, the multi-head self attention modules attend to different aspects of the data based on the type of partition applied. For example the first type of partition rearranges joints to keep neighboring joints together and neighboring temporal frames together making this module focus on learning features that depend on local relationships. Similarly a partition grouping together distant joints and distant frames puts the constraint on the module to learn features that represent distant joint relationships and longer temporal windows. We will discuss the partitioning functions in more detail in the subsequent sections.

Why use partitions?

If we have infinite data, the vanilla Transformer can learn complex global relationships automatically and generalize but in real-world datasets we have limited data so we can use our domain knowledge to focus on key parts of the data which can aid in the learning process. We know human actions are varied in nature. Sometimes neighboring joints are important, for example for the action class "waving hand," the model should only pay attention to the arm which is performing the action of waving. On the other hand,

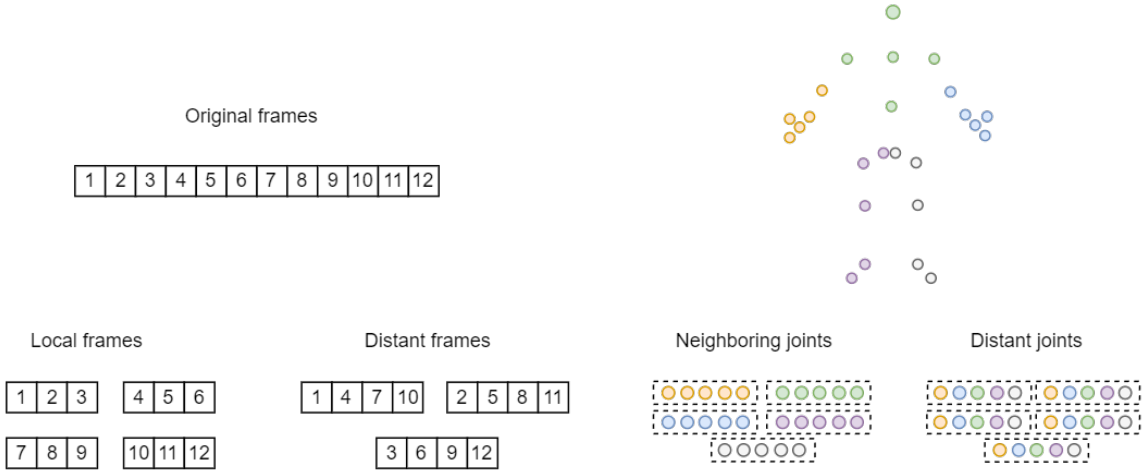

Figure 4.6: An overview of our strategy of partitioning the temporal and skeletal data.

another action like "clapping" requires attending to both hands which are not neighboring joints. In the same way, time also plays an important role in determining an action, for example the action "sitting" requires us seeing a person standing in the initial frames and eventually in a siting pose which covers the sitting action, this represents a global temporal attention to figure out this example. Other examples like "kicking" only requires seeing the short motion of performing the kicking action to classify it as "kicking". In this case the global context is not important, the important part is only the short local duration of the action.

This nuance of human actions can be enforced on the model by partitioning the data and grouping data points it in a way which forces this constrained on the self-attention modules. For our model we are creating four partitions as shown in Figure 4.6, which cover local and global relationships for both joints and time frames. Designing the model this way, allows the model to learn each action focusing on these use cases rather than having to figure it out on its own. This constraint on the model results in better learning and improves the overall performance of the model on the same data as shown in the ablative study in Chapter 5.

Partition-Transformer Block

The partition-transformer block serves as the fundamental building block of our Transformer architecture. This block encompasses several key components that contribute to its functionality. Initially, we apply Layer Normalization and a linear projection to the input data, which helps in stabilizing the learning process [121, 5]. Subsequently, the input is divided into four distinct parts. During the Input Partitioning phase, the skeletal data corresponding to each split is partitioned based on the specific types of relationships that the block is designed to address. For instance, neighboring joints in local frames are grouped together, creating partitions that are informed by both neighboring joints and localized temporal windows as shown in Figure 4.6.

For partitioning we first start with our input which is $\mathbf{X}_{\text{gcn}} \in \mathbb{R}^{T\times N\times D'}$. We start by creating a set of nearby joints partition given by $\mathbf{n}_k^{njp} = [n_{k,1}, n_{k,2}, \ldots, n_{k,L}]$, where $k = 1, 2, \ldots, K$, with K being the total number of body parts, and L represents nodes per body part. Since there is no overlap $K \times L = N$, where N is the total number of nodes. Now if we stack these partitions together we get $\mathbf{n}^{njp} = [\mathbf{n}_1^{njp}|\mathbf{n}_2^{njp}|\ldots|\mathbf{n}_K^{njp}] \in \mathbb{R}^{L\times K}$. Additionally, we can use this matrix to create a matrix of distant joint partitions by transposing this matrix to bring together distant joints closer using $(\mathbf{n}^{djp})^T = \mathbf{n}^{djp} = [\mathbf{n}_1^{djp}|\mathbf{n}_2^{djp}|\ldots|\mathbf{n}_L^{djp}] \in \mathbb{R}^{K\times L}$.

Next set of partitions deal with the time dimension. We define $\mathbf{t}^{local}$ and $\mathbf{t}^{global}$ to capture local and global temporal settings. $\mathbf{t}_m^{local} = [(m-1)N+1, (m-1)N+2, \ldots, mN]$, where $n = 1, 2, \ldots, N$ with the total N number of $\mathbf{t}_m^{local}$. $\mathbf{t}^{local}$ is basically a set of consecutive time indices of size N each. Similarly, for the last partition we have $\mathbf{t}_n^{global} = [n, n+N, \ldots, n+(M-1)N]$, where $m = 1, 2, \ldots, M$ with the total M number of $\mathbf{t}_n^{global}$. This defines an N-strided sparse time index to capture global motion patterns.

Bringing all these spatial and temporal partitioning strategies together we define four partitions, (1) near joints with local motion, based on $\mathbf{n}_k^{njp}$ and $\mathbf{t}_m^{local}$ which results in $\mathcal{P}_1 : \mathbf{X} \in \mathbb{R}^{T\times N\times D/4} \rightarrow \mathbb{R}^{MN\times LK\times D/4}$. (2) distant joints with local motion, based on

$\mathbf{n}_l^{djp}$ and $\mathbf{t}_m^{local}$ which results in $\mathcal{P}_2 : \mathbf{X} \in \mathbb{R}^{T\times N\times D/4} \rightarrow \mathbb{R}^{MN\times KL\times D/4}$. (3) near joints with global motion, based on $\mathbf{n}_k^{njp}$ and $\mathbf{t}_n^{global}$ which results in $\mathcal{P}_3 : \mathbf{X} \in \mathbb{R}^{T\times N\times D/4} \rightarrow \mathbb{R}^{NM\times LK\times D/4}$. (4) distant joints with global motion, based on $\mathbf{n}_l^{djp}$ and $\mathbf{t}_n^{global}$ which results in $\mathcal{P}_4 : \mathbf{X} \in \mathbb{R}^{T\times N\times D/4} \rightarrow \mathbb{R}^{NM\times KL\times D/4}$.

Within each partition, a self-attention mechanism is employed to effectively capture the dependencies among the elements similar to the one discussed in section 3.4.3. This mechanism is vital for learning the complex patterns inherent in human actions. Following the application of self-attention, a reverse partitioning operation is conducted to reconstruct the outputs for each partition type.

The reverse partitioning step applies the inverse transformation on each partition to get the original partition back. (1) near joints with local motion, based on $\mathbf{n}_k^{njp}$ and $\mathbf{t}_m^{local}$ which results in $\mathcal{R}_1 : \mathbf{X} \in \mathbb{R}^{MN\times LK\times D/4} \rightarrow \mathbb{R}^{T\times N\times D/4}$. (2) distant joints with local motion, based on $\mathbf{n}_l^{djp}$ and $\mathbf{t}_m^{local}$ which results in $\mathcal{R}_2 : \mathbf{X} \in \mathbb{R}^{MN\times KL\times D/4} \rightarrow \mathbb{R}^{T\times N\times D/4}$. (3) near joints with global motion, based on $\mathbf{n}_k^{njp}$ and $\mathbf{t}_n^{global}$ which results in $\mathcal{R}_3 : \mathbf{X} \in \mathbb{R}^{NM\times LK\times D/4} \rightarrow \mathbb{R}^{T\times N\times D/4}$. (4) distant joints with global motion, based on $\mathbf{n}_l^{djp}$ and $\mathbf{t}_n^{global}$ which results in $\mathcal{R}_4 : \mathbf{X} \in \mathbb{R}^{NM\times KL\times D/4} \rightarrow \mathbb{R}^{T\times N\times D/4}$. Once all partitions are reversed we apply a channel-wise concatenation step to get the tensor of input size $concat(\mathbf{X}_{\mathcal{R}^1}, \ldots, \mathbf{X}_{\mathcal{R}^4}) \in \mathbb{R}^{T\times N\times D/4} \rightarrow \mathbb{R}^{T\times N\times D}$.

The results of the self-attention block are then aggregated in the output aggregation step, forming the final output of the partition-transformer block, which is subsequently fed into the subsequent layers of the architecture. Finally, we incorporate a linear layer accompanied by a residual connection from the input, followed by another layer normalization and a feed forward layer, further enhancing the representational capacity of the model.

4.1.4 Classification Head

The classification head of our architecture consists of two main components: a Global Average Pooling layer and a Linear projection layer. This setup ensures that the learned features from the previous layers are effectively summarized and projected onto the output space, corresponding to the action classes.

Firstly, the Global Average Pooling (GAP) layer is applied to the feature map output by the preceding transformer layer. This operation reduces the dimensions by computing the average of all features across the frames, joints, and channels, thereby producing a fixed-size vector for each input sequence in the batch. Mathematically, this can be expressed as

$$\mathbf{H}_{\text{gap}} = \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{j=1}^{N} \mathbf{H}_{t,j}, \tag{4.4}$$

where $\mathbf{H}_{t,j} \in \mathbb{R}^C$ denotes the feature vector at the t-th frame and j-th joint, T is the total number of frames, N is the total number of joints, and C is the number of channels. The resultant vector $\mathbf{H}_{\text{gap}} \in \mathbb{R}^C$ is obtained by averaging over the temporal and spatial dimensions.

Subsequently, this pooled representation is fed into a linear projection layer, which maps the fixed-size vector into a lower-dimensional space corresponding to the number of action classes. This linear transformation is formulated as

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{H}_{\text{gap}} + \mathbf{b}, \tag{4.5}$$

where $\mathbf{W} \in \mathbb{R}^{C \times K}$ and $\mathbf{b} \in \mathbb{R}^K$ are the learnable weights and biases of the linear layer, respectively, and $\hat{\mathbf{y}} \in \mathbb{R}^K$ is the output vector containing the predicted scores for each action class. The overall architecture of the classification head ensures that the rich feature representations captured from the input sequence are effectively utilized for accurate action classification.

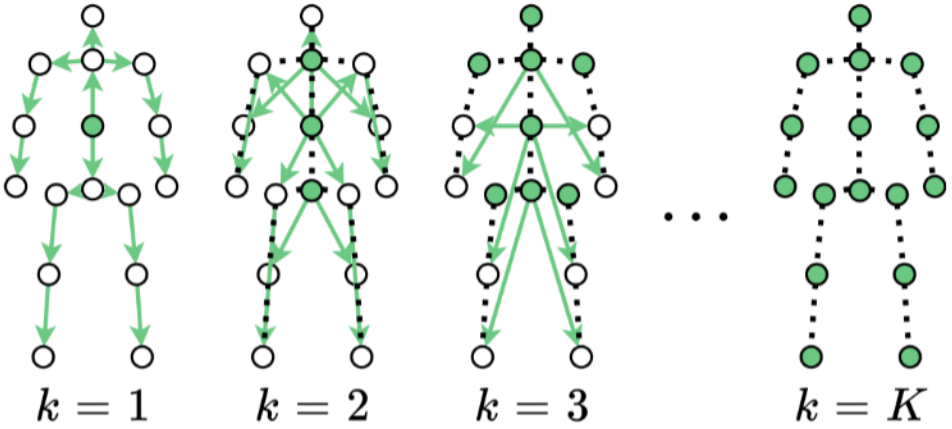

Figure 4.7: Demonstration of Multi-modal Skeleton Representation: Arrows depict the k-th mode representation of pointed vertices. Following the convention established in [16, 97], we designate the joint closest to the center of mass as the source joint, and the joint farthest from it as the target joint. Green dots represent vertices lacking a corresponding source.

4.2 Ensemble with multi-modal inputs

In this section we talk about the input modalities that we are using for our model. Using popular modalities for graphical skeleton data such as bones and joints we can train our model using each of these and then ensemble them during inference time. This representation gives us useful features by looking at how joints are positioned in relation to each other. We also use both position and velocity modalities, as both offer slightly different data points. The position modality is the default data which is the (x, y, z) values of each joint for all frames of our datasets. The velocity modality is the difference between corresponding joints in subsequent frames.

Shi et al. [97] introduced bone information, which is described as a vector pointing from its source joint to its target joint, indicating their physical connection, as depicted at $k = 1$ in Figure 4.7. Previous studies [13, 16, 71, 97] demonstrate that combining models trained with both bone and joint information significantly enhances action recognition performance. This suggests that these different ways of representing the skeleton complement each other. We utilize the same multi-modal skeleton representation to de-

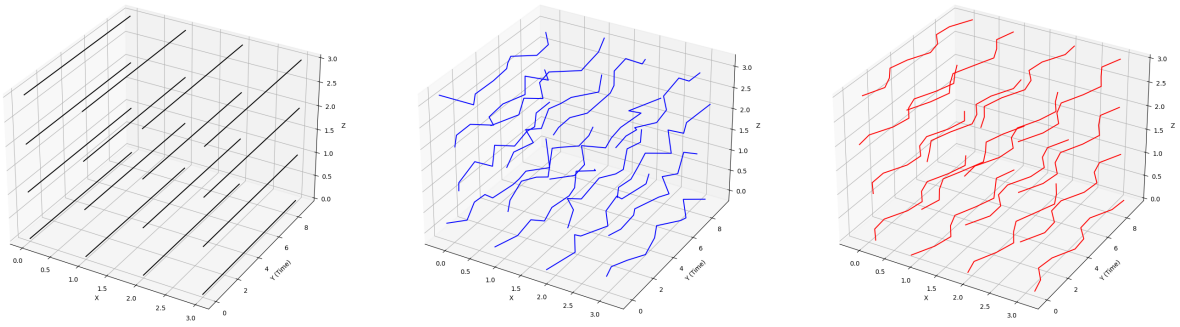

Figure 4.8: A visual representation comparison of random noise and Cosine-based noise on 2D points with time on the Y axis. (Left) Points with no noise. (Middle) Random noise applied. (Right) Cosine-based noise applied.

fine additional representations, building on the understanding that bone information can be perceived as a linear transformation of joint information. Specifically, we extend the joint-bone relationship at time t as

$$\mathbf{X}_t^{(k)} = (\mathbf{I} - \mathbf{P}^k)\mathbf{X}_t, \tag{4.6}$$

where $\mathbf{P} \in \mathbb{R}^{N \times N}$ denotes a binary matrix that contains source-target relations of the skeleton graph, with $\mathbf{P}_{ij} = 1$ if the i-th joint is the source of the j-th joint, and 0 otherwise. We set the row corresponding to the center of mass in $\mathbf{P}$ as a zero vector, ensuring that it does not have an associated source joint. We refer to $\mathbf{X}_t^{(k)}$ as the k-th mode representation of the skeleton. The representations for different k values capture distinct spatial features of a joint. We define K as $\max_v d(N) + 1$ for $n \in N$, where $d(n)$ gives the shortest distance in terms of hops between vertex N and the center of mass. When $k = 1$, the k-th mode representation $\mathbf{X}_t^{(k)}$ corresponds to the bone representation, as defined in [97], and when $k = K$, it corresponds to the joint representation, as $\mathbf{P}^k = 0$. For example, at $k = 1$ in Figure 4.7, the joint at the center of mass is shown as a green dot, making K equal to 5 in this instance.

