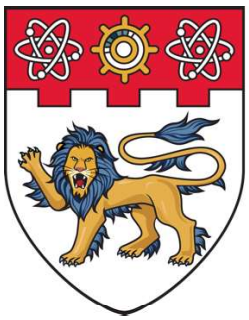


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

LEARNING WITH FEW LABELS FOR SKELETON-BASED ACTION RECOGNITION

YANG SIYUAN

INTERDISCIPLINARY GRADUATE PROGRAMME
Rapid-Rich Object Search Lab

2023

LEARNING WITH FEW LABELS FOR SKELETON-BASED ACTION RECOGNITION

YANG SIYUAN

INTERDISCIPLINARY GRADUATE PROGRAMME
Rapid-Rich Object Search Lab

A thesis submitted to the Nanyang Technological University in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

23/08/2023

Date

SIYUAN YANG

Yang Siyuan

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23/08/2023

.....

Date

.....

Prof. Kot Chichung, Alex

Authorship Attribution Statement

This thesis contains material from 2 papers accepted at conferences in which I am listed as an author, and also from two works that are under review.

Chapter 3 is published as:

Siyuan Yang, Jun Liu, Shijian Lu, Er Meng Hua and Alex C. Kot, "Collaborative Learning of Gesture Recognition and 3D Hand Pose Estimation with Multi-Order Feature Analysis," in 2020 European Conference on Computer Vision (ECCV), pp. 769–786, doi: 10.1007/978-3-030-58580-8_45.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Assist/Prof. Jun Liu suggested the topic.
- I designed the initial method, and the method was improved by discussions with Assist/Prof. Jun Liu, Prof. Shijian Lu, and Prof. Alex C. Kot.
- I implemented the proposed method.
- I designed and conducted experiments with the suggestions provided by Assist/Prof. Jun Liu.
- I wrote the initial manuscript. Assist/Prof. Jun Liu, Prof. Shijian Lu, Prof. Er Meng Hua, and Prof. Alex C. Kot revised the manuscript.

Chapter 4 is published as:

Siyuan Yang, Jun Liu, Shijian Lu, Er Meng Hua and Alex C. Kot, "Skeleton Cloud Colorization for Unsupervised 3D Action Representation Learning," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), doi: 10.1109/ICCV48922.2021.01317.

And:

Siyuan Yang, Jun Liu, Shijian Lu, Er Meng Hua, Yongjian Hu and Alex C. Kot, "Self-Supervised 3D Action Representation Learning with Skeleton Cloud Colorization," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, doi: 10.1109/TPAMI.2023.3325463.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Assist/Prof. Jun Liu suggested the topic.
- I designed the initial method, and the method was improved by discussions with Assist/Prof. Jun Liu, Prof. Shijian Lu, and Prof. Alex C. Kot.
- I implemented the proposed method.

- I designed and conducted experiments with the suggestions provided by Assist/Prof. Jun Liu.
- I wrote the initial manuscript. Assist/Prof. Jun Liu, Prof. Shijian Lu, Prof. Er Meng Hua, Prof. Yongjian Hu, and Prof. Alex C. Kot revised the manuscript.

Chapter 5 is published as:

Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C Kot, "One-Shot Action Recognition via Multi-Scale Spatial-Temporal Skeleton Matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, DOI 10.1109/TPAMI.2024.3363831.

The contributions of the co-authors are as follows:

- Prof. Alex C. Kot and Assist/Prof. Jun Liu suggested the topic.
- I designed the initial method, and the method was improved by discussions with Assist/Prof. Jun Liu, Prof. Shijian Lu, and Prof. Alex C. Kot.
- I implemented the proposed method.
- I designed and conducted experiments with the suggestions provided by Assist/Prof. Jun Liu.
- I wrote the initial manuscript. Assist/Prof. Jun Liu, Prof. Shijian Lu, Prof. Er Meng Hua, and Prof. Alex C. Kot revised the manuscript.

23/08/2022

.....

Date

.....

Yang Siyuan

Acknowledgements

First of all, I would like to express my greatest gratitude to my supervisor Prof. Alex C. Kot, who gave me the opportunity to pursue my Ph.D. degree in Nanyang Technological University, Singapore. His unwavering support and invaluable insights have been instrumental in my research journey. Prof. Kot's passion for both work and life has deeply influenced me. I am thankful that I can be a student under his supervision, and I am confident this experience will benefit the rest of my life.

I gratefully acknowledge my co-supervisor, Prof. Shijian Lu. His unwavering confidence in my research capabilities has been a tremendous source of encouragement. The emphasis he places on the practicality and demonstrability of ideas has profoundly shaped my approach to research. His meticulous nature and keen attention to detail serve as a constant reminder of the hallmarks of an exceptional researcher.

My sincere appreciation extends to Prof. Jun Liu from the Singapore University of Technology and Design, with who I closely worked during my Ph.D. journey. Especially during the early stages of my research, his patient guidance and unwavering support were invaluable. Under his mentorship, I transformed from a fledgling student to a meticulous researcher. His commitment to refining every aspect of our papers and his exacting approach to the scientific inquiry have not only influenced this thesis but also set a high standard I will always strive for. I would also like to express my appreciation to my TAC members, Prof. Saman S. Abeysekera and Prof. Kezhi Mao, for their diligent review of my research progress and invaluable guidance.

During my Ph.D. study, I enjoyed the time spent with my colleagues in the ROSE lab, including Renjie Wan, Haoliang Li, Zhi Li, Rizhao Cai, Chenyu Yi, Ling Li, Yufei Wang, Yi Yu, Shan Lin, Wenhan Yang, Hao Cheng, Lanqing Guo, Chong Wang, Zhan Lu, Peijun Bao, Qichen Zheng, Chenqi Kong, Yang Yu, Qing Zhang,

Ruohan Meng. I would also like to extend my heartfelt thanks to Dr. Dennis Sng, Ms. Qian Wang, and Ms. Ooy Mei Chai for their consistent administrative support and assistance during my academic journey.

Lastly, I owe a depth of gratitude to my family and friends. My heartfelt thanks go to my friends Rongqian Yu, Nan Yang, Haoyi Ma, Hang Zhang, Wenbin Pan, Hexin Liu, Meirong Li, Yi Wu, Ruihong Xie, and Jiaxu Wang for their unwavering encouragement and for providing much-needed relief during the most stressful times. I'm deeply grateful to my mother and relatives for their constant support and encouragement. Their continued encouragement for my overseas studies, even after the passing of my father, is something I deeply cherish. To my late father, it's my hope to make you proud with every achievement I secure.

Contents

List of Figures

List of Tables

Symbols

Symbols in Chapter 3

$\mathcal{J}$	the joint-aware feature maps
N	the number of hand joints
$\mathcal{J}_i$	the joint-aware feature maps of i_{th} joint
$\mathcal{H}$	the generated 2D heatmaps
$\mathcal{H}_i$	the generated 2D heatmap of i_{th} joint
D	the predicted depth values
D_i	the predicted depth values of i_{th} joint
$\mathcal{P}$	the pose-optimized joint-aware feature maps
$\mathcal{G}$	the gesture-optimized joint-aware feature maps
y	the predicted gesture category
Zo	the zero-order Features
Fo	the first-order features
So	the second-order features
m	the margin from the smaller to the larger hypersphere
L_2	the L_2 norm
FD	the feature difference
L_{2d}	the 2D Heatmaps loss
$\hat{\mathcal{H}}$	the ground-truth heatmaps
L_{3d}	the depth regression loss
$\hat{D}$	the ground-truth depth values
L_c	the classification loss

$\hat{y}$	the ground-truth gesture category
λ_{2d}	the weight of 2D Heatmaps loss
λ_{3d}	the weight of depth regression loss
λ_c	the weight of classification loss