4.3 Cosine-based Noise

In this work, to further improve our results, we proposed a noise augmentation strategy for skeleton based human action recognition using Cosine waves across time as noise instead of a random noise as shown in Figure 4.8. Each node in the skeleton undergoes a perturbation based on a Cosine wave, which adds controlled noise to the data. The noise has a randomly determined amplitude and frequency, modulated by a scaling factor. The intuition behind using such an augmentation strategy is to try and mitigate the random noise generated by either the sensor used to detect the skeleton joints or the model used to estimate skeleton pose data from RGB frame sequences. Applying a wave function over time on the nodes applies a similar perturbation across all nodes making the training process more robust to noise in the input data.

Let $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$ represent the skeleton data, where T is the number of time frames, N is the number of joints (nodes) in the skeleton, C is the number of channels (e.g., x, y, z coordinates). We apply a Cosine wave-based noise perturbation to each joint $n \in \{1, 2, \ldots, N\}$ of the skeleton as follows:

$$\mathbf{X}_n^{\text{noisy}} = \mathbf{X}_n + \alpha_n \cos(2\pi f_n t + \phi), \tag{4.7}$$

where $\mathbf{X}_n^{\text{noisy}} \in \mathbb{R}^{T \times C}$, $\alpha_n \in \mathbb{R}$ is the amplitude of the noise applied to node N, $f_n \in \mathbb{R}$ is the frequency of the Cosine wave applied to node n, $t \in \{1, 2, \ldots, T\}$ is the time frame index, $\phi \in \mathbb{R}$ is a phase shift. additionally, for each node n, the amplitude α and frequency f are drawn randomly from a normal distribution:

$$\alpha \sim \mathcal{N}(0, A), \quad f \sim \mathcal{U}(0, F) \tag{4.8}$$

where A and F are predefined scaling factors. The noise function is then applied equally to all nodes across the skeleton. The Cosine wave for each node is independent, resulting in a unique perturbation for each node at each time frame. The modified data is then

used as input for subsequent processing steps, with the added noise intended to improve the model's robustness or simulate real-world noise in sensor data. This approach adds controlled perturbations to skeleton data using Cosine wave noise. Each joint in the skeleton is perturbed independently with randomly chosen frequencies and amplitudes, ensuring diverse and realistic variations in the input data.

4.4 Summary

This chapter presents a comprehensive methodology for our model architecture aimed at human action recognition through skeleton data. Our core approach uses a GCN backbone with a partitioning transformer, which strategically partitions skeletal joints and frames according to specific joint-temporal relationships. This partitioning allows the model to leverage self-attention mechanisms tailored to focus on both global and local dependencies, thus enhancing the model's ability to learn intricate patterns associated with diverse human actions. Finally we have a classification head to classify the action.

The proposed methodology leverages multi-modal inputs by integrating bone and joint representations, enabling the model to better capture the relationship between joints. We also introduce a new data augmentation strategy using Cosine-based noise which is applied to all input modalities. This multi-modal ensemble approach enhances the model's capacity for action recognition, leading to more precise classifications. By emphasizing the spatial and temporal dynamics of human actions, the methodology significantly improves the model's overall performance and effectiveness.

Chapter 5

Experiments

In this section, we evaluate the effectiveness of the proposed method through a series of experiments. We begin by describing the datasets utilized in this study, followed by a comprehensive explanation of the evaluation methodology. Next, we perform an in-depth ablation study to investigate the contributions of various components of our model and their role in achieving results comparable to state-of-the-art approaches. Finally, we present and analyze the results obtained from these experiments

5.1 Datasets

For our model we decided to work on benchmarks which are used for multi-view and multi-subject action classification tasks as shown in Table 5.1.

Datasets	Subjects	Classes	Joints	Split	# Train	# Test	# Total
N-UCLA	10	10	20	-	1020	474	1,494
NTU60	40	60	25	Cross-Subject	40,091	16,487	56,578
	40	60	25	Cross-View	37,646	18,932	56,578
NTU120	106	120	25	Cross-Subject	63,026	50,919	113,945
	106	120	25	Cross-View	54,468	59,447	113,945

Table 5.1: Here is the breakdown of the key metrics for the benchmark datasets used.

Figure 5.1: Sample video frames from NTU60 and NTU120 datasets showing rgb frames of the actions being performed by different subjects from different camera angles and settings.

5.1.1 NTU-RGBD 60

The NTU RGB+D action recognition dataset [95] is collected using Microsoft Kinect v2 sensors and includes four distinct data modalities. Depth maps consist of sequences of 2D depth values in millimeters with a resolution of **512** $\times$ **424**, and lossless compression is applied to preserve information. 3D joint information provides 3D locations of **25** major body joints for detected humans, with corresponding pixels on both RGB frames and depth maps. RGB frames are videos recorded at a resolution of **1920** $\times$ **1080**, while infrared sequences are collected frame by frame at a resolution of **512** $\times$ **424**. These four modalities collectively provide a comprehensive multi-view representation of human actions for analysis.

The dataset includes **60** action classes, which are categorized into three groups. **40** daily actions represent common activities such as drinking, eating, and reading. **9** health-related actions involve behaviors like sneezing, staggering, and falling down. Additionally, **11** mutual actions capture interactions between individuals, such as punching, kicking,

and hugging. Data collection involved **40** subjects aged between **10** and **35** years, with each subject assigned a consistent ID across samples. The dataset captures a range of subject characteristics, including variations in age, gender, and height, ensuring a diverse representation.

For data collection, three cameras were used to capture the actions from different horizontal perspectives. These camera angles include **-45°**, **0°**, and **+45°**. The view types consist of two front views, one left side view, one right side view, one left side **45°** view, and one right side **45°** view. The camera setups were varied in both height and distance from the subjects, and detailed camera and setup numbers are provided for each video sample, allowing for a diverse set of perspectives in the dataset.

Two benchmark evaluation protocols are used for training and testing. In the cross-subject evaluation, subjects are divided into training and testing groups, each comprising **20** subjects. The training set contains **40,320** samples, while the testing set contains **16,560** samples, with specific training and testing subject IDs provided. In the cross-view evaluation, training is performed using samples from cameras 2 and 3, capturing front and side views, while testing is conducted on samples from camera 1, which captures the **45°** views. The training set consists of **37,920** samples, and the testing set includes **18,960** samples.

5.1.2 NTU-RGBD 120

The NTU RGB+D 120 dataset [70] is an extension of the NTU RGB+D dataset, maintaining similar characteristics with an increased number of action classes to **120** and a total of **114,480** videos. It is divided into an auxiliary set comprising **100** classes, allowing all samples from these classes for training, and a one-shot evaluation set consisting of **20** novel classes. In the evaluation set, one sample from each novel class serves as an exemplar, while the remaining samples are used for testing recognition performance. Evaluation protocols for this dataset mirror those of the NTU RGB+D dataset, with

cross-subject retaining the same designation and cross-view renamed to cross-setting in the NTU RGB+D dataset. To evaluate performance, classification accuracy is assessed under cross-subject and cross-view settings.

The dataset is collected using Microsoft Kinect sensors and includes four major data modalities. Depth maps consist of sequences of 2D depth values in millimeters, with a resolution of **512** $\times$ **424**, and lossless compression is applied to preserve data. 3D joint information provides 3D locations of **25** major body joints for each detected and tracked human, with corresponding pixels on both RGB frames and depth maps. RGB frames are recorded at a resolution of **1920** $\times$ **1080**, while infrared sequences are collected and stored frame by frame at a resolution of **512** $\times$ **424**.

The dataset includes **120** action categories, divided into **82** daily actions (examples include eating, writing, and moving objects), **12** health-related actions (examples include blowing nose, vomiting, and falling down), and **26** mutual actions (examples include handshaking, pushing, and hugging). Compared to NTU-60, this dataset features fine-grained hand/finger motions (such as "make ok sign" and "snapping fingers"), fine-grained object-related actions (such as "counting money" and "playing magic cube"), and object-related mutual actions (such as "wield knife towards other person" and "hit other person with object"). It also includes actions with similar postures but different speeds (such as "grab other person's stuff" vs. "touch other person's pocket (steal)"), similar body motions with different objects (such as "put on bag/backpack" vs. "put on jacket"), and similar objects with different body motions (such as "put on bag/backpack" vs. "take something out of a bag/backpack"). The dataset includes **106** subjects from **15** different countries, aged between **10** and **57** years, and ranging in height from **1.3m to 1.9m**. Each subject is assigned a consistent ID.

For data collection, **3** cameras capture three different horizontal views of the same action from angles of **-45°**, **0°**, and **+45°**. Each subject performs each action twice, providing a variety of views. View types include front, left side, right side, left side 45°,

and right side 45° views. Camera setup variations involve different heights and distances of cameras used across **32** setups, with all camera and setup numbers provided for each video sample.

Two benchmark evaluation protocols are used for training and testing. In the cross-subject evaluation, **106** subjects are divided into training and testing groups, each consisting of **53** subjects. The remaining subjects are reserved for testing. In the cross-setup evaluation, training uses samples from even collection setup IDs, while testing uses samples from odd setup IDs. **16** setups are used for training and **16** for testing.

5.1.3 NW-UCLA

The Northwestern UCLA Multiview Action 3D (NW-UCLA) dataset [125] is recorded using the MS Kinect version 1 sensor from multiple perspectives as shown in Figure 5.2. Training data are collected from **view 1** and **view 2**, while testing data are obtained from **view 3**. This dataset encompasses **10** action categories, including pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throwing, and carrying. Each action is executed by **10** subjects.

5.2 Implementation Details

5.2.1 Data Prepossessing

For the NTU RGB+D 60 and 120 datasets, we implement the pre-processing protocol outlined in [13, 16, 149], which involves aligning the skeletons' spines using a view-invariant transformation [108] to ensure they are perpendicular to the ground. For the NW-UCLA dataset, we adhere to the pre-processing protocol detailed in [13, 14, 16].

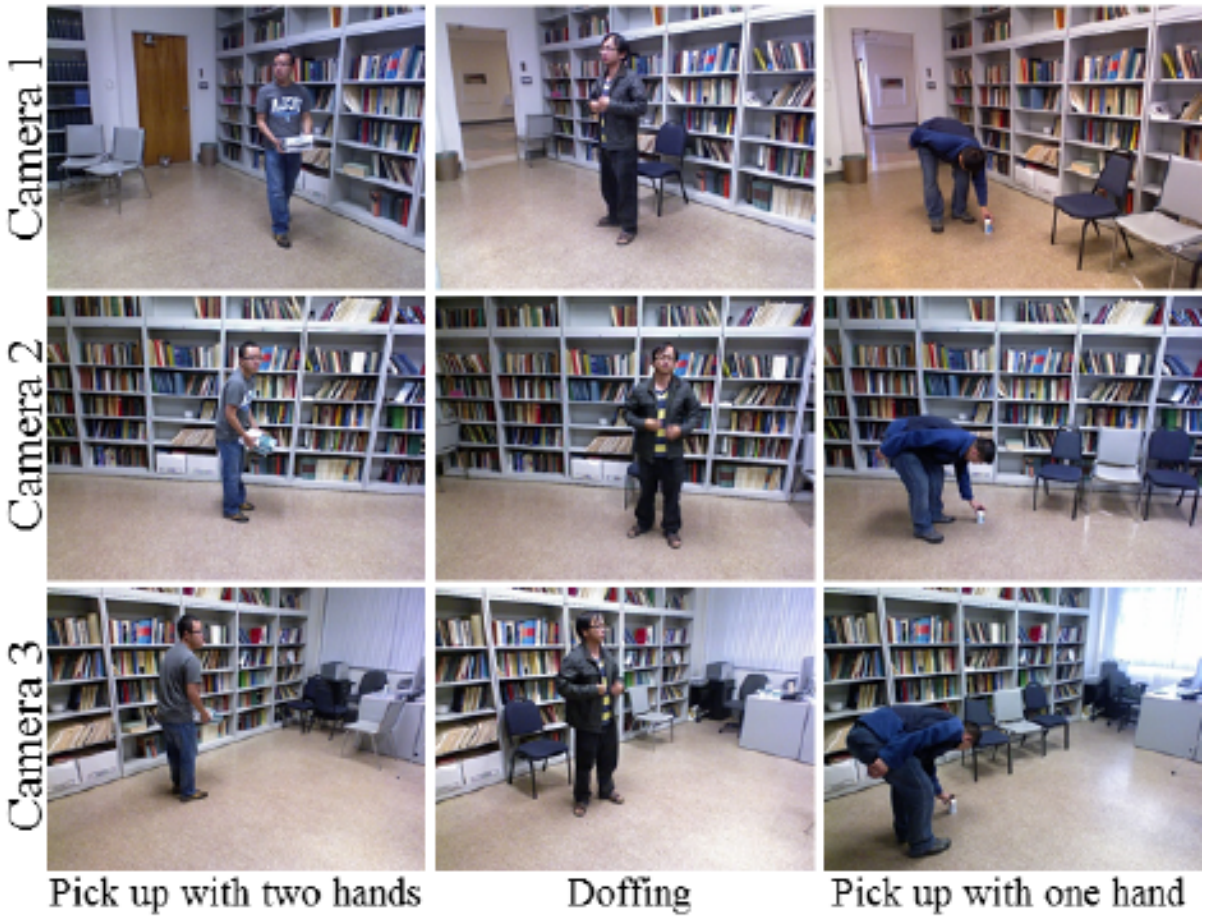

Figure 5.2: Sample video frames from NW-UCLA dataset showing RGB frames of the actions being performed from different camera angles.

5.2.2 Training

Our model is implemented using Pytorch [83] and training was performed on a Titan V100 GPU. We train our models for 130 epochs, using a warm-up strategy for the first 5 epochs, following the approach outlined in [16]. The learning rate is initialized to 0.1 for the NTU RGB+D 60 and 120 datasets and reduced to 0.05 for the NW-UCLA dataset. Learning rates decay with a factor of 0.1 at epochs 90, 100, and 120. For the NTU RGB+D 60 and 120 datasets, a weight decay of 5×10^{-4} is utilized, while for the NW-UCLA dataset, a weight decay of 4×10^{-4} is employed.

For Cosine-based noise generation, we use an amplitude of 0.008 and a frequency of 1 without damping. The batch size is configured to be approximately twice the number of classes, ensuring that, on average, each mini-batch contains data from two classes. This results in a batch size of 32 for NW-UCLA, 128 for NTU RGB+D 60, and 256 for NTU RGB+D 120.

For the NTU RGB+D and NTU RGB+D 120 datasets, the configuration for our partition transformer was: $V = 50$ (25 joints per individual), $T = 64$, $L = 4$, $K = 12$, $M = 8$, $N = 8$, and $C = 96$. For the NW-UCLA dataset, the configuration for our partition transformer was: $V = 20$, $T = 64$, $L = 4$, $K = 5$, $M = 8$, $N = 8$, and $C = 96$. For each partition, we have $H = 3$ heads.

5.3 Ablation Studies

To analyze the effects of individual components, we conducted multiple experiments. In this section, we present our findings regarding the various design choices made in our model. We discuss the impact of our novel noise strategy using Cosine waves and compare it with random noise data augmentation. Additionally, we report the results of using different joint modalities, along with position and velocity modalities. Finally, we examine the effects of employing both a vanilla and a partitioning transformer. All of

	Position	Velocity	$Pos + \epsilon$	$Pos + \epsilon_{Cosine}$	$Vel + \epsilon$	$Vel + \epsilon_{Cosine}$
K=1 (joints)	90.31%	88.27%	89.03%	**89.81%**	85.63%	**86.15%**
K=2	90.24%	88.55%	88.51%	**89.95%**	86.38%	**87.05%**
K=3	89.71%	88.24%	88.96%	**89.63%**	86.67%	**87.70%**
K=4	89.48%	88.00%	88.35%	**89.24%**	86.42%	**87.59%**
K=5	89.28%	87.95%	87.83%	**88.46%**	86.61%	**87.73%**
K=6	89.17%	87.62%	87.64%	**88.31%**	86.47%	**87.68%**
K=7	89.24%	88.39%	88.21%	**89.11%**	86.53%	**87.65%**
K=8 (bones)	89.42%	88.27%	88.10%	**89.05%**	86.38%	**87.82%**

Table 5.2: Results for NTU60 Cross Subject split using various input modalities with both random noise and Cosine-based noise. Bold numbers represent better performance using data augmentations.

our ablative studies were performed on the NTU-60 dataset.

5.3.1 Cosine-based Noise

For evaluating the effectiveness of our novel Cosine-based noise data augmentation, we ran our experiments on NTU-60 cross subject split, training each run for 130 epochs, with 0.1 starting learning rate, with a 0.1 decay at 90, 100 and 120 steps, with weight decay of 5×10^{-4}. Table 5.2 shows the results of our experiments. From the results we can see clearly that for every case, the Cosine-based noise performs better than the random noise using the same amplitude of noise, which for our experiments was set to 0.008 and the frequency for the Cosine-based noise was set to 1 without damping.

5.3.2 Vanilla vs partitioning Transformer

We applied various partitioning strategies and our results are shown in Table 5.3. We applied three different partitioning strategies, in the first one we only partitioned on the joints, in the second approach we only partitioned on the frames and on the third strategy we applied both, creating four partitions which produced the best results, showing that paying attention to local and global aspect of the skeleton structure and time frames both are crucial in distinguishing complex actions.

Model	NTU-60	
	CS	CV
Vanilla Transformer	93.11	97.41
2-Partition Transformer (local joints), (distant joints)	93.36	97.52
2-Partition Transformer (local frames) , (distant frames)	93.54	97.67
4-Partition Transformer (local joints, local frames)(local joints, distant frames) (distant joints, local frames)(distant joints, distant frames)	**94.46**†	**97.87**†

Table 5.3: Results on NTU60 Cross Subject and Cross View split using various partitioning strategies. † represents the highest accuracy for that split.

N-models	Ensemble cases	Accuracy
2-models	(pos, vel) K={1} (bones)	91.08%
2-models	(pos, vel) K={8} (joints)	90.47%
4-models	(pos, vel) K={1,2}	92.01%
6-models	(pos, vel) K={1,2,8}	92.35%
6-models	(pos, vel) K={2,8}, (pos+ϵ, vel+ϵ) K={1}	92.48%
8-models	(pos, vel) K={1,2,3,8}	92.23%
2-models	(pos+ϵ, vel+ϵ) K={1}	91.79%
4-models	(pos+ϵ, vel+ϵ) K={1,2}	92.12%
6-models	(pos+ϵ, vel+ϵ) K={1,2,8}	92.63%
6-models	(pos+ϵ_{Cosine}, vel+ϵ_{Cosine}) K={1,2,8}	**93.46%**†

Table 5.4: Results for NTU60 Cross Subject split using various ensembling combinations with both random noise and Cosine-based noise, different K values and both position and velocity modalities. † represents the best result.

5.3.3 Multi-modal representation

We perform a comparative analysis of ensemble models trained using various combinations of modalities. This highlights the importance of multi-modal representations in enhancing the diversity of input features and the corresponding trained models, thereby increasing the effectiveness of the ensemble. Notably, the accuracy tends to plateau beyond the inclusion of six modalities.

As the parameter k increases, the number of vertices lacking a source also rises, as illustrated by the green dots in Figure5.3, resulting in a lack of distinctive features.

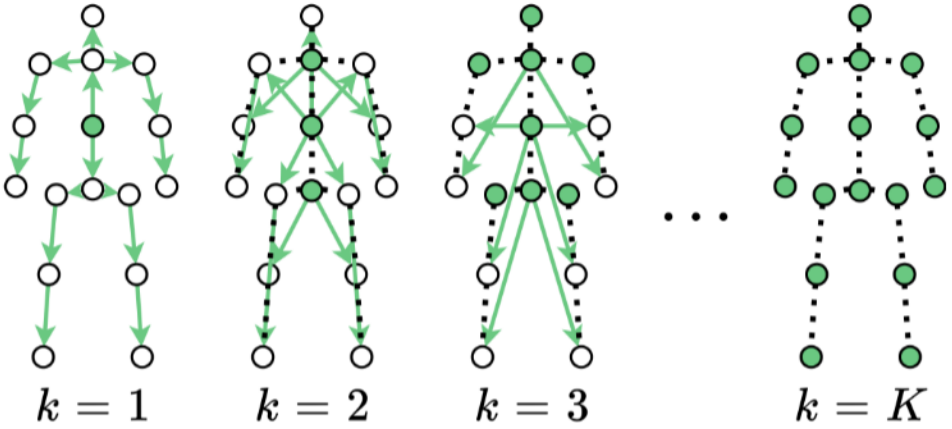

Figure 5.3: Demonstration of Multi-modal Skeleton Representation: Arrows depict the k-th mode representation of pointed vertices. We designate the joint closest to the center of mass as the source joint, and the joint farthest from it as the target joint. Green dots represent vertices lacking a corresponding source. K=8 for NTU-60 and NTU-120 and K=5 for NW-UCLA.

Additionally, we incorporate both position and velocity modalities, each providing unique data points. The position modality consists of the (x, y, z) coordinates of each joint across all frames in our datasets, while the velocity modality captures the differences between corresponding joints in subsequent frames.

As demonstrated in Table 5.4, the optimal ensemble configuration consists of six models utilizing $k = 1, 2, 8$, incorporating both position and velocity inputs, and applying Cosine-based noise across all instances. This configuration consistently yields the best performance, and this ensemble model, referred to as "**Hybrid-Graformer**" is used to report our results in the subsequent section.

5.4 Experimental Results and Discussion

Table 5.5 shows the results for various approaches on the NTU-60, NTU-120, and NW-UCLA datasets. We compare our model with state-of-the-art approaches [87, 24, 66, 126, 153] using skeleton data for action recognition. For the other methods, we report the results as stated in their respective papers without reproducing them ourselves. While all

methods use the same data splits as specified by the dataset owners, other experimental setups, such as validation sets and ensemble configurations, differ. Our model achieves results across all three benchmarks that are on par with or surpass current state-of-the-art approaches, demonstrating the clear efficacy of our method. Each result was obtained by training across multiple modalities, with the Hybrid-Graformer ensemble yielding the best overall performance.

5.4.1 Analyzing Results

When we analyze our prediction results we can see a clear trend in the classes which our model misclassifies the most. For example the top most misclassified classes are the following: writing, typing on a keyboard, reading, playing with phone/tablet, wear a shoe, take off a shoe, touch head (headache), eat meal/snack, rub two hands together, sneeze/cough, touch chest (stomachache/heart pain), pointing to something with finger, touch neck (neckache), nausea or vomiting condition, brushing hair, make a phone call/answer phone. These top misclassified predictions can be broadly categorized into two main types of interactions: human-object interaction or actions requiring higher resolution (especially finger joints). If we extract the top most misclassified class pairs we see this hypothesis further confirmed as seen on Table 5.6.

The results show two clear gaps in our model. Firstly, human object interaction is not fully captured, and that is true, we only consider human skeleton joints as inputs which lose the object information. For cases like "Typing" and "Writing" since our model does not have any contextual knowledge of a keyboard or a pen, it fails to differentiate between the two. similarly for examples like "wearing a shoe" and "taking off a shoe," both actions seem similar when only seeing the skeleton and not having any contextual input for the shoe. Another key gap in our model is the lack of higher resolution in the joints. Our model uses 25 key points all over the human body, so if we have actions "rubbing hands together" and "clapping" where we only see the up to the palm joint, it

Type	Methods	NTU-60		NTU-120		NW-UCLA
		CS	CV	CS	CV	
GNNs	Shift-GCN [CVPR'20]	90.7	96.5	85.9	87.6	94.6
	DC-GCN+ADG	90.8	96.6	86.5	88.1	95.3
	PA-ResGCN-B19	90.9	96.0	87.3	88.3	
	DDGCN	91.1	97.1			
	Dynamic GCN	91.5	96.0	87.3	88.6	
	MS-G3D [CVPR'20]	91.5	96.2			
	MST-GCN	91.5	96.6	87.5	88.8	
	CTR-GCN [ICCV'21]	92.4	96.8	88.9	90.6	96.5
	infoGCN [CVPR'22]	93.0	97.1	89.8	91.2	97.0
	LA-GCN [CoRP'23]	93.5	97.1	90.7	91.8	97.6
CNNs	PoseC3D [CVPR'22]	94.1	97.1	86.9	90.3	
Hypergraphs	Hyper-GNN [TIP'21]	89.5	95.7			
	DHGCN [CoRR'21]	90.7	96.0	86.0	87.9	
	SD-HGCN[ICONIP'21]	90.9	96.7	87.0	88.2	
	S-HCN [ICMR'22]	90.8	96.6			
Transformers	ST-TR [CVIU'21]	90.3	96.3	85.1	87.1	
	MTT [LSP'21]	90.8	96.7	86.1	87.6	
	STST [ACM MM'21]	91.9	96.8			
	4s-GSTN[Symmetry'22]	91.3	96.6	86.4	88.7	
	FGSTFormer[ACCV'22]	92.6	96.7	89.0	90.6	97.0
	Hyperformer [2022]	92.9	96.5	89.9	91.3	96.7
	3MFormer [CVPR'23]	**94.6**†	97.7	**91.2**†	**93.5***	97.8
Ours	Hybrid-Graformer	94.46	**97.87**†	90.68	**93.52***	**97.91**†

Table 5.5: Results for various methods on NTU-60, NTU120 and NW-UCLA datasets including our method at the end. † represents the best result for that split and * represent results that are comparable in that split.

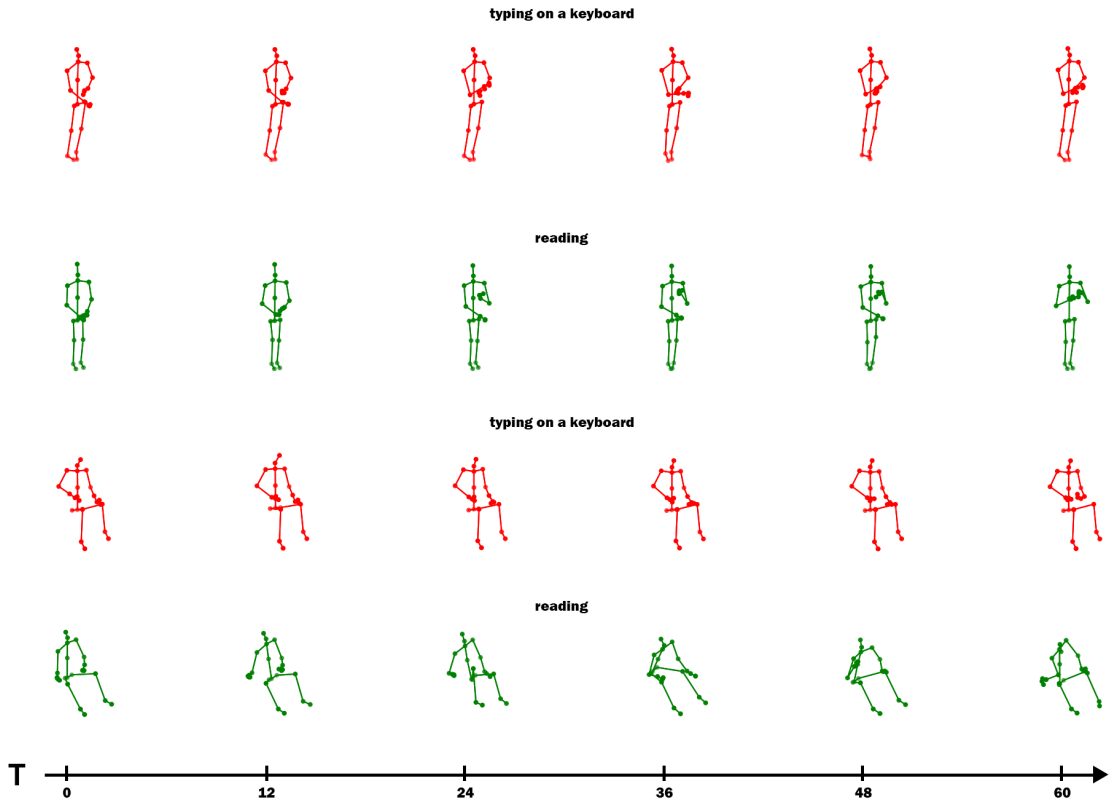

Figure 5.4: Plotting the two top misclassified action classes show the issue with misclassifications. The action frame time increases from left to right. Red frames are from "typing on a keyboard" class and green frames are from "reading" class. Both classes look like similar actions without the context of the object being interacted with.

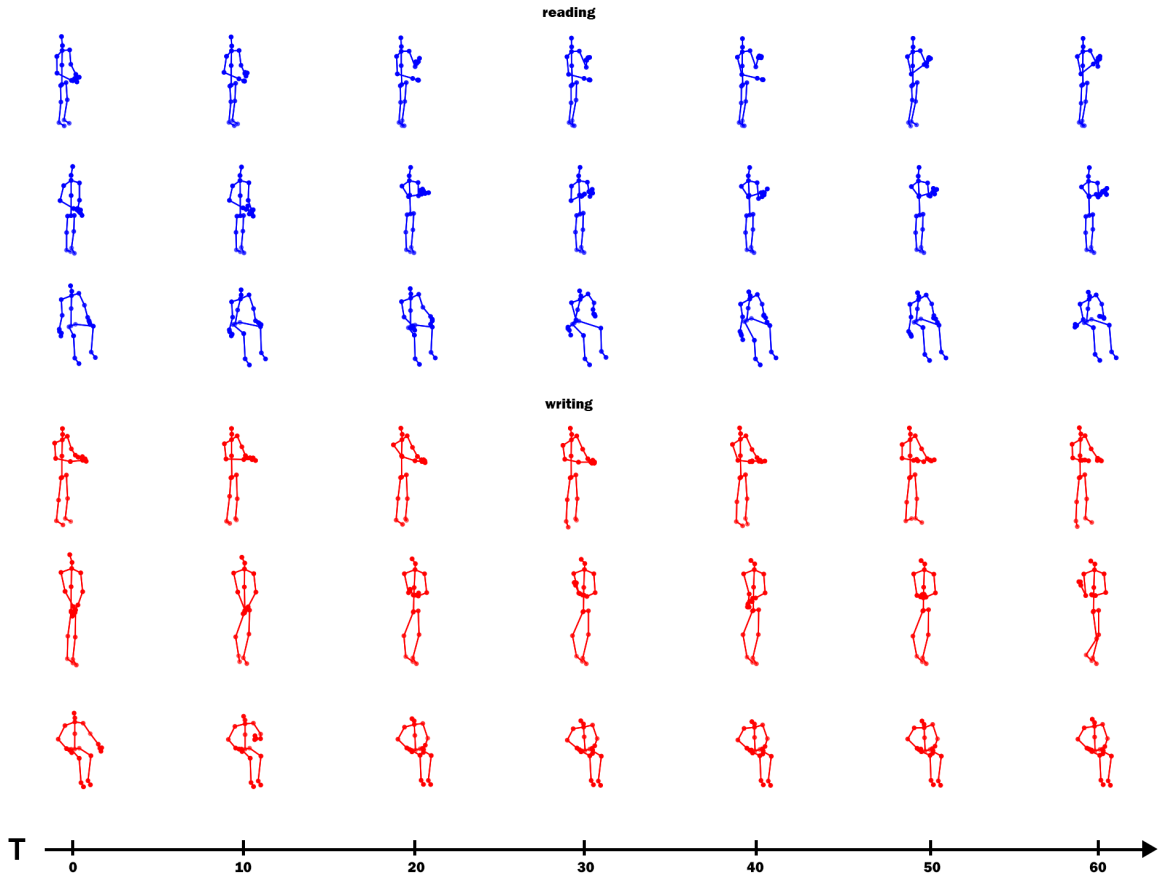

Figure 5.5: Human object interaction: A set of class pairs where without the context of object or higher resolution in finger joints makes it difficult to predict accurately. Top blue frames are "reading" and the red color is "writing".