Symbols in Chapter 4

$v_{t,j}$	the value of j^{th} skeleton joint in the t^{th} frame
$x_{t,j}$	the x value of j^{th} skeleton joint in the t^{th} frame
$y_{t,j}$	the y value of j^{th} skeleton joint in the t^{th} frame
$z_{t,j}$	the z value of j^{th} skeleton joint in the t^{th} frame
T	the number of frames
J	the number of joints
V_t	the set of joints in the t^{th} frame
P_r	the raw skeleton cloud
P_τ	the temporally colorized skeleton cloud
$r^\tau_{t,j}$	the r value of t^{th} frame's j^{th} skeleton joint in P_τ
$g^\tau_{t,j}$	the g value of t^{th} frame's j^{th} skeleton joint in P_τ
$b^\tau_{t,j}$	the b value of t^{th} frame's j^{th} skeleton joint in P_τ
P_s	the spatially colorized skeleton cloud
$r^s_{t,j}$	the r value of t^{th} frame's j^{th} skeleton joint in P_s
$g^s_{t,j}$	the g value of t^{th} frame's j^{th} skeleton joint in P_s
$b^s_{t,j}$	the b value of t^{th} frame's j^{th} skeleton joint in P_s
P_p	the Person-level colorized skeleton cloud
$r^p_{t,j,n}$	the r value of t^{th} frame's j^{th} skeleton joint in P_p
$g^p_{t,j,n}$	the g value of t^{th} frame's j^{th} skeleton joint in P_p
$b^p_{t,j,n}$	the b value of t^{th} frame's j^{th} skeleton joint in P_p
$E(.)$	the encoder
$D(.)$	the decoder
$E_\tau(.)$	the encoder for the temporally colorized skeleton cloud
$E_s(.)$	the encoder for the spatially colorized skeleton cloud

$E_p(.)$	the encoder for the Person-level colorized skeleton cloud
$D_\tau(.)$	the decoder for the temporally colorized skeleton cloud
$D_s(.)$	the decoder for the spatially colorized skeleton cloud
$D_p(.)$	the decoder for the Person-level colorized skeleton cloud
$\widehat{P_\tau}$	the obtained repainted temporally colorized skeleton cloud
$d_{CH}(.,.)$	the Chamfer distance
r_i	the raw image of the i-th data sample
$P_{\tau c}$	the temporally coarse-grained colorized skeleton cloud
$r_{t,j}^{\tau c}$	the r value of t^{th} frame's j^{th} skeleton joint in $P_{\tau c}$
$g_{t,j}^{\tau c}$	the g value of t^{th} frame's j^{th} skeleton joint in $P_{\tau c}$
$b_{t,j}^{\tau c}$	the b value of t^{th} frame's j^{th} skeleton joint in $P_{\tau c}$
P_{sc}	the spatially coarse-grained colorized skeleton cloud
$r_{t,j}^{cs}$	the r value of t^{th} frame's j^{th} skeleton joint in P_{sc}
$g_{t,j}^{cs}$	the g value of t^{th} frame's j^{th} skeleton joint in P_{sc}
$b_{t,j}^{cs}$	the b value of t^{th} frame's j^{th} skeleton joint in P_{sc}
L_{cls}	the classification loss

Symbols in Chapter 5

S	support set
Q	query set
D_{train}	meta-training set
D_{test}	meta-testing set
s_1	the body-joint scale
s_2	the part-level scale
s_3	the limb-level scale
$s(\cdot, \cdot)$	the semantic relevance score between two skeleton features
$\mathcal{X}$	suppliers
$\mathcal{Y}$	demanders
x_i	the i_{th} supplier
y_j	the j_{th} demander

$OT(\cdot, \cdot)$	the optimal transportation cost between two sets of representations
π	the optimal matching flow between two distributions
r_i	the weight for i_{th} node in suppliers
c_j	the weight for j_{th} node in demanders
d_{ij}	the pair-wise distance between i_{th} supplier and j_{th} demander
$D_{emd}(\cdot, \cdot)$	the Earth Mover's Distance between feature maps $\mathbf{X}$ and $\mathbf{Y}$
$s_{emd}(\cdot, \cdot)$	the semantic relevance score between feature maps $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{X}_{s1}$	the first-scale spatial feature map
$\mathbf{X}_{s2}$	the second-scale spatial feature map
$\mathbf{X}_{s3}$	the third-scale spatial feature map
N	the number of nodes for the first-scale spatial graph
N_2	the number of nodes for the second-scale spatial graph
N_3	the number of nodes for the third-scale spatial graph
$s_{ms}(\cdot, \cdot)$	the multi-spatial scale semantic relevance score
$\mathbf{X}_{t1}$	the first-scale temporal feature map
$\mathbf{X}_{t2}$	the second-scale temporal feature map
$\mathbf{X}_{t3}$	the third-scale temporal feature map
T	the number of frames for the first-scale temporal graph
T_2	the number of frames for the second-scale temporal graph
T_3	the number of frames for the third-scale temporal graph
$s_{mt}(\cdot, \cdot)$	the multi-temporal scale semantic relevance score
$s_{cs}(\cdot, \cdot)$	the cross-spatial scale semantic relevance score
$s_{ct}(\cdot, \cdot)$	the cross-temporal scale semantic relevance score

Abbreviations

Acc.	Accuracy
AGCN	Adaptive Graph Convolutional Network
AvgPool	Average pooling
CNNs	Convolutional Neural Networks
C-F	Coarse-fine
CS	Cross-subject
CV	Cross-view
C-scale	Cross-scale matching
EMD	Earth Mover's Distance
FC	fully-connected
FPHA	First-Person Hand Action
GCNs	Graph convolutional networks
MAE	Mask Auto-Encoder
MSE	Mean squared error
M-scale	Multi-scale matching
M&Cscale	Multi-scale and cross-scale matching
NW-UCLA	Northwestern-UCLA
PCK	Percentage of correct keypoints
PS	Person stream
RNNs	Recurrent Neural Networks
Smarthome	Toyota Smarthome

SS	Spatial stream
S-scale	Single-scale matching
TCN	Temporal convolution
TS	Temporal stream
UWA3D	Multiview Activity II

Abstract

Human Action Recognition, which involves discerning human actions, is vital for many real-world applications. Skeleton sequences, tracing human body joint trajectories, capture essential human motions, making them appropriate for action recognition. Compared to RGB videos or depth data, 3D skeleton data offers concise representations of human behaviors, proving robust against appearance variations, distractions, and viewpoint changes. This has led to increased interest in skeleton-based action recognition research.

With the advance of deep learning, deep neural networks (e.g., CNN, RNN, and GCN) have been widely studied to model the spatio-temporal representation of skeleton action sequences under supervised scenarios. However, supervised learning methods typically necessitate substantial data with expensive labels for model training, which is often challenging and costly to obtain. Additionally, labeling and vetting massive amounts of real-world training data is certainly difficult, expensive, or time-consuming. As such, learning effective feature representations with minimal annotations becomes a critical necessity. Thus, in this thesis, we make efforts to explore efficient ways to address this problem.

Particularly, we investigate the weakly-supervised, self-supervised, and one-shot learning methods to solve the skeleton action recognition under the fewer label issue:

- **Collaborative Learning of Gesture Recognition and 3D Hand Pose Estimation:** We introduce a unique collaborative learning network designed for simultaneous gesture recognition and 3D hand pose estimation, capitalizing on joint-aware features. Additionally, we propose a weakly supervised

learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) models.

- **Skeleton Cloud Colorization:** We present the concept of self-supervised action representation learning as a task of repainting 3D skeleton clouds. In this framework, each skeleton sequence is viewed as a skeleton cloud and processed using a point cloud auto-encoder. We introduce an innovative colorization technique for the skeleton cloud where each point is colored according to its temporal and spatial orders in the sequence. These color labels act as self-supervision signals, greatly enhancing the self-supervised learning of skeleton action representations.

- **Multi-Scale Spatial-Temporal Skeleton Matching:** We formulate one-shot skeleton action recognition as an optimal matching problem and design an effective network framework for one-shot skeleton action recognition. We propose a multi-scale matching strategy that can capture scale-wise skeleton semantic relevance at multiple spatial and temporal scales. Building on this, we design a novel cross-scale matching scheme that can model the within-class variation of human actions in motion magnitudes and motion paces.

To validate the efficacy of our proposed approaches, we carried out comprehensive experiments across various datasets. The findings demonstrate a notable improvement over existing methodologies.

Chapter 1

Introduction

1.1 Background

Human Action Recognition, i.e., recognizing and understanding human actions, is vital for various real-world applications. Such recognition is essential for visual surveillance systems, where identifying potentially dangerous human activities is paramount, and for autonomous navigation systems that require understanding human behaviors to ensure safe operations. Additionally, it is significant for diverse applications including video retrieval, human-robot interaction, healthcare, sports analysis, and entertainment.