True Class	Predicted Class
typing on a keyboard	writing
writing	reading
writing	typing on a keyboard
reading	writing
wear a shoe	take off a shoe
take off a shoe	wear a shoe
writing	playing with phone/tablet
playing with phone/tablet	writing
playing with phone/tablet	typing on a keyboard
typing on a keyboard	reading
touch head (headache)	wipe face
kicking other person	kicking something
typing on a keyboard	playing with phone/tablet
rub two hands together	clapping
touch chest (stomachache/heart pain)	nausea or vomiting condition
nausea or vomiting condition	sneeze/cough
playing with phone/tablet	reading
pointing to something with finger	taking a selfie

Table 5.6: Top most misclassified pairs of classes.

is not able to differentiate effectively. A detailed breakdown of the results, along with detailed truth vs. prediction plots and further analysis, is provided in Appendix A.

5.4.2 Model Parameters

We analyze our models parameters and compare them with other models in Table 5.7. Our model maintains a comparable number of parameters with respect to other state-of-the-art methods at 3.78M. Individually our GCN backbone is 1.5M parameters and our Transformer block is 2.28M in size.

5.5 Summary

In this chapter, we conducted an extensive series of experiments to evaluate the performance and limitations of our proposed model for human action recognition using

Type	Methods	NTU-60		Params (M)
		CS	CV	
GNNs	DC-GCN+ADG	90.8	96.6	4.90
	MS-G3D [CVPR'20]	91.5	96.2	3.22
	MST-GCN	91.5	96.6	12.00
	CTR-GCN [ICCV'21]	92.4	96.8	**1.46***
	infoGCN [CVPR'22]	93.0	97.1	1.56
	HD-GCN [ICCV'23]	93.4	97.2	1.68
CNNs	PoseC3D [CVPR'22]	94.1	97.1	2.00
Transformers	ST-TR [CVIU'21]	90.3	96.3	12.10
	DSTA [ACCV'20]	91.5	96.4	4.10
	Hyperformer [2022]	92.9	96.5	2.71
	3MFormer [CVPR'23]	**94.6**†	97.7	4.37
Ours	Hybrid-Graformer	94.46	**97.87**†	3.78

Table 5.7: A comparison of the number of model parameters on NTU-60 represented in millions. GNNs in general have a low parameter count, the only CNN performing at par with state of the art models also has modest parameters and Transformers generally have higher parameters. † represents the best accuracy for that split and * represents the lowest parameter count.

skeleton data. The ablation studies concentrated on data augmentation strategies, input data modalities, and transformer architectures. Our findings revealed that the best performance was achieved with a combination of Cosine-based noise augmentation, an ensemble model incorporating both positional and velocity information from skeleton modalities, and the partitioning transformer.

We compared our approach (Hybrid-Graformer) against state-of-the-art methods, demonstrating competitive results. However, our analysis identified two key limitations: The model struggled to consistently capture human-object interactions and secondly, the joint resolution used was insufficient for recognizing actions involving fine-grained finger movements.

Chapter 6

Conclusion

In this thesis, we present a novel deep learning architecture that integrates a GCN backbone with a partitioning transformer. The GCN backbone, enhanced by attention mechanisms, effectively captures the context-dependent structure of human skeletal topology while amplifying discriminative features. The transformer component complements this by aggregating long-range temporal dependencies, enabling the model to accurately recognize complex actions across both short- and long-term temporal windows. This is made possible through our partitioning strategy, which efficiently models the relationships between both neighboring and distant joints, facilitating a comprehensive understanding of human movement dynamics.

Furthermore, we introduce a Cosine-based noise augmentation strategy that enhances the model's robustness and accuracy. Our proposed Hybrid-Graformer model demonstrates competitive results across multiple skeleton-based action recognition benchmarks.

6.1 Limitations and Future Work

While our approach delivers results comparable to state-of-the-art methods, several limitations still exist, particularly in handling "human-object interactions" and "joint resolution." Our current model lacks explicit modeling of interactions between humans and

objects, which is crucial for certain action recognition tasks. To address this, fusing RGB information with skeleton data could provide richer contextual information about objects involved in actions, potentially improving recognition accuracy in object-related activities. The resolution of skeletal joints remains another limitation. Using more advanced pose estimation models, as discussed in recent literature [93, 152], could yield more accurate and detailed joint data, improving the model's overall performance in complex motions.

Additionally, we can include more data points such as joint names in our input feature which can potentially improve the understanding of the model about how different body joints interact with each other for different actions.

For future work, several promising directions emerge from this study. The superior performance of Cosine-based noise augmentation over random noise highlights its potential for further development. Future work could refine this technique, exploring different noise schedules or hybrid augmentation methods to further boost model robustness. Incorporating 2D or 3D volumetric heat maps of joints as an additional input modality is another avenue worth exploring. This could provide more granular spatial information about joint positions, potentially improving prediction stability and overall performance.

Our current model utilizes an ensemble of six to twelve individual models. Recent advances in multi-modal fusion architectures, such as those in [126, 24], have demonstrated the ability to integrate multiple modalities at earlier stages of training, offering enhanced performance without the complexity and computational overhead of large ensembles. Transitioning to such architectures could simplify the model while maintaining or improving performance.

While our model performs competitively across three datasets, it has yet to be evaluated on large-scale datasets such as Kinetics-400 [48], which contains 400 action categories. Future research should focus on testing the scalability of the model to handle a broader range of classes and larger batch sizes.

Additionally, although currently focused on human skeleton modeling, the framework of this model is generalizable to other forms of structured data. For example, it could be adapted to track the motion of particles, articulated objects, or other systems with structured dependencies, broadening its application domain.

6.2 Social and Ethical Implications

The use of detailed bio-metric data, such as skeleton data in our model, provides insights into an individual's body posture and movement patterns. This can reveal sensitive information about physical and behavioral traits, raising significant privacy concerns. Given the sensitivity of such data, informed consent is crucial. Users need to understand what data is being collected, how it is used, and the potential risks associated with use.

The accuracy of skeleton data enhances the detection of specific actions and behaviors, leading to improvements in identifying criminal activities or emergencies. This data can contribute to public safety by enabling more precise detection of criminal activity and supporting rapid response during emergency situations. Advanced human action recognition models offer valuable insights into individuals' physical states and movements during critical moments.

On the positive side, these models significantly impact assistive technologies, such as personalized rehabilitation programs, fitness solutions, and ergonomic designs. Additionally, better movement analysis can improve healthcare by aiding diagnostics and treatment plans, and it supports research into understanding physical health and behavior but such surveillance comes with its own social cost.

However, balancing these benefits requires implementation of strong data protection measures, including secure storage, encryption, and strict access controls to safeguard sensitive data. Furthermore, the development of ethical use guidelines is essential to ensure collection, use, and sharing of skeleton data respects individuals' rights and privacy.

Bibliography

[1] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A. Alemi. Watch your step: Learning node embeddings via graph attention. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9198–9208, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8a94ecfa54dcb88a2fa993bfa6388f9e-Abstract.html.

[2] Tasweer Ahmad, Lianwen Jin, Xin Zhang, Songxuan Lai, Guozhi Tang, and Luojun Lin. Graph convolutional neural network for human action recognition: A comprehensive survey. *IEEE Transactions on Artificial Intelligence*, 2(2):128–145, 2021. doi: 10.1109/TAI.2021.3076974.

[3] Amr Ahmed, Nino Shervashidze, Shravan M. Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. Distributed large-scale natural graph factorization. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 37–48. International World Wide Web Conferences Steering Committee / ACM, 2013. doi: 10.1145/2488388.2488393. URL https://doi.org/10.1145/2488388.2488393.

[4] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: transformers for multimodal self-supervised learning from raw video, audio and text. *CoRR*, abs/2104.11178, 2021. URL https://arxiv.org/abs/2104.11178.

[5] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

[6] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402 Vol. 2, 2005. doi: 10.1109/ICCV.2005.28.

[8] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402 Vol. 2, 2005. doi: 10.1109/ICCV.2005.28.

[9] Carlos Caetano, Jessica Sena, François Brémond, Jefersson A. dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. *CoRR*, abs/1907.13025, 2019. URL http://arxiv.org/abs/1907.13025.

[10] Deng Cai and Wai Lam. Graph transformer for graph-to-sequence learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7464–7471. AAAI

Press, 2020. doi: 10.1609/AAAI.V34I05.6243. URL https://doi.org/10.1609/aaai.v34i05.6243.

[11] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017. URL http://arxiv.org/abs/1705.07750.

[12] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C.-C. Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020. doi: 10.1017/ATSIP.2020.13.

[13] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. *CoRR*, abs/2107.12213, 2021. URL https://arxiv.org/abs/2107.12213.

[14] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 180–189, 2020. URL https://api.semanticscholar.org/CorpusID:219964813.

[15] Yi-Bin Cheng, Xipeng Chen, Dongyu Zhang, and Liang Lin. Motion-transformer: self-supervised pre-training for skeleton-based action recognition. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, MMAsia '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383080. doi: 10.1145/3444685.3446289. URL https://doi.org/10.1145/3444685.3446289.

[16] Hyung-Gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based

action recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20154–20164, 2022. doi: 10.1109/CVPR52688.2022.01955.

[17] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3273–3280. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995707. URL https://doi.org/10.1109/CVPR.2011.5995707.

[18] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893. IEEE Computer Society, 2005. doi: 10.1109/CVPR.2005.177. URL https://doi.org/10.1109/CVPR.2005.177.

[19] Jeonghyeok Do and Munchurl Kim. Skateformer: Skeletal-temporal transformer for human action recognition. *CoRR*, abs/2403.09508, 2024. doi: 10.48550/ARXIV.2403.09508. URL https://doi.org/10.48550/arXiv.2403.09508.

[20] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014. URL http://arxiv.org/abs/1411.4389.

[21] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 135–144. ACM, 2017. doi: 10.1145/3097983.3098036. URL https://doi.org/10.1145/3097983.3098036.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[24] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. *CoRR*, abs/2104.13586, 2021. URL https://arxiv.org/abs/2104.13586.

[25] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition. *ArXiv*, abs/2210.05895, 2022. URL https://api.semanticscholar.org/CorpusID:252846543.

[26] Thi V. Duong, Hung Hai Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 838–845. IEEE Computer Society, 2005. doi: 10.1109/CVPR.2005.61. URL https://doi.org/10.1109/CVPR.2005.61.

[27] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020. URL https://arxiv.org/abs/2012.09699.

[28] Claudio Fanti, Lihi Zelnik-Manor, and Pietro Perona. Hybrid models for human motion recognition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 1166–1173. IEEE Computer Society, 2005. doi: 10.1109/CVPR.2005.179. URL https://doi.org/10.1109/CVPR.2005.179.

[29] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *CoRR*, abs/1604.06573, 2016. URL http://arxiv.org/abs/1604.06573.

[30] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[31] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *CoRR*, abs/1812.03982, 2018. URL http://arxiv.org/abs/1812.03982.

[32] B. Fernando, P. Anderson, M. Hutter, and S. Gould. Discriminative hierarchical rank pooling for activity recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1932, 2016. doi: 10.1109/CVPR.2016.212.

[33] Daniel R. Figueiredo, Leonardo Filipe Rodrigues Ribeiro, and Pedro H. P. Saverese. struc2vec: Learning node representations from structural identity. *CoRR*, abs/1704.03165, 2017. URL http://arxiv.org/abs/1704.03165.

[34] Tao-Yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng,

Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li, editors, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1797–1806. ACM, 2017. doi: 10.1145/3132847.3132953. URL https://doi.org/10.1145/3132847.3132953.

[35] Zhimin Gao, Peitao Wang, Pei Lv, Xiaoheng Jiang, Qidong Liu, Pichao Wang, Mingliang Xu, and Wanqing Li. Focal and global spatial-temporal transformer for skeleton-based action recognition. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 382–398, December 2022.

[36] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970. ISSN 0945-3245. doi: 10.1007/BF02163027. URL https://doi.org/10.1007/BF02163027.

[37] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007. doi: 10.1109/TPAMI.2007.70711.

[38] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2018.03.022. URL https://www.sciencedirect.com/science/article/pii/S0950705118301540.

[39] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016. URL http://arxiv.org/abs/1607.00653.

[40] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long*

Beach, CA, USA, pages 1024–1034, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

[41] Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in neural information processing systems*, 16, 2003.

[42] Lianyu Hu, Shenglan Liu, and Wei Feng. Spatial temporal graph attention network for skeleton-based action recognition, 2022.

[43] Md. Shamim Hussain, Mohammed J. Zaki, and Dharmashankar Subramanian. Edge-augmented graph transformers: Global self-attention is enough for graphs. *CoRR*, abs/2108.03348, 2021. URL https://arxiv.org/abs/2108.03348.

[44] Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. *CoRR*, abs/1511.05298, 2015. URL http://arxiv.org/abs/1511.05298.

[45] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. doi: 10.1109/TPAMI.2012.59.

[46] Kui Jia and Dit-Yan Yeung. Human action recognition using local spatio-temporal discriminant embedding. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008. doi: 10.1109/CVPR.2008.4587732. URL https://doi.org/10.1109/CVPR.2008.4587732.

[47] Rui Jiang, Weijie Fu, Li Wen, Shijie Hao, and Richang Hong. Dimensionality reduction on anchorgraph with an efficient locality preserving projection. *Neurocomputing*, 187:109–118, 2016. doi: 10.1016/J.NEUCOM.2015.07.128. URL https://doi.org/10.1016/j.neucom.2015.07.128.

[48] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. URL https://arxiv.org/abs/1705.06950.

[49] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Comput. Surv.*, 54(10s), sep 2022. ISSN 0360-0300. doi: 10.1145/3505244. URL https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3505244.

[50] Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. Interpretable rumor detection in microblogs by attending to user interactions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8783–8790. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6405. URL https://doi.org/10.1609/aaai.v34i05.6405.

[51] Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. Transformers generalize deepsets and can be extended to graphs and hypergraphs. *CoRR*, abs/2110.14416, 2021. URL https://arxiv.org/abs/2110.14416.

[52] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. *CoRR*, abs/1704.04516, 2017. URL http://arxiv.org/abs/1704.04516.

[53] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

[54] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

[55] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, May 2022. ISSN 1573-1405. doi: 10.1007/s11263-022-01594-9. URL https://doi.org/10.1007/s11263-022-01594-9.

[56] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21618–21629, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b4fd1d2cb085390fbbadae65e07876a7-Abstract.html.

[57] Laptev and Lindeberg. Space-time interest points. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 432–439 vol.1, 2003. doi: 10.1109/ICCV.2003.1238378.

[58] Ivan Laptev. On space-time interest points. *Int. J. Comput. Vis.*, 64(2-3):107–123, 2005. doi: 10.1007/S11263-005-1838-7. URL https://doi.org/10.1007/s11263-005-1838-7.

[59] Ivan Laptev and Patrick Pérez. Retrieving actions in movies. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil,*

October 14-20, 2007, pages 1–8. IEEE Computer Society, 2007. doi: 10.1109/ICCV.2007.4409105. URL https://doi.org/10.1109/ICCV.2007.4409105.