Initially, research in human action recognition predominantly utilized RGB or grayscale videos, owing to their accessibility and straightforward evaluation. However, recent developments [2–8] have broadened the scope to include a variety of modalities, such as skeleton, depth, infrared sequences, point clouds, event streams, and WiFi signals. Despite this expansion, RGB-based [9–11], depth-based [6, 12, 13], and 3D skeleton-based methods [3, 14–16] remain central to human action recognition. These approaches are valued for their complementary nature and the comprehensive insights they offer into human motion and environmental contexts.

RGB video, depth video, and 3D skeleton data each fulfill unique roles within human action recognition (HAR), presenting their own benefits and challenges. **RGB video** offers rich visual details crucial for identifying actions but struggles with varying lighting and background clutter. **Depth video** offers critical 3D spatial insights, improving recognition within intricate settings but at the expense of visual detail such as textures and colors. **3D skeleton data** simplifies human motion into a series of joints, focusing on movement and posture with high efficiency and robustness to environmental changes, though it may overlook important object interactions. The choice among these data types hinges on the specific needs of the human action recognition system, balancing between detail richness and computational efficiency. Skeleton data is particularly important for its direct capture of human movement, making it indispensable for applications requiring real-time processing and adaptability to diverse settings. Consequently, skeleton-based human action recognition has garnered growing interest among researchers, highlighting its significance in advancing the field.

Specifically, deep neural networks have been widely studied to model the spatio-temporal representation of skeleton sequences under supervised scenarios [3, 14–16]. For instance, Recurrent Neural Networks (RNNs), known for their aptitude in capturing temporal relationships, have been instrumental in skeleton action modeling [3, 15, 17–19]. Convolutional Neural Networks (CNNs) have also been explored to build skeleton-based recognition frameworks by converting joint coordinates to 2D pseudo-images [14, 20–22]. In the recent past, graph convolutional networks (GCNs), which generalize CNNs to graph structures, have achieved increasing attention and have been adopted in many studies [23–28]. However, supervised learning methods typically necessitate a large amount of data with expensive labels for model training.

The standard workflow for skeleton-based human action recognition, illustrated in Figure 1.1, involves two primary methods for acquiring skeleton data: first, through a depth sensor, which is infrequently used in surveillance and similar applications; second, by capturing data with an image sensor and then applying sophisticated

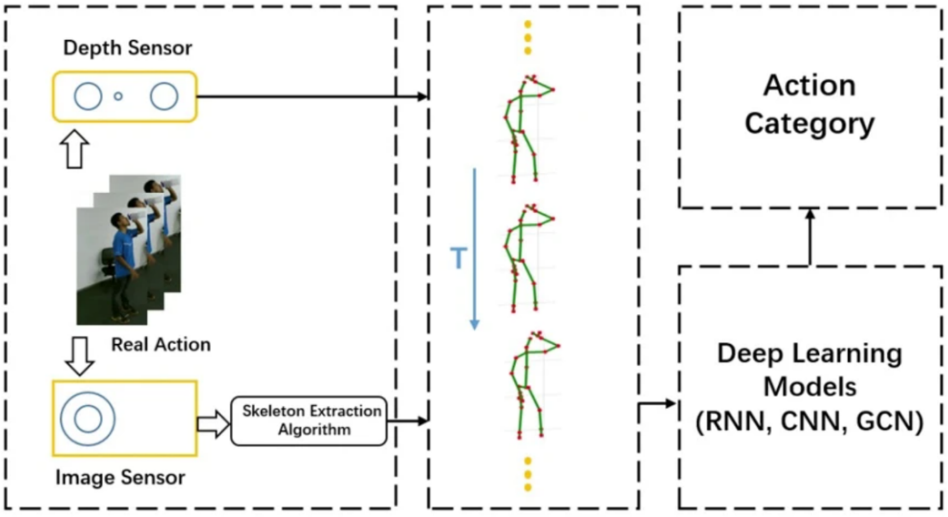


FIGURE 1.1: The workflow for recognizing actions based on skeleton data

skeleton extraction or pose estimation algorithms to produce precise 2D/3D skeleton data. Collecting comprehensive training data for all action categories using either method proves to be challenging and expensive. Moreover, the process of labeling and reviewing vast quantities of real-world training data is arduous, costly, and time-intensive, necessitating the meticulous work of skilled annotators or specialists. Consequently, developing efficient feature representations with minimal annotation is of paramount importance.

In the realm of computer vision, especially the image recognition areas, the scarcity of labeled data also poses a significant barrier to developing robust models. To overcome this challenge, methodologies such as weakly-supervised, self-supervised, and one-shot learning have been pivotal. **Weakly-supervised learning** [29–31] utilizes loosely labeled datasets, enabling models to glean insights from data that may not be precisely annotated, thus circumventing the intensive labor and resources typically required for detailed labeling. **Self-supervised learning** [32–34], by contrast, generates supervisory signals from the data itself, eliminating the need for external labels and unlocking the potential of vast unlabeled datasets. **One-shot learning** [35–37] pushes the boundaries further by enabling models to recognize and generalize from extremely limited examples—often just one or a few instances—thereby drastically reducing the need for extensive labeled datasets.

These methodologies each uniquely mitigate the challenges posed by scarce labels, markedly decreasing the dependence on thoroughly labeled datasets.

With limited research focused on the minimal labeling challenges in skeleton-based action recognition, our thesis endeavors to forge new paths in addressing these intricate issues through the lens of weakly-supervised, self-supervised, and one-shot learning techniques.

1.2 Thesis Contributions

Since learning with fewer labels has emerged as a significant concern in the field of skeleton action recognition, in this thesis, we address this pressing challenge and propose several methods for different fewer-label scenarios, including weakly-supervised, self-supervised, and one-shot learning approaches.

Our main contributions are stated as follows:

- **Collaborative Learning of Gesture Recognition and 3D Hand Pose Estimation (Chapter 3):**

  The first work delves into weakly-supervised learning techniques, specifically focusing on gesture recognition and 3D hand pose estimation. In this context, label information for both tasks is available, and the labels from one task are employed as a weak supervisory signal for the other. We propose a novel collaborative learning network that leverages joint-aware features for both gesture recognition and 3D hand pose estimation simultaneously. To the best of our knowledge, this is the first network that exploits and optimizes joint-aware features for both gesture recognition and 3D hand pose estimation. We design a multi-order feature analysis module that employs a novel slow-fast feature analysis scheme to learn joint-aware motion features which improves gesture recognition greatly. Moreover, we design a multi-scale relation module to learn hierarchical hand structure relations at multiple scales

which enhances the performance of gesture recognition. Lastly, we propose a weakly supervised learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) models. The weakly supervised learning greatly relieves the data annotation burden, especially considering the very limited annotated 3D pose data and the wide availability of annotated hand gesture data.

- **Skeleton Cloud Colorization (Chapter 4):**

  The second work delves deeper into self-supervised learning techniques for 3D skeleton action recognition. In this research, lacking explicit labels for skeleton data, we utilize the data's intrinsic properties as a source of self-supervision. We formulate self-supervised action representation learning as a 3D skeleton cloud repainting problem, where each skeleton sequence is treated as a skeleton cloud and can be directly processed with a point cloud auto-encoder framework. We propose a novel skeleton cloud colorization scheme that colorizes each point in the skeleton cloud based on its temporal and spatial orders in the skeleton sequence. Notably, these color labels act as self-supervision signals, substantially enhancing the efficacy of self-supervised skeleton action representation learning. Additionally, we further extend the design of our skeleton colorization methods with the Masked Skeleton Cloud Repainting task and propose a more powerful coarse-fine alignment framework for better feature pre-training.

- **Multi-Scale Spatial-Temporal Skeleton Matching (Chapter 5):**

  The third work delves into one-shot learning for 3D skeleton action recognition. We formulate one-shot skeleton action recognition as an optimal matching problem and design an effective network framework for one-shot skeleton action recognition. We propose a multi-scale matching strategy that can capture scale-wise skeleton semantic relevance at multiple spatial and temporal scales. Building on this, we design a novel cross-scale matching scheme that can model the within-class variation of human actions in motion magnitude and paces. To the best of our knowledge, this is the first work that exploits

multi-scale representations and cross-scale matching to capture multi-scale skeleton semantic relevance and maintains consistency across motion scales in one-shot skeleton action recognition.

1.3 Organization of Thesis

Chapter 1 introduces the background and challenges of skeleton action recognition with fewer labels, the major contribution of our work, and the organization of the thesis.

Chapter 2 reviews the related work of gesture recognition and 3D hand pose estimation, supervised, Self-Supervised, and One-Shot skeleton-based action recognition methodologies.

Chapter 3 introduces our work that tackles the weakly-supervised learning for both gesture recognition and 3D hand pose estimation with a collaborative learning strategy.

Chapter 4 focuses on self-supervised representation learning in skeleton action recognition with the 3D skeleton cloud colorization method.

Chapter 5 presents our approach to address the one-shot learning challenge in skeleton action recognition with a method based on optimal matching, multi-scale matching, and cross-scale matching.

Chapter 6 summarizes the work included in the thesis and discusses possible future directions.

Chapter 2

Literature Review

In this Chapter, we review the existing methods that are relevant to our proposed methods for skeleton-based action recognition wih fewer labels.

2.1 Gesture Recognition and 3D Hand Pose Estimation

Gesture and Action Recognition. Many early gesture and action recognition methods were developed based on handcrafted features [38–41]. With the advance of deep learning, Convolutional neural networks (CNNs) have been applied to gesture recognition and action recognition. Simonyan and Zisserman [10] proposed a two-stream architecture, where one stream operates on RGB frame, and the other on optical flow. Many works follow and extend their framework [9, 42, 43], all of which utilized optical flow to capture motion information. Wang *et al.* [44] built a new motion representation: RGB difference, which stacks the differences between consecutive frames, to save the time of optical flow extraction. Both the optical flow [10, 44] and RGB difference [44] computations are pre-processed, which is independent of the learning procedure.

Inspired by the above-mentioned works, in our work presented in Chapter 3, we propose a novel multi-order multi-stream feature analysis module, which is operated at the intermediate features that capture more discriminative and representative motion information as compared to the original video data. To further refine this, a slow-fast feature analysis module is added to consolidate the features of both the slow and fast-moving joints at multiple orders, which significantly enhances the gesture-aware features for more reliable gesture recognition.

3D Hand pose estimation. 3D hand pose estimation from RGB images has received much attention recently [45–48]. However, only a few papers [49, 50] focused on performing gesture recognition and 3D hand pose estimation from RGB videos jointly. Tekin *et al.* [50] predicted hand pose and action categories first, and then used the predicted information to do gesture recognition. In contrast, in our work presented in Chapter 3, we introduce an innovative collaborative learning approach. This method capitalizes on joint-aware features for synchronized 3D pose estimation and gesture recognition, iteratively enhancing the performance of both tasks. Furthermore, our approach also facilitates weakly-supervised learning for 3D hand pose estimation.

Joint gesture/action recognition and 3D pose estimation. Gesture (or action) recognition and 3D pose estimation are highly related, leading many papers perform gesture (or action) recognition based on the outcomes of pose estimation. In the Skeleton-based gesture (or action) recognition studies [15, 51–53], joint location (pose) information is used for recognizing the gesture (or action) categories. In RGB-based action recognition, Liu *et al.* [54] proposes to recognize human actions via pose estimation maps. Both Nie *et al.* [55] and Luvizon*et al.* [49] integrate pose estimation and action recognition within a unified network. However, they did not consider these two tasks mutually to optimize the performance of each other, i.e., they performed the two tasks either in a parallel way or in a sequential way.

In contrast to the aforementioned methods, in our work presented in Chapter 3, we design a novel collaborative learning method that boosts the learning of gesture

recognition and 3D hand pose estimation in an *iterative* manner, as shown in Fig. 3.1. To the best of our knowledge, our method is the first that learns gesture-aware and hand pose-aware information for boosting the two tasks progressively.

Weakly-Supervised learning on 3D hand pose estimation. Over recent years, several papers focus on weakly-supervised learning in 3D pose estimation and 3D hand pose estimation areas, since it is hard to obtain 3D pose annotations. Cai *et al.* [56] proposed a weakly-supervised adaptation method by bridging the gap between fully annotated images and weakly-labeled images. Zhou *et al.* [57] transformed knowledge from 2D pose to 3D pose estimation network using re-projection constraint to 2D results. Chen *et al.* [58] used multi-view 2D annotation as the weak supervision to learn geometry-aware 3D representations.

All aforementioned methods still used 2D joint information as the weak supervision to generate 3D hand poses. In contrast, in Chapter 3, we propose that the gesture label can also be used as weak supervision for 3D hand pose estimation. Our experiments show the efficacy of this weak-supervised learning method.

2.2 RGB-Based Action Recognition

Early methods in RGB-based action recognition relied heavily on handcrafted features, as demonstrated in early research [38–41]. However, the progression of deep learning has seen 2D Convolutional Neural Networks (CNNs) becoming pivotal in action recognition domains. A notable advancement came from Simonyan and Zisserman [10], who introduced a dual-stream architecture that analyzes RGB frames in one stream and optical flow in another, laying the foundation for subsequent enhancements in the field. This framework has been expanded upon by [9, 42, 43], all of whom incorporated optical flow to effectively capture motion. Diverging from traditional optical flow methods, Wang *et al.* [44] developed an innovative motion representation called RGB difference, which leverages the frame-to-frame changes, offering a time-efficient alternative to optical flow calculation.

Later, following the introduction of large-scale datasets like Kinetics [59], a significant body of research [9, 42, 60, 61] has evolved 2D Convolutional Neural Networks (CNNs) into 3D structures to capture both spatial and temporal contexts within videos, a critical aspect of human action recognition. Notably, Tran *et al.* [60] introduced the C3D model, a 3D CNN designed to learn spatio-temporal features from raw videos through an end-to-end learning approach. Carreira and Zisserman [9] developed the Inflated 3D CNN (I3D), which extends a 2D CNN into the temporal domain by inflating its convolutional and pooling layers. Xie *et al.* [43] furthered I3D's development by integrating a mix of 3D and 2D convolutional filters within the I3D framework, adding temporally separable convolutions, and implementing spatio-temporal feature gating to improve Human Action Recognition. Diba *et al.* [62] proposed a novel block for embedding into architectures like ResNext and ResNet, designed to analyze the inter-channel correlations in 3D CNNs relative to temporal and spatial features. Feichtenhofer *et al.* [63] crafted a dual-pathway 3D CNN architecture, featuring both slow and fast pathways that process RGB frames at varying frame rates to simultaneously capture semantic content and motion details.

Following the Vision Transformer's (ViT) [64] breakthrough in image recognition, its methodologies were adapted for video recognition. Bertasius *et al.* [65] expanded the application of ViT to videos by segmenting each video into sequences of frame-level patches and introduced a novel attention mechanism that separately processes spatial and temporal attentions within each segment of the model. Arnab *et al.* [66] developed variants of a pure-Transformer network by segregating the Transformer encoder's components across spatial and temporal dimensions for enhanced video analysis. Yan *et al.* [67] unveiled the Multiview Transformers, which feature multiple distinct encoders, each tailored to a specific input representation, with lateral connections between these encoders to seamlessly integrate diverse input video representations. Liu *et al.* [68] transitioned the Swin Transforme [69]r from its original image recognition domain to video recognition, leveraging spatiotemporal locality to inform the model's inductive bias.

2.3 Skeleton-Based Action Recognition

Skeleton-based action recognition has recently garnered significant attention in the research arena. Initial approaches centered on extracting handcrafted features from skeleton sequences to recognize actions. These handcrafted feature methodologies can be broadly categorized into joint-based and body part-based techniques. Due to the strong feature learning capability, recent methods for skeleton-based action recognition pay more attention to deep networks. Deep-learning based skeleton action recognition methods employ Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Graph Convolution Networks (GCNs), or Transformers, to learn skeleton-sequence representation directly. Specifically, RNNs have been widely used to model temporal dependencies and capture motion features for skeleton-based action recognition. For example, Du *et al.* [17] used a hierarchical RNN model to represent human body structures and temporal dynamics of the body joints. Liu *et al.* [3, 15] proposed a 2D Spatio-Temporal LSTM framework to employ the hidden sources of action-related information over both spatial and temporal domains concurrently. Zhang *et al.* [19] added a view-adaptation scheme to the LSTM to regulate the observation viewpoints. Song *et al.* [70] proposed a Deep LSTM with spatial-temporal attention, where a spatial attention sub-network and a temporal attention sub-network work jointly.