[60] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008. doi: 10.1109/CVPR.2008.4587756. URL https://doi.org/10.1109/CVPR.2008.4587756.

[61] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3361–3368. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995496. URL https://doi.org/10.1109/CVPR.2011.5995496.

[62] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[63] John Boaz Lee, Ryan A. Rossi, and Xiangnan Kong. Graph classification using structural attention. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1666–1674. ACM, 2018. doi: 10.1145/3219819.3219980. URL https://doi.org/10.1145/3219819.3219980.

[64] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition, 2023.

[65] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recog-

nition with convolutional neural networks. *CoRR*, abs/1704.07595, 2017. URL http://arxiv.org/abs/1704.07595.

[66] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 786–792. ijcai.org, 2018. doi: 10.24963/IJCAI.2018/109. URL https://doi.org/10.24963/ijcai.2018/109.

[67] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Unsupervised learning of view-invariant action representations. *CoRR*, abs/1809.01844, 2018. URL http://arxiv.org/abs/1809.01844.

[68] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12919–12928. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01270. URL https://doi.org/10.1109/ICCV48922.2021.01270.

[69] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3337–3344. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995353. URL https://doi.org/10.1109/CVPR.2011.5995353.

[70] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10): 2684–2701, 2020. doi: 10.1109/TPAMI.2019.2916873.

[71] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang.

Disentangling and unifying graph convolutions for skeleton-based action recognition, 2020.

[72] Guan Luo, Shuang Yang, Guodong Tian, Chunfeng Yuan, Weiming Hu, and Stephen J. Maybank. Learning human actions by combining global dynamics and local appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(12):2466–2482, 2014. doi: 10.1109/TPAMI.2014.2329301. URL https://doi.org/10.1109/TPAMI.2014.2329301.

[73] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. *CoRR*, abs/1701.01821, 2017. URL http://arxiv.org/abs/1701.01821.

[74] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2929–2936. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206557. URL https://doi.org/10.1109/CVPR.2009.5206557.

[75] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge. Action transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, 124:108487, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108487. URL https://www.sciencedirect.com/science/article/pii/S0031320321006634.

[76] Yue Meng, Mengqi Shi, and Wenlu Yang. Skeleton action recognition based on transformer adaptive graph convolution. *Journal of Physics: Conference Series*, 2170(1):012007, feb 2022. doi: 10.1088/1742-6596/2170/1/012007. URL https://dx.doi.org/10.1088/1742-6596/2170/1/012007.

[77] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit:

Encoding graph structure in transformers. *CoRR*, abs/2106.05667, 2021. URL https://arxiv.org/abs/2106.05667.

[78] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

[79] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 969–976. ACM, 2018. doi: 10.1145/3184558.3191526. URL https://doi.org/10.1145/3184558.3191526.

[80] Juan Carlos Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383132. URL https://doi.org/10.1109/CVPR.2007.383132.

[81] Antonio Ortega, Pascal Frossard, Jelena Kovacevic, José M. F. Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE*, 106(5):808–828, 2018. doi: 10.1109/JPROC.2018.2820126. URL https://doi.org/10.1109/JPROC.2018.2820126.

[82] Yunsheng Pang, Qiuhong Ke, Hossein Rahmani, James Bailey, and Jun Liu. Igformer: Interaction graph transformer for skeleton-based human interaction recognition. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria

Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 605–622, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19806-9.

[83] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017. URL https://openreview.net/forum?id=BJJsrmfCZ.

[84] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2669–2676, Apr. 2020. doi: 10.1609/aaai.v34i03.5652. URL https://ojs.aaai.org/index.php/AAAI/article/view/5652.

[85] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014. URL http://arxiv.org/abs/1403.6652.

[86] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. *CoRR*, abs/1711.10305, 2017. URL http://arxiv.org/abs/1711.10305.

[87] Tung Nguyen Quang and Thi-Oanh Nguyen. Language knowledge-assisted in topology construction for skeleton-based action recognition. In *Proceedings of the 12th International Symposium on Information and Communication Technology, SOICT 2023, Ho Chi Minh, Vietnam, December 7-8, 2023*, pages 443–449. ACM, 2023. doi: 10.1145/3628797.3629008. URL https://doi.org/10.1145/3628797.3629008.

[88] Stjepan Rajko, Gang Qian, Todd Ingalls, and Jodi James. Real-time gesture recognition with minimal training requirements and on-line learning. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*

2007), 18-23 June 2007, Minneapolis, Minnesota, USA. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383330. URL https://doi.org/10.1109/CVPR.2007.383330.

[89] Kanchana Ranasinghe, Muzammal Naseer, Salman H. Khan, Fahad Shahbaz Khan, and Michael S. Ryoo. Self-supervised video transformer. *CoRR*, abs/2112.01514, 2021. URL https://arxiv.org/abs/2112.01514.

[90] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/94aef38441efa3380a3bed3faf1f9d5d-Abstract.html.

[91] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[92] Aliaksei Sandryhaila and José M. F. Moura. Discrete signal processing on graphs. *IEEE Trans. Signal Process.*, 61(7):1644–1656, 2013. doi: 10.1109/TSP.2013.2238935. URL https://doi.org/10.1109/TSP.2013.2238935.

[93] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2016.09.002. URL https://www.sciencedirect.com/science/article/pii/S1077314216301369.

[94] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004. doi: 10.1109/ICPR.2004.1334462.

[95] A. Shahroudy, J. Liu, T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.115. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.115.

[96] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015. URL http://arxiv.org/abs/1511.04119.

[97] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Adaptive spectral graph convolutional networks for skeleton-based action recognition. *CoRR*, abs/1805.07694, 2018. URL http://arxiv.org/abs/1805.07694.

[98] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Adaptive spectral graph convolutional networks for skeleton-based action recognition. *CoRR*, abs/1805.07694, 2018. URL http://arxiv.org/abs/1805.07694.

[99] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[100] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Asian Conference on Computer Vision*, 2020. URL https://api.semanticscholar.org/CorpusID:229622392.

[101] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015. URL http://arxiv.org/abs/1506.04214.

[102] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192. URL https://doi.org/10.1109/MSP.2012.2235192.

[103] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. *CoRR*, abs/1805.02335, 2018. URL http://arxiv.org/abs/1805.02335.

[104] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. *CoRR*, abs/1902.09130, 2019. URL http://arxiv.org/abs/1902.09130.

[105] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition, 2019.

[106] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014. URL http://arxiv.org/abs/1406.2199.

[107] Cristian Sminchisescu, Atul Kanaujia, and Dimitris N. Metaxas. Conditional models for contextual human motion recognition. *Comput. Vis. Image Underst.*, 104 (2-3):210–220, 2006. doi: 10.1016/J.CVIU.2006.07.014. URL https://doi.org/10.1016/j.cviu.2006.07.014.

[108] Kun Su, Xiulong Liu, and Eli Shlizerman. PREDICT & CLUSTER: unsupervised skeleton based action recognition. *CoRR*, abs/1911.12409, 2019. URL http://arxiv.org/abs/1911.12409.

[109] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. *CoRR*, abs/1510.00562, 2015. URL http://arxiv.org/abs/1510.00562.

[110] Yan Sun, Yixin Shen, and Liyan Ma. Msst-rt: Multi-stream spatial-temporal relative transformer for skeleton-based action recognition. *Sensors*, 21(16), 2021. ISSN 1424-8220. doi: 10.3390/s21165339. URL https://www.mdpi.com/1424-8220/21/16/5339.

[111] Zehua Sun, Jun Liu, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, and Gang Wang. Human action recognition from various data modalities: A review, 2021.

[112] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. *CoRR*, abs/1503.03578, 2015. URL http://arxiv.org/abs/1503.03578.

[113] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018. doi: 10.1109/CVPR.2018.00558.

[114] Nusrat Tasnim, Md. Mahbubul Islam, and Joong-Hwan Baek. Deep learning-based action recognition using 3d skeleton joints information. *Inventions*, 5(3), 2020. ISSN 2411-5134. doi: 10.3390/inventions5030049. URL https://www.mdpi.com/2411-5134/5/3/49.

[115] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022.

[116] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3d: Generic features for video analysis. *CoRR*, abs/1412.0767, 2014. URL http://arxiv.org/abs/1412.0767.

[117] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017. URL http://arxiv.org/abs/1711.11248.

[118] Neel Trivedi and Ravi Kiran Sarvadevabhatla. Psumnet: Unified modality part streams are all you need for efficient pose-based action recognition, 2022.

[119] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3889–3895. IJCAI/AAAI Press, 2016. URL http://www.ijcai.org/Abstract/16/547.

[120] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.

[121] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.,

2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[122] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

[123] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In Andrea Cavallaro, Simon Prince, and Daniel C. Alexander, editors, *British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings*, pages 1–11. British Machine Vision Association, 2009. doi: 10.5244/C.23.124. URL https://doi.org/10.5244/C.23.124.

[124] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, May 2013. ISSN 1573-1405. doi: 10.1007/s11263-012-0594-8. URL https://doi.org/10.1007/s11263-012-0594-8.

[125] Jiang Wang and Xiaohan Nie. Northwestern-ucla multiview action 3d dataset, 2023. URL https://wangjiangb.github.io/my_data.html. Accessed: 2024-10-10.

[126] Lei Wang and Piotr Koniusz. 3mformer: Multi-order multi-mode transformer for skeletal action recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5620–5631, 2023. doi: 10.1109/CVPR52729.2023.00544.

[127] Liang Wang and David Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *2007 IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383298. URL https://doi.org/10.1109/CVPR.2007.383298.

[128] Shengqin Wang, Yongji Zhang, Minghao Zhao, Hong Qi, Kai Wang, Fenglin Wei, and Yu Jiang. Skeleton-based action recognition via temporal-channel aggregation, 2022.

[129] Yang Wang and Greg Mori. Learning a discriminative hidden part model for human action recognition. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1721–1728. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/eddea82ad2755b24c4e168c5fc2ebd40-Abstract.html.

[130] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33 (7):1310–1323, 2011. doi: 10.1109/TPAMI.2010.214. URL https://doi.org/10.1109/TPAMI.2010.214.

[131] Chen Wei, Haoqi Fan, Saining Xie, Chaoxia Wu, Alan Loddon Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14648–14658, 2021. URL https://api.semanticscholar.org/CorpusID:245218767.

[132] Yu-Hui Wen, Lin Gao, Hongbo Fu, Fang-Lue Zhang, and Shihong Xia. Graph cnns with motif and variable temporal block for skeleton-based action recognition. *Pro-*

ceedings of the AAAI Conference on Artificial Intelligence, 33(01):8989–8996, Jul. 2019. doi: 10.1609/aaai.v33i01.33018989. URL https://ojs.aaai.org/index.php/AAAI/article/view/4929.

[133] Yuhang Wen, Zixuan Tang, Yunsheng Pang, Beichen Ding, and Mengyuan Liu. Interactive spatiotemporal token attention network for skeleton-based general interactive action recognition. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7886–7892, 2023. doi: 10.1109/IROS55552.2023.10342472.

[134] Zhanghao Wu, Paras Jain, Matthew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13266–13279, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6e67691b60ed3e4a55935261314dd534-Abstract.html.

[135] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Trans. Artif. Intell.*, 2(2):109–127, 2021. doi: 10.1109/TAI.2021.3076021. URL https://doi.org/10.1109/TAI.2021.3076021.

[136] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. Generative action description prompts for skeleton-based action recognition, 2023.

[137] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017. URL http://arxiv.org/abs/1712.04851.

[138] Wentian Xin, Ruyi Liu, Yi Liu, Yu Chen, Wenxin Yu, and Qiguang Miao. Transformer for skeleton-based action recognition: A review of recent advances. *Neurocomputing*, 537:164–186, 2023. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.03.001. URL https://www.sciencedirect.com/science/article/pii/S0925231223002217.

[139] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *CoRR*, abs/1801.07455, 2018. URL http://arxiv.org/abs/1801.07455.

[140] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2111–2117. AAAI Press, 2015. URL http://ijcai.org/Abstract/15/299.

[141] Hongye Yang, Yuzhang Gu, Jianchao Zhu, Keli Hu, and Xiaolin Zhang. Pgcn-tca: Pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition. *IEEE Access*, 8:10040–10047, 2020. doi: 10.1109/ACCESS.2020.2964115.

[142] Shuang Yang, Chunfeng Yuan, Baoxin Wu, Weiming Hu, and Fangshi Wang. Multi-feature max-margin hierarchical bayesian model for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1610–1618. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298769. URL https://doi.org/10.1109/CVPR.2015.7298769.

[143] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-

supervised learning with graph embeddings. *CoRR*, abs/1603.08861, 2016. URL http://arxiv.org/abs/1603.08861.

[144] Linfei Yin and Jiaxing Xie. Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. *Applied Energy*, 283:116328, 2021. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2020.116328. URL https://www.sciencedirect.com/science/article/pii/S0306261920317128.

[145] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28877–28888, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html.

[146] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. *CoRR*, abs/1604.07669, 2016. URL http://arxiv.org/abs/1604.07669.

[147] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 339–349. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/139.pdf.

[148] Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only

attention is needed for learning graph representations. *CoRR*, abs/2001.05140, 2020. URL https://arxiv.org/abs/2001.05140.

[149] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. *CoRR*, abs/1904.01189, 2019. URL http://arxiv.org/abs/1904.01189.

[150] Xikun Zhang, Chang Xu, Xinmei Tian, and Dacheng Tao. Graph edge convolutional neural networks for skeleton based action recognition. *CoRR*, abs/1805.06184, 2018. URL http://arxiv.org/abs/1805.06184.

[151] Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-graph transformer for node classification. *CoRR*, abs/2110.13094, 2021. URL https://arxiv.org/abs/2110.13094.

[152] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Comput. Surv.*, 56(1), August 2023. ISSN 0360-0300. doi: 10.1145/3603618. URL https://doi-org.uproxy.library.dc-uoit.ca/10.1145/3603618.

[153] Yuxuan Zhou, Zhi-Qi Cheng, Chao Li, Yanwen Fang, Yifeng Geng, Xuansong Xie, and Margret Keuper. Hypergraph transformer for skeleton-based action recognition, 2023.

[154] G. Zhu, L. Zhang, P. Shen, and J. Song. Multimodal gesture recognition using 3-d convolution and convolutional lstm. *IEEE Access*, 5:4517–4524, 2017. doi: 10.1109/ACCESS.2017.2684186.

[155] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. Deep graph structure learning for robust representations: A survey. *CoRR*, abs/2103.03036, 2021. URL https://arxiv.org/abs/2103.03036.

[156] Yi Zhu, Zhen-Zhong Lan, Shawn D. Newsam, and Alexander G. Hauptmann. Hidden two-stream convolutional networks for action recognition. *CoRR*, abs/1704.00389, 2017. URL http://arxiv.org/abs/1704.00389.

[157] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2857–2866. ACM, 2018. doi: 10.1145/3219819.3220054. URL https://doi.org/10.1145/3219819.3220054.

Appendix A: Classification results

A.1 Classification performance

NTU-60 Cross Subject: In the NTU-60 Cross Subject split, the dataset is divided by assigning specific subjects to the training and testing sets, ensuring no overlap between the subjects in each. This split tests the model's ability to generalize to unseen individuals performing various actions. The confusion matrix presented illustrates the classification performance across different action classes, highlighting both correctly classified actions and common misclassifications as seen in Figure A.1.

NTU-60 Cross View: The NTU-60 Cross View split focuses on evaluating the model's ability to recognize actions from different viewpoints. In this setting, the training and testing sets consist of the same subjects, but the camera views are kept distinct. The confusion matrix provides insights into how well the model generalizes across varying perspectives and identifies potential challenges in view-invariant action recognition as seen in Figure A.2.