CNN-based methods [14, 20–22, 71, 72] have also been proposed for skeleton action recognition. They usually transform the skeleton sequences into skeleton maps of the same target size and then use CNNs to learn the spatial and temporal dynamics or apply temporal convolution on skeleton sequences. For example, Du *et al.* [20] and Li *et al.* [21] transformed a skeleton sequence to an image by treating the joint coordinate (x,y,z) as the R, G, and B channels of a pixel and then adopted CNNs for action recognition. Ke *et al.* [14] transformed the 3D skeleton sequence into three skeleton clips, which are then fed to a CNN network for robust action feature learning. Wang *et al.* [22] presented a "scene flow to action map" representation for action recognition with CNNs. Kim *et al.* [72] utilized the Temporal CNN (TCN),

which provides a way to explicitly learn interpretable spatial-temporal representations for skeleton-based action recognition. Li *et al.* [71] designed an end-to-end convolutional framework for learning co-occurrence features with a hierarchical methodology.

Inspired by the observation that the human 3D skeleton is naturally a topological graph, Graph Convolutional Networks (GCN) have attracted increasing attention in skeleton-based action recognition. For example, Yan *et al.* [73] presented a spatial-temporal GCN to learn both spatial and temporal patterns from skeleton data. Shi *et al.* [16] used a non-local method to learn the individual topology of graphs instead of using the manually designed one. Additionally, Shi *et al.* [16] proposed a two-stream Adaptive GCN with the additional bone information. Peng *et al.* [74] recognized actions by searching for different graphs at different layers via neural architecture search. To reduce computational costs of GCNs, Cheng *et al.* [75] designed a Shift-GCN, which leverages the shift graph operations and lightweight point-wise convolutions. Chen *et al.* [76] proposed a Channel-wise Topology Refinement Graph Convolution (CTR-GC) to learn different topologies in different channels for skeleton-based action recognition. More recently, Chi *et al.* [77] proposed InfoGCN, which combines an information bottleneck framework to learn informative representation and an attention-based graph convolution that infers context-dependent skeleton topology.

The advent of vision transformers has led to the application of transformer-based methodologies in analyzing skeleton data. Recent studies [78–83]. have successfully adapted the Transformer model to recognize actions from skeleton data, taking into account both spatial and temporal dimensions. Specifically, Wang *et al.* [83] introduced the IIP-Transformer, employing self-attention mechanisms to discern the interconnections among joints. Meanwhile, Gao *et al.* [81] developed the FG-STFormer, aimed at understanding the intricate relations between pivotal local joints and the overarching global context across spatial and temporal planes. Zhang *et al.* [80] explored various joint organization strategies to spatiotemporally model the skeleton sequence. Additionally, Ahn *et al.* [84] proposed a spatio-temporal

cross-transformer that includes an encoder and decoder designed to learn feature representations for cross-modal data.

Though the aforementioned methods have demonstrated remarkable results, they all use supervised learning, requiring a large amount of labeled data which is prohibitively time-consuming to collect. In this thesis, we study self-supervised representation learning (Chapter 4) and one-shot learning (Chapter 5) in skeleton-based action recognition, which mitigates the data labeling constraint greatly.

2.4 Self-Supervised Representation learning based Skeleton-Based Action Recognition

Self-supervised action recognition aims to learn effective feature representations by predicting future frames of input sequences or by re-generating the sequences. Most existing methods focus on RGB videos or RGB-D videos. For example, Srivastava *et al.* [85] used an LSTM-based Encoder-Decoder architecture to learn video representations. Luo *et al.* [86] used an RNN-based encoder-decoder framework to predict the sequences of flows computed with RGB-D modalities. Li *et al.* [87] used unlabeled video to learn view-invariant video representations.

Self-supervised skeleton-based action recognition was largely neglected though a few works have attempted to address this challenging task very recently. For example, Zheng *et al.* [88] presented a GAN encoder-decoder to re-generate masked input sequences. Kundu *et al.* [89] adopted a hierarchical fusion approach to improve human motion generation. Su *et al.* [90] presented a decoder-weakening strategy to drive the encoder to learn discriminative action features. In recent years, with the development of contrastive learning, several self-supervised contrastive skeleton-based action recognition methods have emerged. Rao *et al.* [91] used the momentum encoder [34] for contrastive learning with single-stream skeleton sequence, while Li *et al.* [92] proposed a cross-stream knowledge mining strategy to improve the performance with multi-types of skeleton sequences. Guo *et al.* [93]

introduced an extreme augmentation strategy to force the model to learn more general representation by providing harder contrastive pairs. Kim *et al.* [94] proposed GL-Transformer, which is able to effectively capture the global context and local dynamics of the sequence. Zhang *et al.* [95] followed the SimSiam [96] structure and introduces a novel positive-enhanced learning strategy for unsupervised skeleton representation learning. Zhang *et al.* [97] proposed a new hierarchical contrastive learning framework, HiCLR, to take advantage of the strong augmentations.

The aforementioned methods process skeleton sequences frame by frame and extract temporal features from ordered sequences or leverage contrastive learning to process the self-supervised skeleton action recognition by contrastive pairs. In Chapter 4, we instead treat a skeleton sequence as a novel colored skeleton cloud by stacking human joints of each frame together. We design a novel skeleton colorization scheme and leverage the color information for self-supervised spatial-temporal representation learning. Additionally, we introduce two types of colorization strategies (fine-grained and coarse-grained colorization) by assigning colors from different spatial levels and different temporal levels.

2.5 One-Shot learning based Skeleton-Based Action Recognition

One-shot Skeleton Action Recognition has attracted increasing interest in recent years. Leveraging the NTU RGB+D 120 dataset, Liu *et al.* [98] first presented an Action-Part Semantic-Relevance aware (APSR) approach for one-shot skeleton action recognition. Sabater *et al.* [99] introduced a one-shot action recognition approach based on a Temporal Convolutional Network (TCN). Memmesheimer *et al.* [100] proposed formulating the one-shot skeleton action learning problem as a deep metric learning problem. Additionally, Memmesheimer *et al.* [101] presented an image-based skeleton representation, which performs well in the deep metric

learning manner. Ma *et al.* [102] proved that maximally preserving disentangled joint-level spatial features is beneficial to increase representation diversity and recognizability for few-shot classes in one-shot skeleton action recognition. Wang *et al.* [103] proposed JEANIE, which performs the joint alignment of temporal blocks and simulated viewpoint indexes of skeletons between support-query sequences to select the smoothest path without abrupt jumps in matching temporal locations and view indexes. In a more recent study, Wang *et al.* [104] introduced the uncertainty-DTW, which take into account the uncertainty of in frame-wise (or block-wise) features by selecting the path which maximizes the Maximum Likelihood Estimation (MLE). Zhu *et al.* [105] devised a metric that centers on the summation of similarity measures for aligned local embedding.

Most existing methods take the skeleton representation as a whole and measure the skeleton similarity globally which often misses useful structure and temporal information. In contrast, in Chapter 5, we propose to treat one-shot skeleton action recognition as an optimal matching problem and design multi-scale matching and cross-scale matching which capture the scale-wise semantic relevance and maintain the spatial and temporal consistency across different scales, respectively.

Chapter 3

Collaborative Learning of Gesture Recognition and 3D Hand Pose Estimation with Multi-Order Feature Analysis

In our first work, we focus on the co-training on gesture recognition and 3D hand pose estimation. A new learning strategy, collaborative learning, is introduced for these two tasks. In addition, we propose a weakly supervised learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) models. This methodology addresses the challenges associated with fewer labels by leveraging the weakly supervised technique.

3.1 Introduction

Gesture recognition and 3D hand pose estimation are both challenging and fast-growing research topics, which have recently garnered ongoing interest due to their

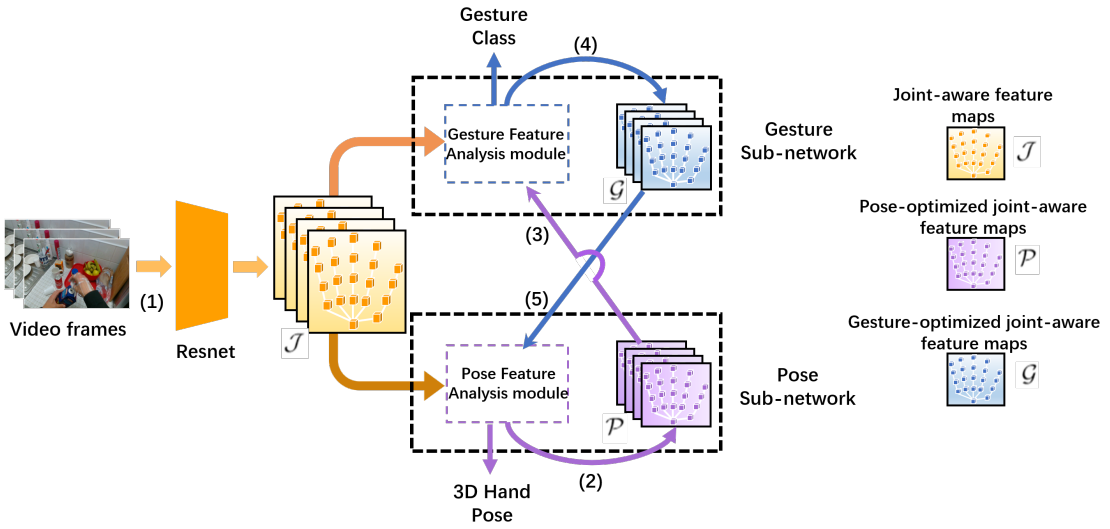


FIGURE 3.1: Overview of our proposed network architecture for gesture recognition and 3D hand pose estimation from videos. The input is video frames, and the output is predicted gesture class of the video and 3D hand joint locations of each video frame. The process flow of our network can be divided into 5 stages: (1) Generating $\mathcal{J}$. (2) Generating $\mathcal{P}$ and Predicting 3D Hand Pose. (3) Aggregating input to Gesture Sub-Network. (4) Generating $\mathcal{G}$ and Recognizing Gesture Class. (5) Aggregating input to Pose Sub-Network. (As shown by the (1) - (5) in this Figure). Stage (2) to (5) are operated in an iterative way (details are introduced in Chapter 3.2.1).

wide range of applications in human-computer interaction, robotics, virtual reality, augmented reality, etc. The two tasks are closely correlated, as they both heavily rely on joint-aware features, *i.e.,* features related to the hand joints [15, 48]. On the other hand, the two tasks are often tackled separately by dedicated systems [46, 51, 53, 56, 106]. Even though some recent efforts [49, 50, 107] attempt to handle the two tasks at one go, they do not take into account to iteratively gain benefits from mutual learning of them.

In this chapter, we propose to perform gesture recognition and 3D hand pose estimation mutually. We design a novel collaborative learning strategy to exploit joint-aware features that are crucial for both tasks, with which gesture recognition and 3D hand pose estimation can learn to boost each other progressively, as illustrated in Fig. 3.1.

Inspired by the successes methodologies [10, 44] that utilize motion information

for recognizing human activity in videos, we exploit motion information for better gesture recognition by focusing more on joint-aware features. Specifically, we distinguish slow-moving and fast-moving hand joints and exploit such motion information in the intermediate network layers to learn enhanced and enriched joint-aware features. In addition to this, we propose a multi-order multi-stream feature analysis module that exploits more discriminative and representative joint motion information according to the derived intermediate joint-aware features.

Additionally, annotating 3D hand poses is often very laborious and time-consuming. To address this issue, we propose a weakly supervised 3D pose estimation technique that is capable of learning accurate 3D pose estimation models from the gesture labels which are widely available in many video data. We observe that weakly supervised learning significantly improves 3D pose estimation, particularly when only a few samples with 3D pose annotations are incorporated. This improvement is largely attributed to the use of joint-aware features, beneficial to both gesture recognition and 3D hand pose estimation tasks. At the other end, the weakly supervised learning approach can also learn accurate gesture estimation models from hand image sequences with 3D pose annotations with similar reasons.

The contributions of this work are summarized as follows:

- We propose a novel collaborative learning network that leverage joint-aware features for both gesture recognition and 3D hand pose estimation simultaneously. To the best of our knowledge, this is the first network that exploits and optimizes the joint-aware features for both gesture recognition and 3D hand pose estimation.
- We design a multi-order feature analysis module that employs an innovative slow-fast feature analysis scheme to learn joint-aware motion features, which significantly improves gesture recognition.
- We propose a multi-scale relation module to understand hierarchical hand structure relations at multiple scales, which clearly enhances the performance of gesture recognition.

- We propose a weakly supervised learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) models. The weakly supervised learning significantly alleviates the burden of data annotation, especially considering the scarcity of annotated 3D hand pose data and the wide availability of annotated hand gesture data.

3.2 Methodology

We predict gesture categories and 3D hand joint locations directly from RGB image sequences as illustrated in Fig. 3.1. Specifically, image sequences, which are centered on the hand, serve as input to a pre-trained ResNet [108] that facilitates the learning of joint-aware feature maps $\mathcal{J}$ (as shown in Fig. 3.1). Subsequently, the learned $\mathcal{J}$ is fed into the pose sub-network and gesture sub-network, which collaboratively learn to generate more discriminative features. The whole network is trained in an end-to-end manner, with further details presented in the subsequent sub-chapters.

3.2.1 Collaborative Learning for Gesture Recognition and 3D Hand Pose Estimation

Gesture recognition and 3D hand pose estimation are both intrinsically related to joint-level features. The location of joints plays a crucial role in skeleton-based action and gesture recognition, while gesture classes also carry valuable information about potential hand postures that is useful for hand pose estimation.

We propose a collaborative learning method that simultaneously learns the gesture features and 3D hand pose features mutually in an iterative way, as illustrated in Fig. 3.1. As described above, the pre-trained ResNet [108] is used to extract the joint-aware feature maps $\mathcal{J}$. Specifically, we equally divide the joint-aware feature

maps $\mathcal{J}$ to N groups, where N is the number of hand joints, *i.e.* $\mathcal{J} = \{\mathcal{J}_i | i = 1, ..., N\}$, and $\mathcal{J}_i$ is the subset of feature maps representing the joint i ($i \in [1, N]$).

Pose Sub-Network: Following previous work [48, 57, 109], we first utilize a Pose Feature Analysis module to estimate the 2D heatmaps based on the intermediate features for generating the 3D hand pose. The Pose Feature Analysis module is composed of two parts: 2D hand pose estimation and depth regression, which is similar to [48, 57, 109]. For the **2D hand pose estimation part**, its input are the joint-aware feature maps $\mathcal{J}$ and its output are N heatmaps (denoted by $\mathcal{H}$). Each heatmap $\mathcal{H}_i$ is a $H \times W$ matrix that represents the 2D probability distribution of each joint in the image.

Follow the deep regression module in [57, 109], we aggregate the joint-aware feature maps $\mathcal{J}$ and the generated 2D heatmaps $\mathcal{H}$ using a 1×1 convolution, followed by a summation operation. The summed feature maps are the input of the **deep regression module**. Here, the 1×1 convolution is used to map the generated 2D heatmaps $\mathcal{H}$ and the joint-aware feature maps $\mathcal{J}$ to the same size. The deep regression module contains a sequence of convolutional layers with pooling and a fully connected layer in order to regress the depth values $D = \{D_i | i = 1, ..., N\}$, where D_i represents the depth value of the i_{th} joint.

Since the output of pose sub-network is the input of the gesture sub-network, and pose sub-network and gesture sub-network operate iteratively (as shown in Fig. 1), we set the input and output of pose sub-network the same size. To achieve this, we first duplicate the depth values to match the heatmaps' size, and then concatenate them with 2D heatmaps. For each joint, the depth value is a scalar, while heatmap's size is $H \times W$. Therefore, we duplicate depth value HW to align with the size of the heatmaps and facilitate feature concatenation. Subsequently, a 1×1 convolution is used to map the concatenated feature maps and the joint-aware feature maps $\mathcal{J}$ to the same size to generate the output of pose sub-network, named pose-optimized joint-aware feature maps $\mathcal{P}$ (see Fig. 3.1).

Gesture Sub-Network: The input of Gesture Sub-Network is obtained by aggregating the joint-aware feature maps $\mathcal{J}$ and pose-optimized joint-aware feature maps $\mathcal{P}$ with 1×1 convolution followed by a summation. The resultant feature maps are fed to the Gesture Feature Analysis module to generate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and gesture category y (see Fig 3.1). Where the Gesture Feature Analysis module contains a sequence of convolutional layers as well as temporal convolution (TCN) layers to get the temporal relation, TCN layers are used here to predict the gesture class y.

Collaborative learning method: As shown in Fig. 3.1, we design a collaborative learning strategy to perform gesture recognition and 3D hand pose estimation in an iterative way. Our proposed framework's learning processes can be described in the following stages:

1. **Generating $\mathcal{J}$**: The pre-trained ResNet [108] is used to learn the joint-aware feature maps $\mathcal{J}$.