NTU-120 Cross Subject: For the NTU-120 Cross Subject split, the larger NTU-120 dataset is divided similarly to NTU-60, with subjects in the training set excluded from the testing set. The increased number of action classes in NTU120 further tests the model's capacity to differentiate between a wider variety of actions. The confusion matrix displays the model's performance, helping to analyze its strengths and weaknesses in cross-subject generalization for this more complex dataset as seen in Figure A.3.

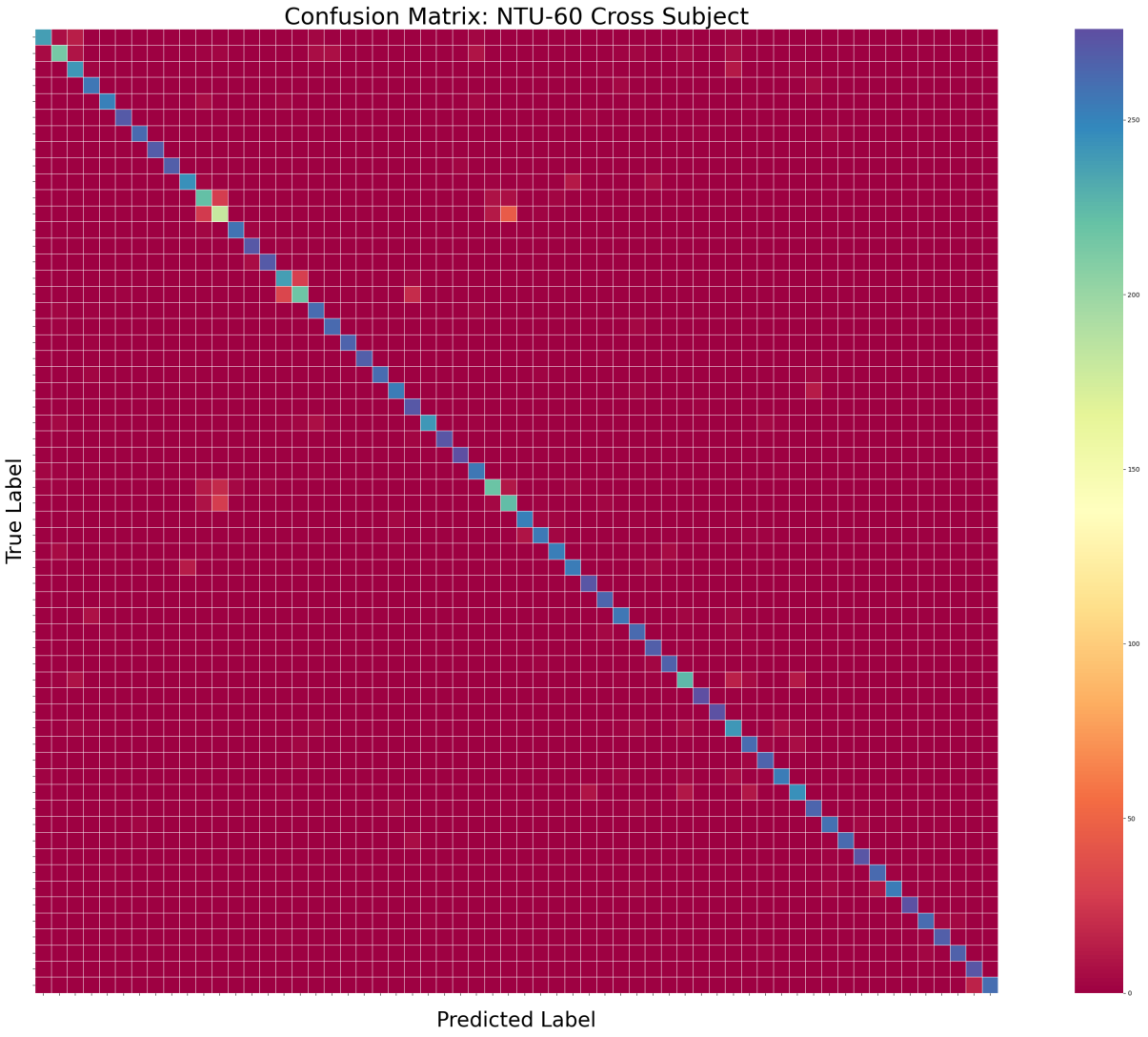

Figure A.1: Plot of the truth vs prediction results for NTU-60 Cross Subject split.

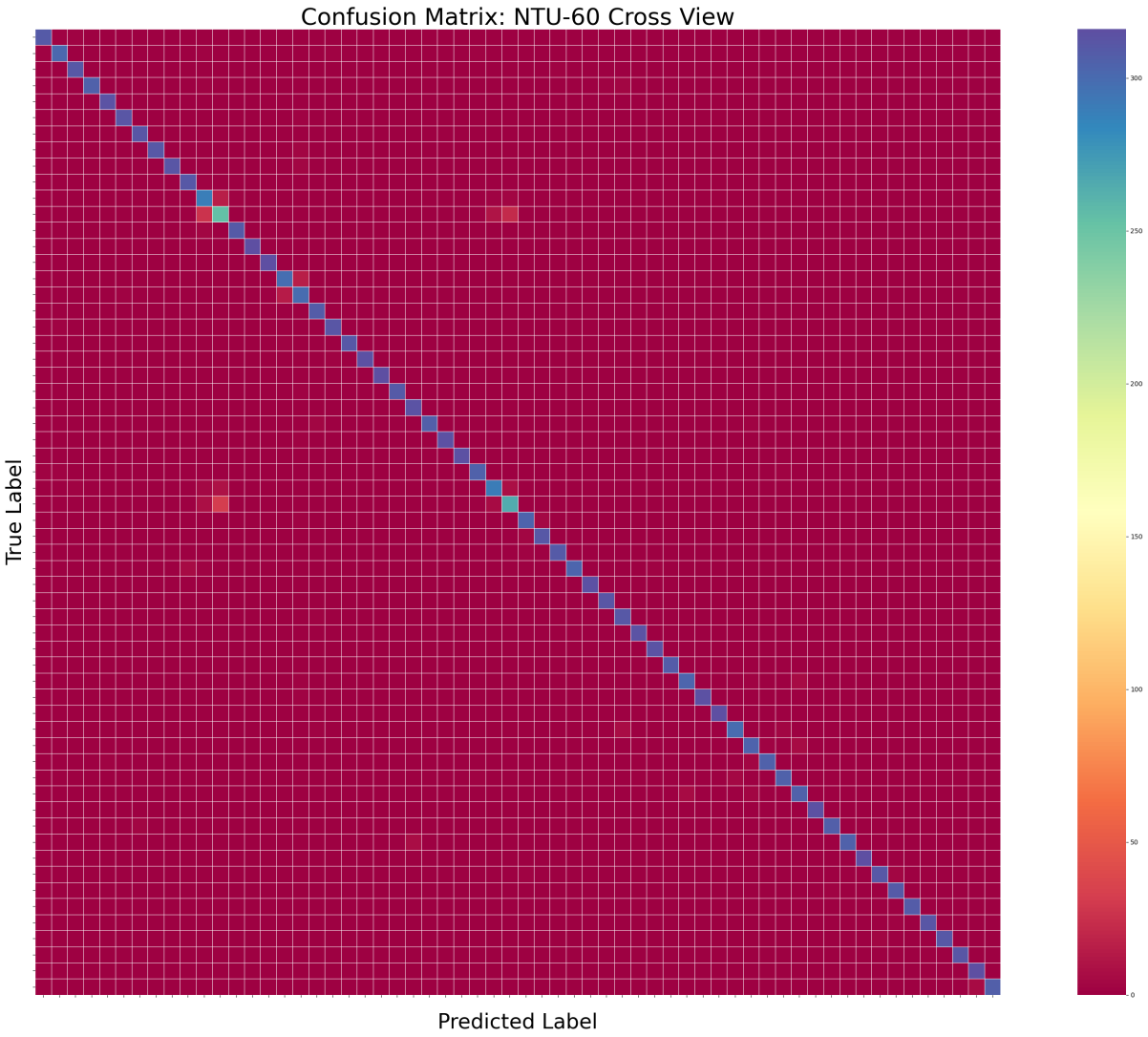

Figure A.2: Plot of the truth vs prediction results for NTU-60 Cross View split.

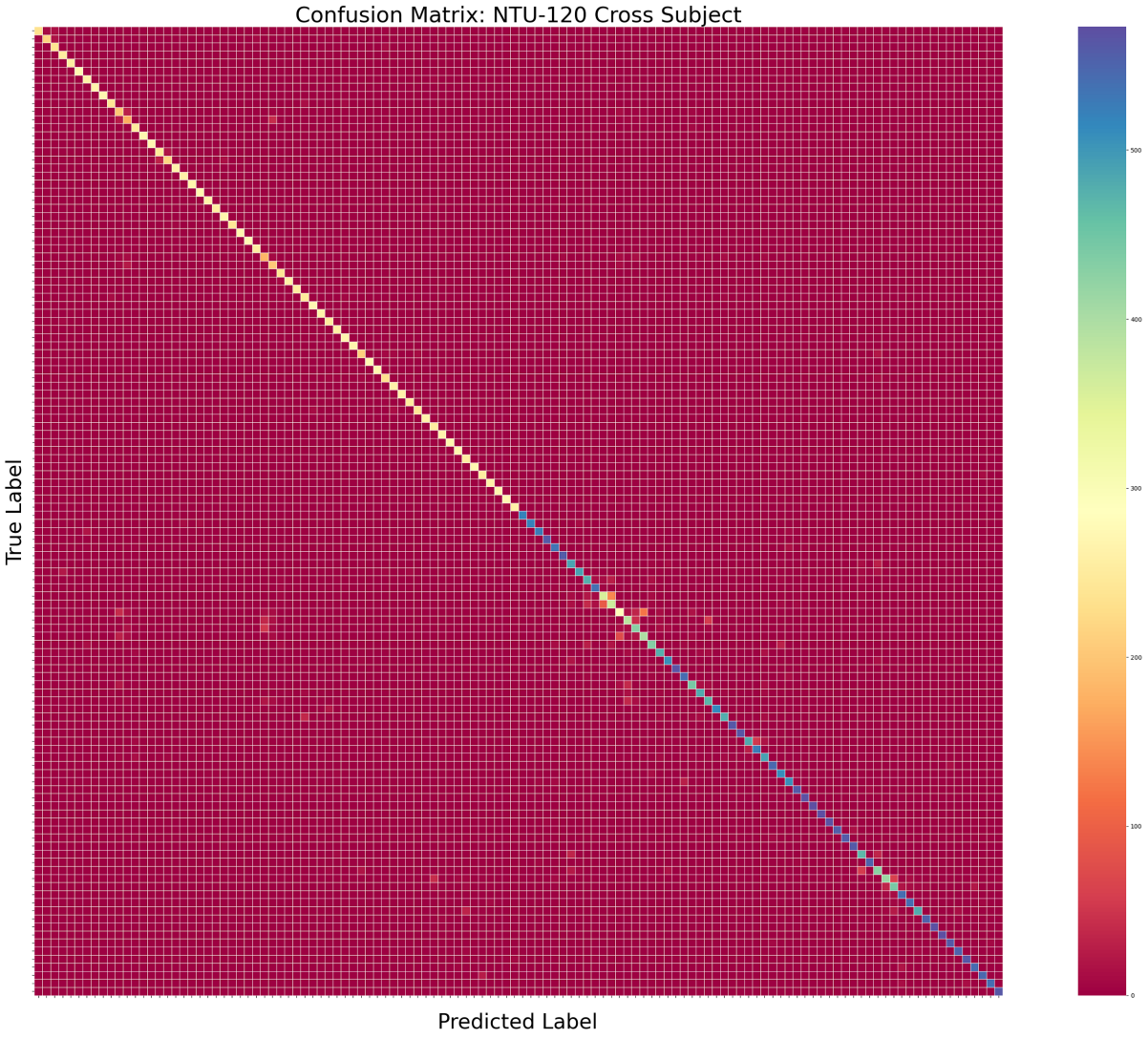

Figure A.3: Plot of the truth vs prediction results for NTU-120 Cross Subject split.

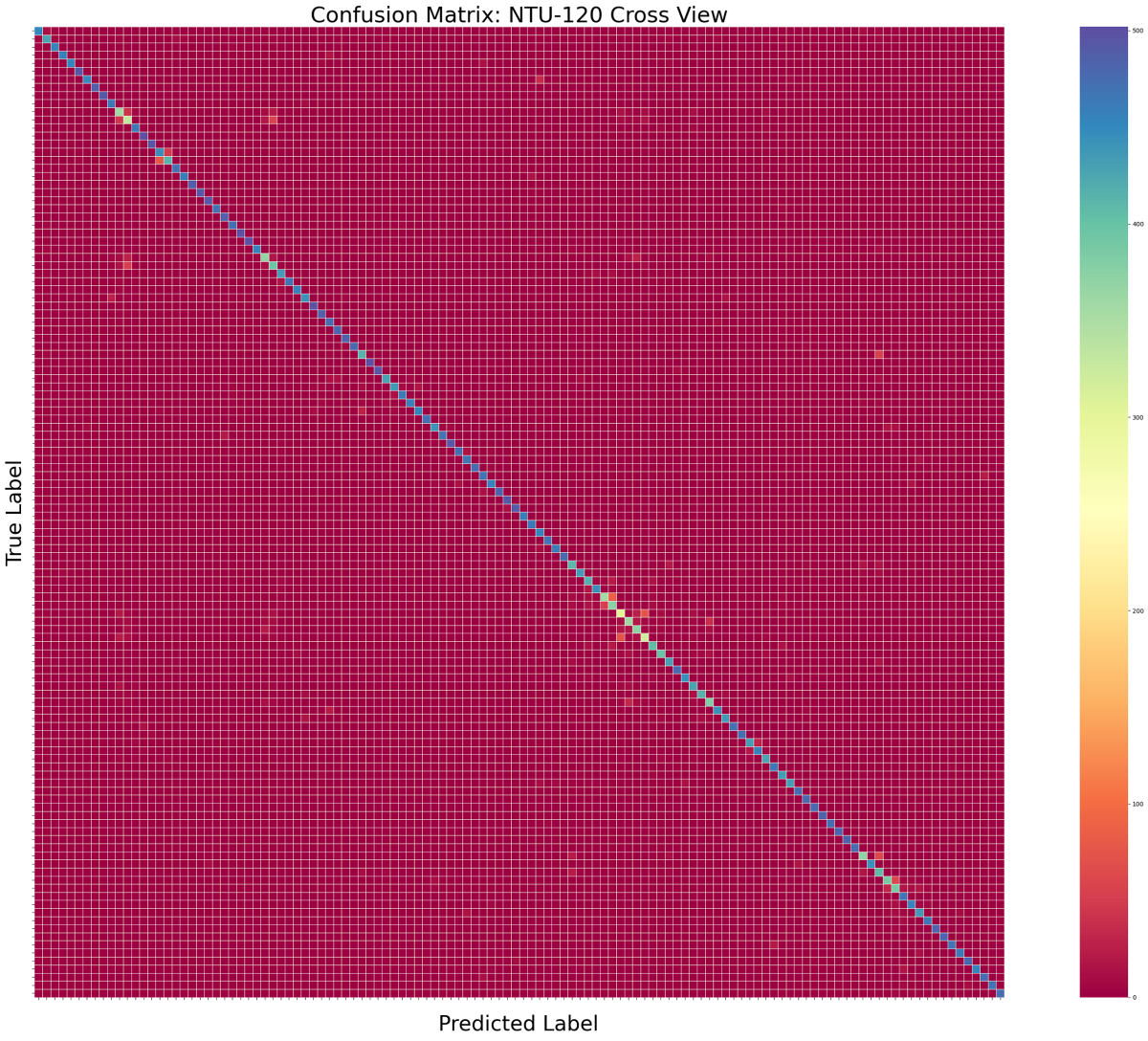

Figure A.4: Plot of the truth vs prediction results for NTU-120 Cross View split.