2. **Generating $\mathcal{P}$ and Predicting 3D Hand Pose**: The learned feature maps $\mathcal{J}$ are fed to Pose Feature Analysis module (shown in Fig. 3.1) to generate 3D hand poses (2D Heatmaps $\mathcal{H}$ and depth values D), as well as the pose-optimized joint-aware feature maps $\mathcal{P}$.

3. **Aggregating input to Gesture Sub-Network**: The 1×1 convolution is used to generate intermediate feature maps by aggregating the joint-aware feature maps $\mathcal{J}$ and the pose-optimized joint-aware feature maps $\mathcal{P}$.

4. **Generating $\mathcal{G}$ and Recognizing Gesture Class**: The intermediate feature maps are fed to Gesture Feature Analysis module as input to generate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and to recognize gesture category y.

5. **Aggregating input to Pose Sub-Network**: We aggregate the gesture-optimized joint-aware feature maps $\mathcal{G}$ and the joint-aware feature maps $\mathcal{J}$ with 1×1 convolution followed by a summation. The aggregated feature

maps are fed to the next iteration's Pose Sub-Network as input for further feature learning.

6. Stage **2** to **5** repeat in an iterative way to perform gesture recognition and hand pose estimation collaboratively for further improving the performance.

3.2.2 Multi-Order Multi-Stream Feature Analysis

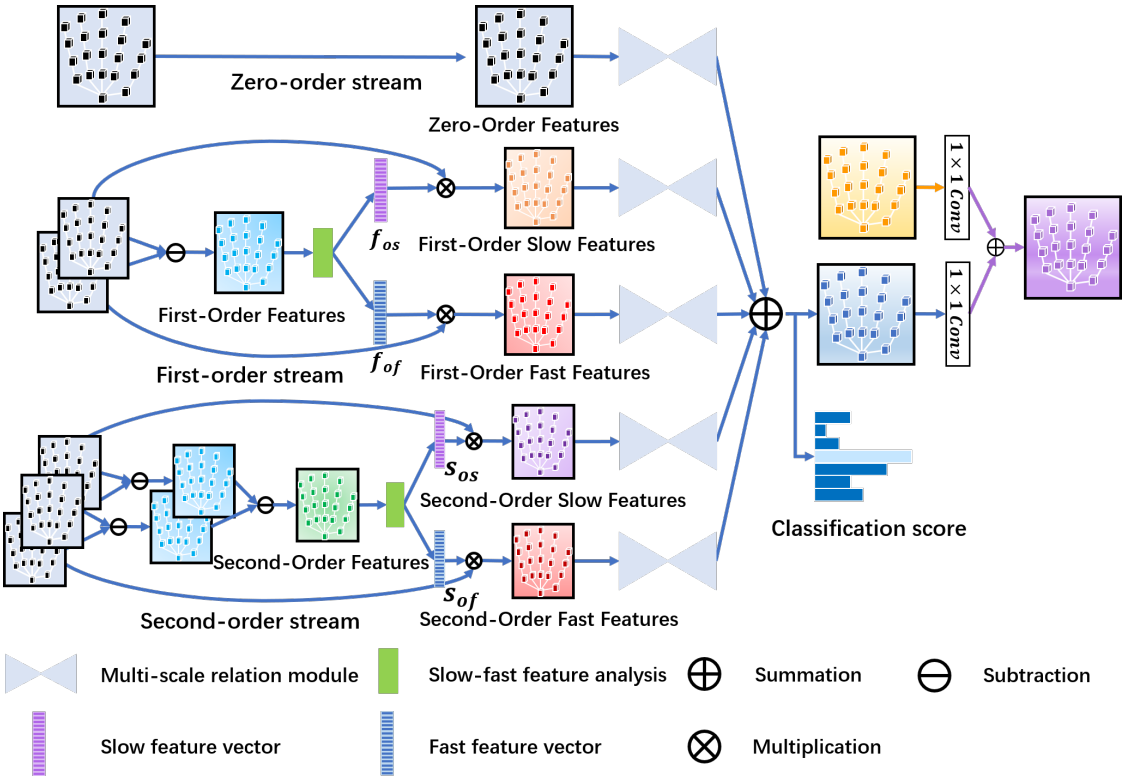


FIGURE 3.2: Illustration of the multi-order multi-stream feature analysis module: With the zero-order features Zo as input, the multi-order multi-stream analysis generates motion information on the intermediate features including first-order slow & fast features and second-order slow & fast features. These four motion features, together with the zero-order features, are fed to five multi-scale relation modules (more details in Fig. 3.3), respectively, to generate gesture-optimized joint-aware feature maps $\mathcal{G}$ and gesture category y. The generated $\mathcal{G}$ are aggregated with joint-aware feature maps $\mathcal{J}$ and fed to the pose sub-network for pose feature learning. Our multi-order multi-stream feature analysis module participates in the Gesture Feature Analysis module, as shown in Fig. 3.1.

As discussed in Chapter 2, prior studies have shown that motion information such as optical flow [10, 44] is crucial in video-based recognition. As we aim to learn

joint-aware features, we propose a multi-order multi-stream feature analysis module as shown in Fig. 3.2 to learn the motion information based on the joint-aware features. The proposed multi-order multi-stream module participates in the Gesture Feature Analysis module (see Fig. 3.1).

Since the pre-trained ResNet [108] and our pose sub-network operate at the image level, their respective feature maps correspond to hand joints within an image. We name the image-level features as **Zero-Order Features** (denote by Zo, which stand for pose information and static information), as shown in the top line of Fig. 3.2, the cubes represent the feature maps of the corresponding hand joints. Zero-Order features form $N \times C \times H \times W$ tensors, where N is the total number of hand joints, C is the number of channels for each hand joint, H and W are the height and width of feature maps, respectively.

First-Order Features can be seen as velocity features. A temporal neighborhood pair of feature maps is constructed from the entire Zero-Order Features as follows:

$$\mathcal{U}_1 = \{\langle Zo_{t-1}, Zo_t\rangle : t \in T\}, \tag{3.1}$$

$$Fo_t = Zo_t - Zo_{t-1}, \tag{3.2}$$

where T denotes the length of input image sequences. First-order features of each joint are calculated by subtracting features of one frame from the previous frame. We derive the first-order features (denoted by Fo) by subtracting Zo_{t-1} from Zo_t as in Eq. 3.2.

Second-Order Features can be seen as the acceleration features. We construct a triplet subset for each frame's features:

$$\mathcal{U}_2 = \{\langle Zo_{t-1}, Zo_t, Zo_{t+1}\rangle : t \in T\}, \tag{3.3}$$

$$So_t = (Zo_{t+1} - Zo_t) - (Zo_t - Zo_{t-1}) = (Fo_{t+1} - Fo_t). \tag{3.4}$$

Similar to the manner of getting first-order features, the second-order features of each joint are calculated by subtracting features of current frame's first-order features from its previous frame's first-order features. We use Fo_{t+1} minus Fo_t to get the second-order features (So) by Eq. 3.4.

Slow-fast Feature Analysis: Slow and fast moving joints are both useful in gesture recognition. The features representing static tendency joints and motion tendency joints encode different levels of motion information. Instead of directly considering these motion features aggregately, we introduce a method for slow-fast feature analysis that distinctly learns these varying levels of motion. Specifically, we develop a strategy to categorically distinguish between slow-moving and fast-moving joint features derived from the First-Order Features (Fo) and Second-Order Features (So). In this way, both static tendency joints and motion tendency joints can be exploited.