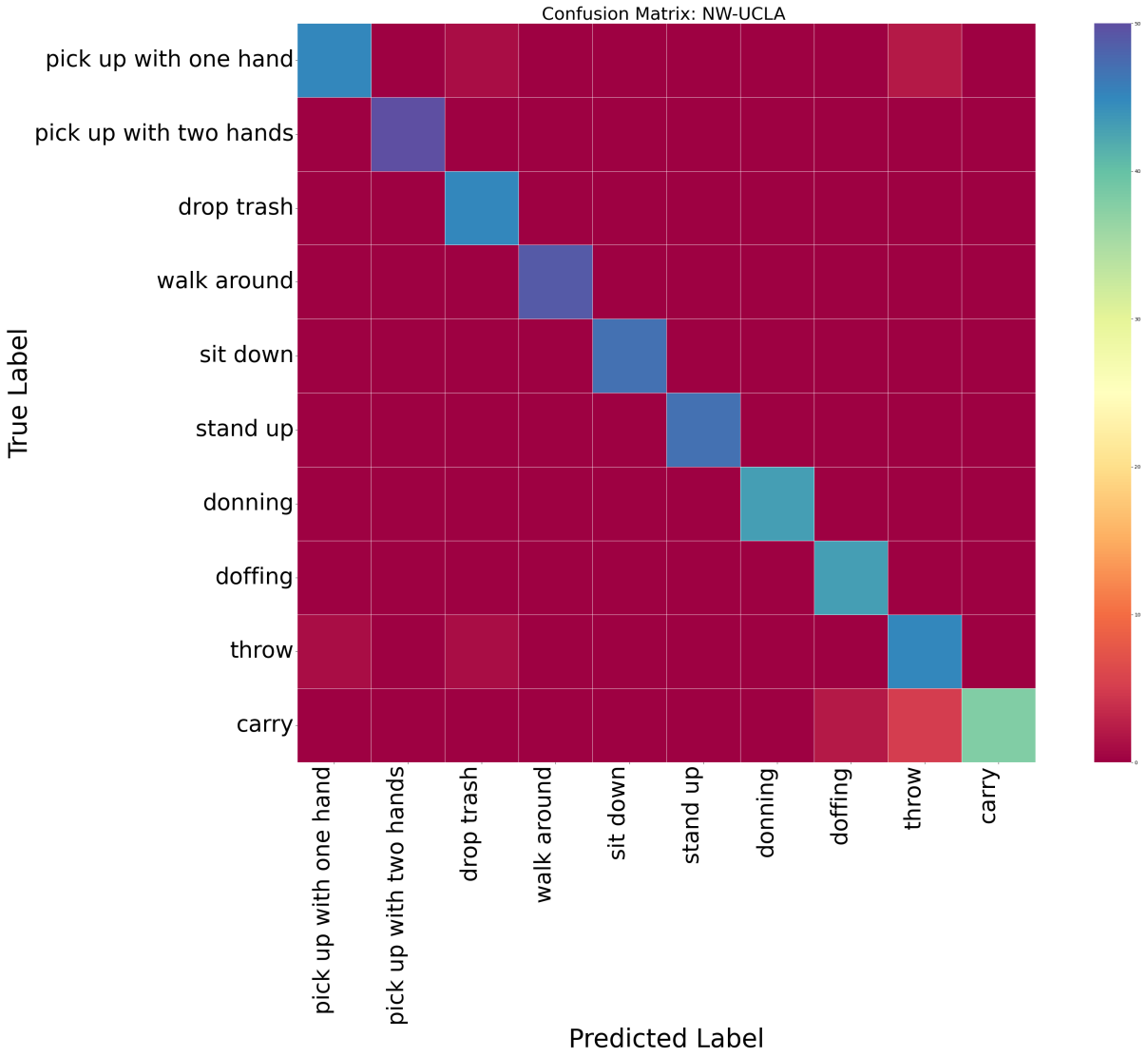

Figure A.5: Plot of the truth vs prediction results for NW-UCLA.

NTU-120 Cross View: In the NTU-120 Cross View split, actions performed by the same subjects are recorded from different camera angles, and the model is trained and tested on distinct views. This split challenges the model's robustness to varying viewpoints, especially with a broader set of action classes. The confusion matrix highlights the classification accuracy and areas where the model struggles with view-based action recognition as seen in Figure A.4.

NW-UCLA: The NW-UCLA dataset is designed to evaluate multi-view action recognition, where subjects perform actions recorded from multiple camera angles. The split follows a similar cross-view setup as NTU but involves a different dataset with its own set of challenges. The confusion matrix here showcases the model's ability to handle both viewpoint variation and inter-class similarities specific to the NW-UCLA dataset as seen in Figure A.5.

A.2 Classification analysis

A.2.1 NTU-60 Top 20 misclassification analysis

Fig. A.6 and Table A.1 highlight key misclassifications between different actions. Notably, the action "writing" is frequently misclassified as "typing on a keyboard," with 45 occurrences, indicating a significant overlap in these activities. Similarly, the actions "take off a shoe" and "wear a shoe" exhibit a close relationship, with 33 and 29 misclassifications, respectively. Other notable pairs include "reading" misclassified as "writing" (28 times) and "typing on a keyboard" misclassified as "writing" (27 times). This suggests that certain actions may share similar motion patterns or contextual features, leading to confusion in classification models. Overall, the data underscores the challenges in distinguishing between closely related activities within the NTU-60 dataset for the cross subject split.

Fig. A.7 and Table A.2 present insights into misclassifications among various actions.

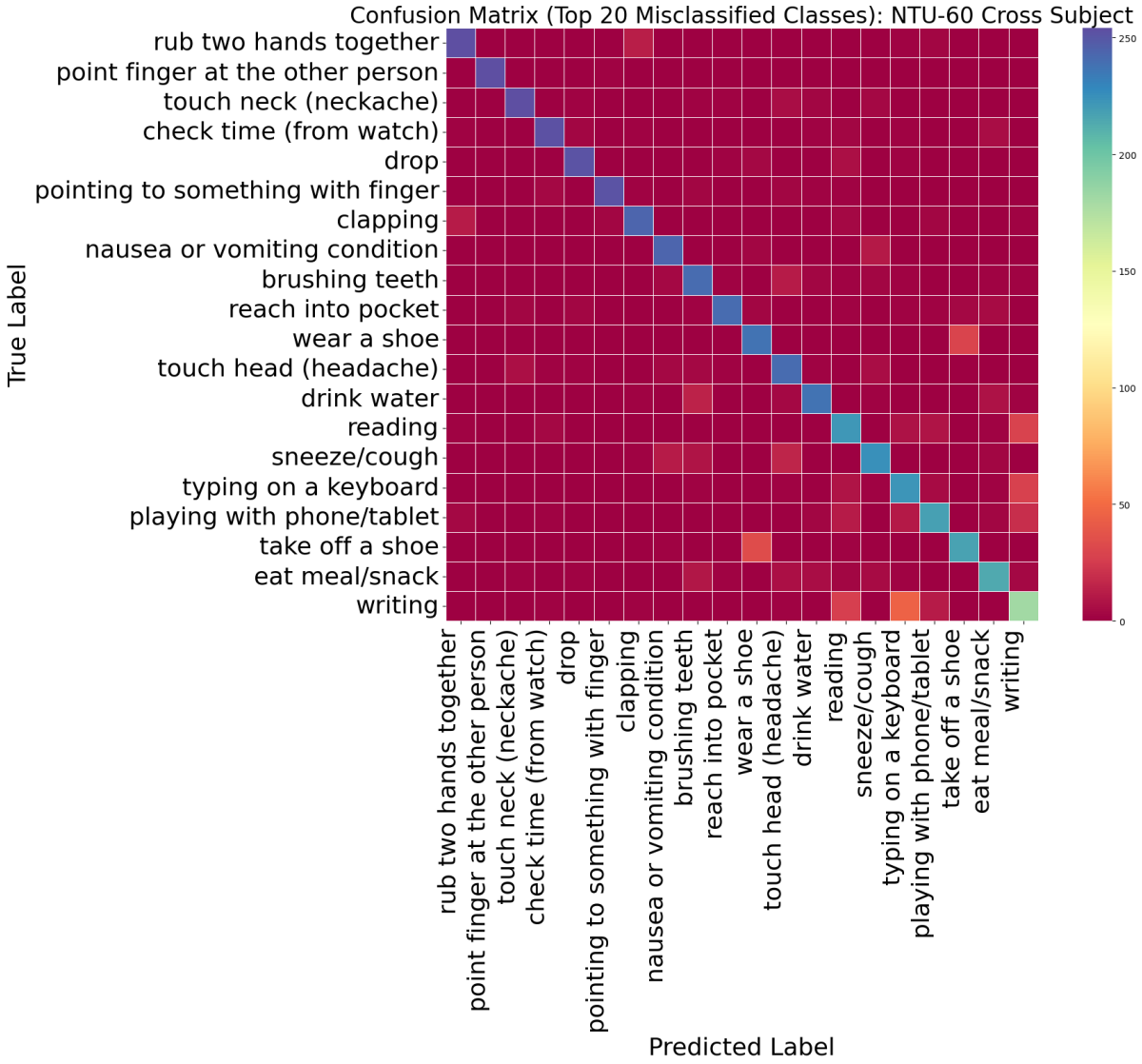

Figure A.6: Plot of the top 20 truth vs prediction results for NTU-60 Cross Subject split.

True	**Prediction**	**Count**
writing	typing on a keyboard	45
take off a shoe	wear a shoe	33
wear a shoe	take off a shoe	29
reading	writing	28
typing on a keyboard	writing	27
writing	reading	26
take off a shoe	kicking something	20
playing with phone/tablet	writing	19
walking apart from each other	walking towards each other	16
sneeze/cough	touch head (headache)	15
drink water	brushing teeth	14
hand waving	use a fan (with hand or paper)	12
rub two hands together	clapping	12
brushing teeth	touch head (headache)	11
clapping	rub two hands together	11
writing	playing with phone/tablet	11
playing with phone/tablet	reading	11
sneeze/cough	nausea or vomiting condition	11
playing with phone/tablet	typing on a keyboard	10
nausea or vomiting condition	sneeze/cough	10

Table A.1: NTU-60 Cross Subject

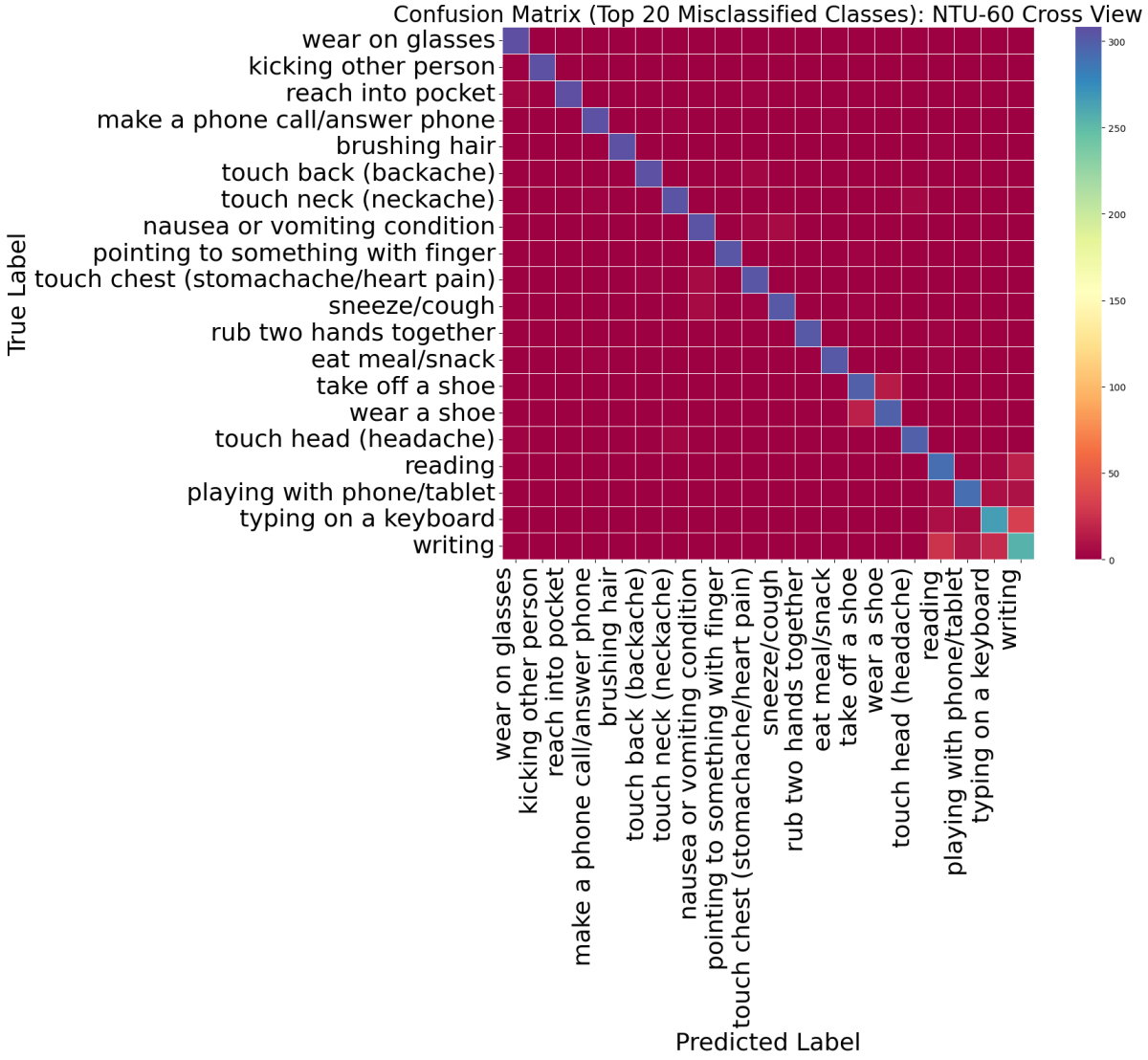

Figure A.7: Plot of the top 20 truth vs prediction results for NTU-60 Cross View split.

True	**Prediction**	**Count**
typing on a keyboard	writing	32
writing	reading	26
writing	typing on a keyboard	21
reading	writing	16
wear a shoe	take off a shoe	16
take off a shoe	wear a shoe	14
writing	playing with phone/tablet	10
playing with phone/tablet	writing	9
playing with phone/tablet	typing on a keyboard	8
typing on a keyboard	reading	8
touch head (headache)	wipe face	7
kicking other person	kicking something	7
typing on a keyboard	playing with phone/tablet	6
rub two hands together	clapping	6
touch chest (stomachache/heart pain)	nausea or vomiting condition	6
nausea or vomiting condition	sneeze/cough	6
walking apart from each other	walking towards each other	6
sneeze/cough	nausea or vomiting condition	5
playing with phone/tablet	reading	4
pointing to something with finger	taking a selfie	4

Table A.2: NTU-60 Cross View

A notable trend is the confusion between "typing on a keyboard" and "writing," with 32 instances of misclassification, highlighting the similarities in hand movements associated with both tasks. The action "writing" is also misclassified as "reading" (26 times), indicating potential overlap in their execution contexts. Additionally, the pair "wear a shoe" and "take off a shoe" shows significant misclassification (16 and 14 times, respectively), further emphasizing the close relationship between these actions. Other notable instances include "playing with phone/tablet" being misclassified as "writing" (10 times) and vice versa (9 times), suggesting that users might engage in similar gestures during these activities. Overall, this data reflects the challenges faced in accurately differentiating between closely related actions within the NTU-60 dataset, particularly in the context of cross-view evaluations.

A.2.2 NTU-120 Top 20 misclassification analysis

Fig. A.8 and Table A.3 present a summary of misclassifications for the NTU-120 Cross Subject dataset, highlighting the most frequently confused actions. The action "make ok sign" is commonly misclassified as "make victory sign," occurring 140 times, which indicates a significant overlap between these gestures. Another notable pair is "staple book," which is frequently misidentified as "cutting paper (using scissors)" with 121 instances. Additionally, the action "make victory sign" is misclassified as "make ok sign" 92 times, further illustrating the challenges in differentiating between similar hand gestures. Other misclassifications include "hit other person with something" mistaken for "wield knife towards other person" (61 counts) and "blow nose" confused with "yawn" (57 counts). These results emphasize the need for improved classification methods, as many actions have closely related predictions, potentially affecting the accuracy of gesture recognition systems.

Fig. A.9 and Table A.4 summarize the NTU-120 Cross View results, highlighting key misclassifications between true and predicted actions. The most significant misclassifi-

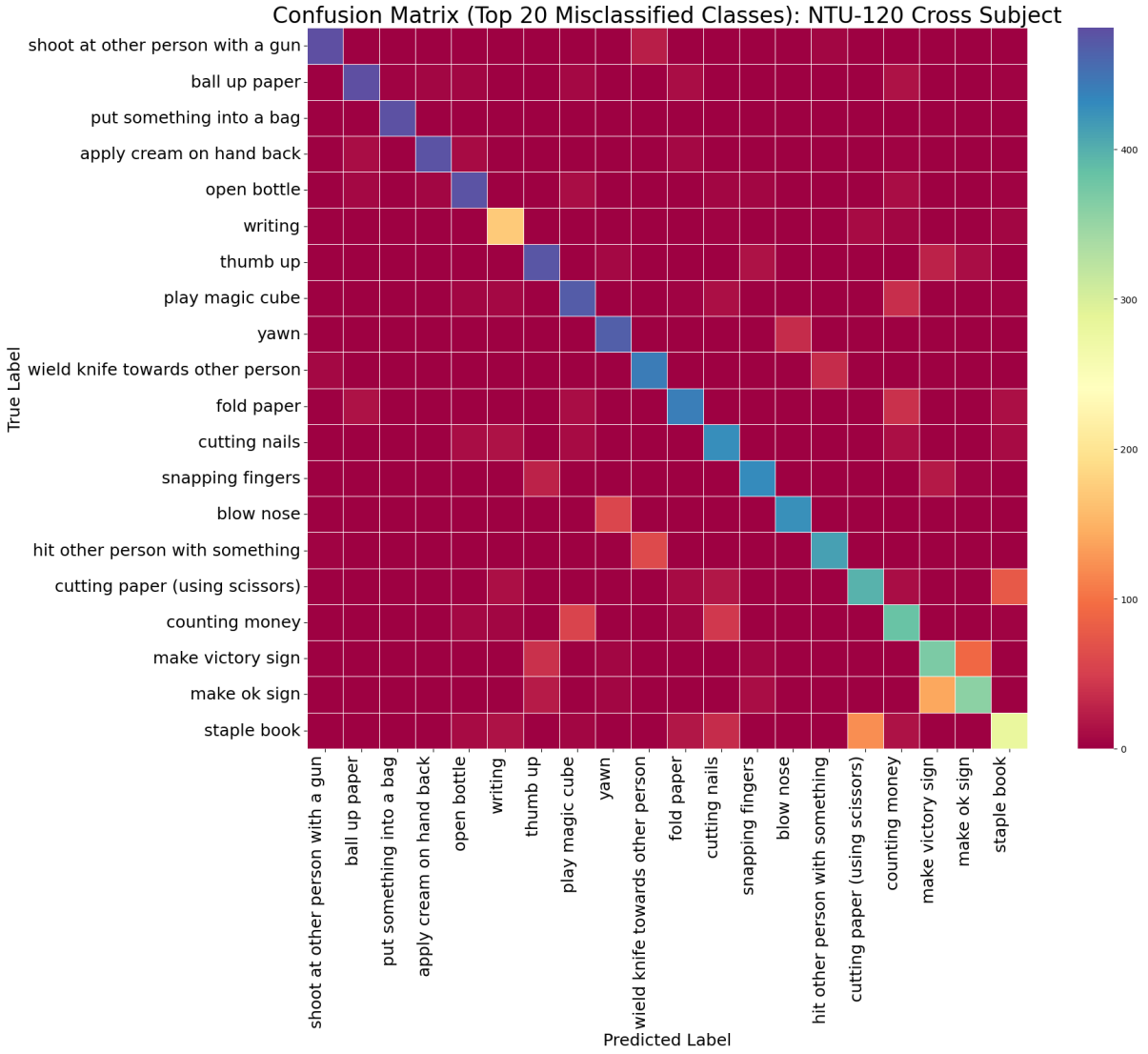

Figure A.8: Plot of the top 20 truth vs prediction results for NTU-120 Cross Subject split.

True	**Prediction**	**Count**
make ok sign	make victory sign	140
staple book	cutting paper (using scissors)	121
make victory sign	make ok sign	92
cutting paper (using scissors)	staple book	76
hit other person with something	wield knife towards other person	61
blow nose	yawn	57
counting money	play magic cube	56
cutting nails	playing with phone/tablet	53
put something into a bag	take something out of a bag	52
counting money	cutting nails	44
writing	typing on a keyboard	42
staple book	reading	41
hit other person with something	punching/slapping other person	41
make victory sign	thumb up	39
fold paper	counting money	39
counting money	playing with phone/tablet	38
apply cream on hand back	rub two hands together	38
play magic cube	counting money	36
staple book	cutting nails	35
yawn	hush (quite)	35

Table A.3: NTU-120 Cross Subject

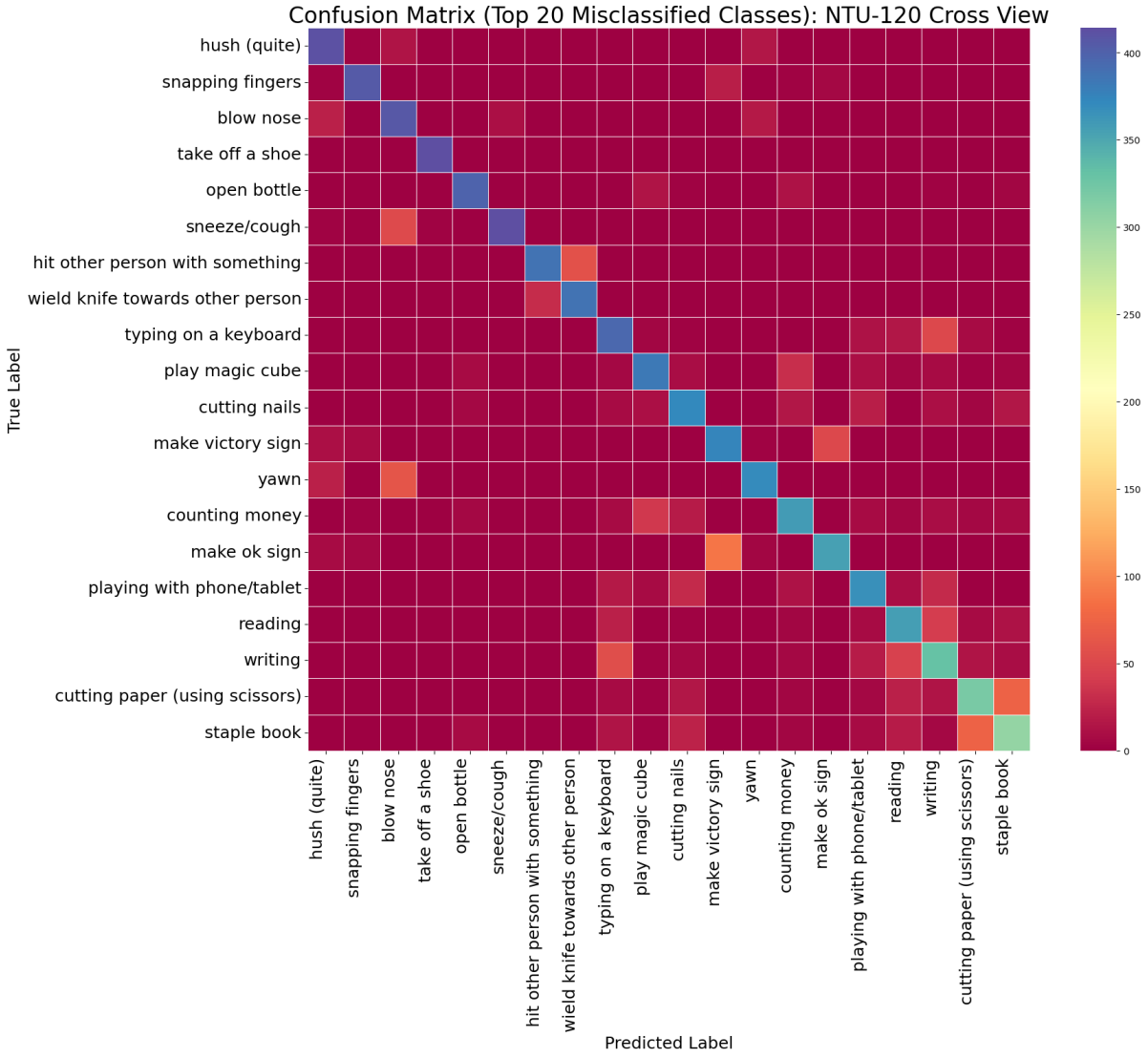

Figure A.9: Plot of the top 20 truth vs prediction results for NTU-120 Cross View split.

True	**Prediction**	**Count**
make ok sign	make victory sign	88
take off a shoe	wear a shoe	78
cutting paper (using scissors)	staple book	74
staple book	cutting paper (using scissors)	73
yawn	blow nose	62
hit other person with something	wield knife towards other person	59
writing	typing on a keyboard	55
sneeze/cough	blow nose	53
typing on a keyboard	writing	51
make victory sign	make ok sign	51
wear a shoe	take off a shoe	49
writing	reading	45
reading	writing	42
counting money	play magic cube	38
throw	shoot at the basket	35
play magic cube	counting money	32
put something into a bag	take something out of a bag	30
wield knife towards other person	hit other person with something	30
playing with phone/tablet	writing	29
nausea or vomiting condition	sneeze/cough	29

Table A.4: NTU-120 Cross View

True	Prediction	Count
carry	throw	5
pick up with one hand	throw	2
carry	doffing	2
pick up with one hand	drop trash	1
throw	pick up with one hand	1

Table A.5: NW-UCLA

cation involves the gesture "make ok sign," which was frequently predicted as "make victory sign" with a count of 88 instances. Another notable confusion occurs between "take off a shoe" and "wear a shoe," with 78 occurrences. Additionally, actions involving cutting paper using scissors and stapling a book were misclassified 74 and 73 times, respectively. Other common mispredictions include "yawn" being predicted as "blow nose" (62 counts) and "hit other person with something" being confused with "wield knife towards other person" (59 counts). The data indicates a pattern of related actions often leading to misclassification, such as writing and typing on a keyboard, with counts of 55 and 51, respectively. These insights could help improve the model's accuracy by refining the recognition of similar gestures and actions.

A.2.3 NW-UCLA Top 5 misclassification analysis

The NW-UCLA dataset presents a limited number of actions with distinct prediction outcomes as shown in Fig. A.10 and Table A.5. The most frequent true action, "carry," is incorrectly predicted as "throw" in 5 instances, indicating a notable confusion between these two actions. Additionally, the action "pick up with one hand" is misclassified as "throw" twice and as "drop trash" once, highlighting potential overlap in motion characteristics. The action "carry" is also confused with "doffing," with 2 occurrences of misclassification. The predictions demonstrate a low overall count, suggesting that the model may require further training or refinement to improve accuracy in distinguishing these closely related actions.

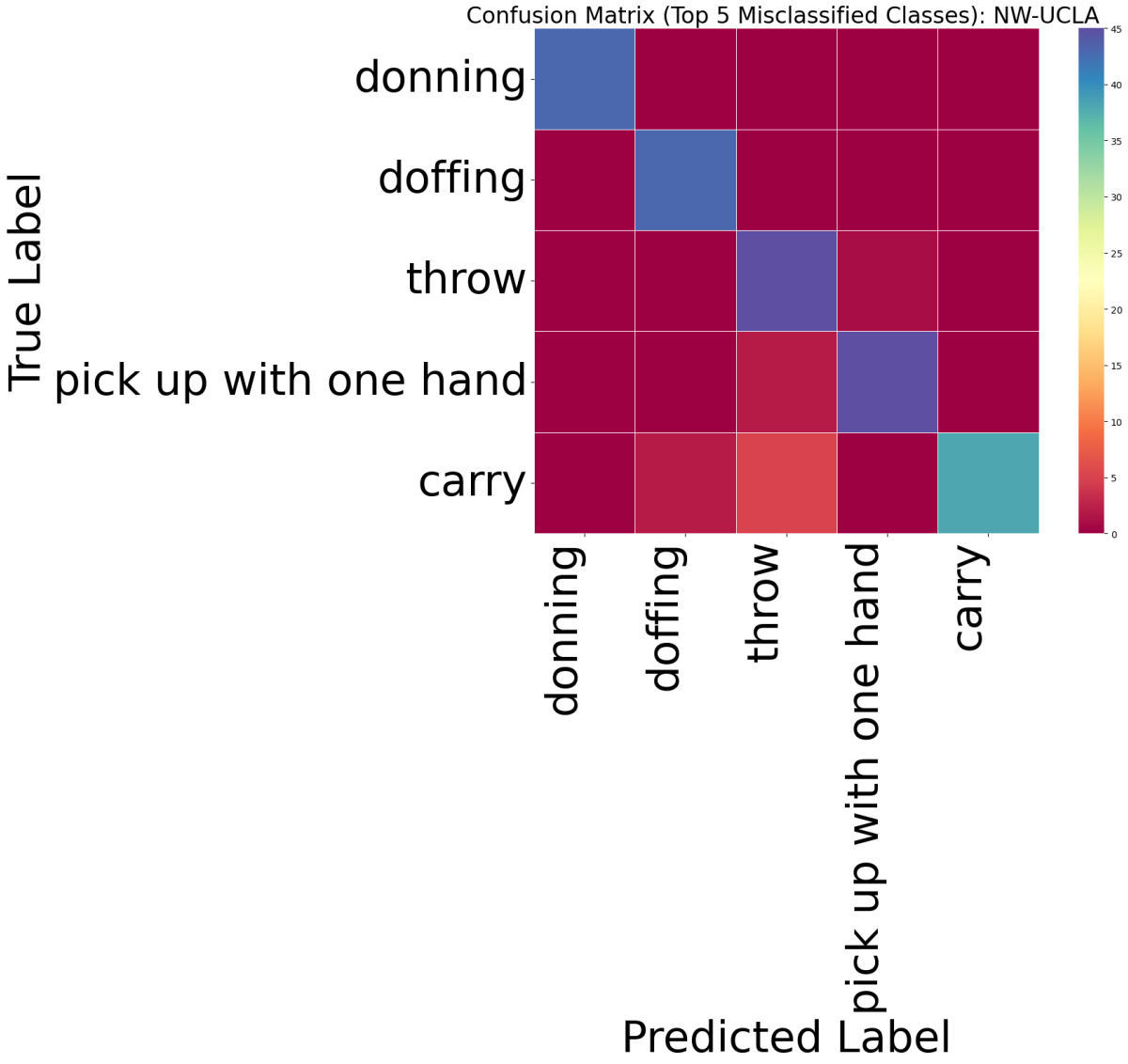

Figure A.10: Plot of the top 20 truth vs prediction results for NTU-120 Cross View split.