Both first-order features and second-order features tensors are of the shape $N \times C \times H \times W$ (the same as the zero-order ones). We first reshape these features to $N \times CHW$ matrices (where N is the number of hand joints), and then calculate the L_2 norm on each joint's first-order and second-order feature vector (with the shape of $1 \times CHW$) from the reshaped features matrices, respectively. This results in N L_2 norm results, denoted by Feature Difference ($FD = \{FD_i | i = 1, ...N\}$, a $N \times 1$ vector). Each FD_i is a value representing the motion magnitude of its corresponding joint. We then adopt Gaussian distributions to derive the feature maps of slow-moving and fast-moving joints. For slow motion analysis, we aim to enhance features from the *more static* joints, i.e., assign larger weights to joints that move more slowly. We use a Gaussian function (with FD_{min} as mean and $(FD_{max} - FD_{min})/3$ as standard deviation) to map FD values to weights (FD_{min}/FD_{max} denotes the min/max FD values). Following this mapping, the weight of the joint with the min/max motion magnitude (FD_{min}/FD_{max}) will be close to 1/0. Given that there are N hand joints, we obtain a $N \times 1$ *slow vector* that contains weights for the features of N joints. In a similar fashion, we aim to enhance features from the *more dynamic* joints using the fast motion analysis module. We thus set FD_{max}

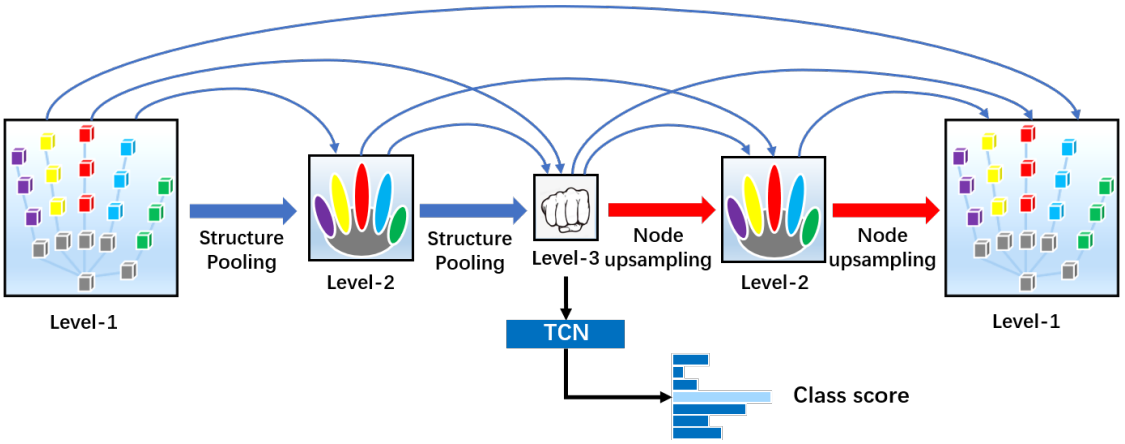


FIGURE 3.3: Illustration of the multi-scale relation module: The multiple scale analysis processes the feature maps from the slow-fast feature analysis at three different levels to generate relations at each level. It interact with the Gesture Sub-network by applying temporal convolution (TCN) on Level 3 (containing global information) to generate the classification scores. Node up-sampling is applied to keep the input and output of the same shape.

and $(FD_{min} - FD_{max})/3$ as the mean and standard deviation. In this way, the joint with the min motion magnitude will be given a weight close to 0, while the joint with the max motion magnitude will assign a weight nearing 1.

When the slow and fast motion analysis modules apply on the first-order and second-order features Fo and So, we obtain four $N \times 1$ vectors that contain weights of features of N joints as shown in Fig. 3.2: 1) First-order slow vector (f_{os}); 2) First-order fast vector (f_{of}); 3) Second-order slow vector (s_{os}); and 4) Second-order fast vector (s_{of}). All these four vectors are used to refine the zero-order features Zo which are first reshaped to an $N \times CHW$ matrix and then multiplied with these four vector separately. The embedding features are then reshaped back to $N \times C \times H \times W$ tensors, namely, first-order-slow features, first-order-fast features, second-order-slow features and second-order-fast features as shown in Fig. 3.2. These four features, together with the zero-order features, are fed to the multi-scale relation module (details to be discussed in Chapter 3.2.3), respectively. Finally, the results of each stream are averaged to derive the gesture-optimized joint-aware feature maps $\mathcal{G}$ and the gesture category y.

3.2.3 Multi-scale relation module

The different levels of semantic information contained in the hierarchical structure of the hand can be defined with different scales. As shown in Fig. 3.3, we present three levels, where level-1 is the local level consisting of the hand joints, and level-2 is the middle level representing five fingers and the palm. For level-3, we see the hand globally as complete holistic information. Following the connection between contiguous scales, we employ structure pooling to perform feature aggregation across these three scales. Further, we employ a Temporal Convolutional Network (TCN) to identify the gesture class y at level-3, leveraging the global information it contains.

Structure pooling refers to the process of applying average pooling over the hand joints by following the hierarchical physical structure of the hand to perform stepwise feature aggregation. We first average features of the joints that belong to each finger or palm, in order to generate features for the five fingers and palm (see Fig. 3.3). Subsequently, we average the features of five fingers and the palm to obtain the final global features, representing the full hand.

Additionally, we calculate a relation matrix for each level to better learn the features at each scale. Take the first level as an example; the whole feature map size is $N \times C \times H \times W$. We first activate it through two embedding functions ($1 \times 1 \times 1$ convolution). The two embedding features are rearranged and reshaped to a $N \times CHW$ matrix and $CHW \times N$ matrix. We then multiply these matrices to obtain a $N \times N$ relation matrix, where each value signifies the degree of relation between each pair of joints. The softmax function is used here to do the normalization. In this way, we can derive relation matrices for each level, which are subsequently used to refine the feature maps at each scale of the hand.

To maintain the input and output of this module in the same shape, we use the node up-sampling method: joints' features from the higher level are duplicated to the corresponding child joint in the lower level. Moreover, we utilize skip-connections (as illustrated by the thin blue arrows in Fig 3.3) across various spatial scales

of the hand to enhance multi-scale hand feature learning and retain the original information. Our multi-scale network participates in each stream of the multi-order multi-stream module, as shown in Fig. 3.2.

3.2.4 Weakly-Supervised Learning Strategy

Weakly-supervised 3D hand pose estimation using gesture labels: Annotation of 3D poses is typically a time-consuming and complex task, making it challenging to accumulate a large volume of video samples with 3D pose annotations for training purposes. In supervised learning, the pose-optimized joint-aware feature maps $\mathcal{P}$ and the gesture-optimized joint-aware feature maps $\mathcal{G}$ are learned based on the joint-aware feature maps $\mathcal{J}$. Consequently, we introduce a weakly-supervised learning technique that utilizes gesture labels as weak supervision to facilitate 3D hand pose estimation. We provide different ratios of training data with 3D pose annotations in the training process.

Weakly-supervised gesture recognition using pose labels: When only a few videos have gesture labels, we can similarly use 3D hand pose annotations as weak supervision to assist in gesture recognition. We incorporate different proportions of training data annotated with gesture labels during training, enhancing the versatility of our approach.

3.2.5 Training

In the training process, we utilize the following loss functions:

2D Heatmaps loss. $L_{2d} = \sum_{n=1}^{N} \|\mathcal{H}_n - \hat{\mathcal{H}}_n\|_2^2$, This loss measures the $L2$ distance between the predicted heatmaps $\mathcal{H}_n$ and the ground-truth heatmaps $\hat{\mathcal{H}}_n$.

Depth Regression loss. $L_{3d} = \sum_{n=1}^{N} \|D_n - \hat{D_n}\|_2^2$, where D_n and $\hat{D_n}$ are the estimated and the ground truth depth values, respectively. L_{3d} is also based on the $L2$ distance.

Classification loss. We use the standard categorical cross-entropy loss to supervise the gesture classification process, which is $L_c = CrossEntropy(y, \tilde{y})$. Here, y stands for the class predicted score, and $\tilde{y}$ is the ground truth category.

Fully-Supervised training strategy. In our implementation, we first fine-tune the ResNet-50 to make it sensitive to human joint information. We then train the entire network in an end-to-end manner with the objective function:

$$L = \lambda_{2d} L_{2d} + \lambda_{3d} L_{3d} + \lambda_c L_c. \tag{3.5}$$

Weakly-Supervised training strategy. Based on Eq. 3.5, we set $\lambda_{2d} = 0$ and $\lambda_{3d} = 0$ when the samples do not have 3D pose annotations and we use gesture categories as weak supervision for 3D hand pose estimation. Analogously, we set $\lambda_c = 0$ for video sequences without gesture labels, where we use 3D pose annotations as weak supervision for gesture recognition.

3.3 Experiments

Implementation Details: We implement our method with the PyTorch framework, and optimize the objective function with the Adam optimizer with mini-batches of size 4. The learning rate starts from 10^{-4}, with a 10 times reduction when the loss is saturated. Following the same setting in [57, 109], the input image is resized to 256×256, and the heatmap resolution is set at 64×64. In the experiment, the parameters in the objective function are set as follows: $\lambda_{2d} = 1$, $\lambda_{3d} = 0.001$ and $\lambda_c = 0.001$. For the **weakly-supervised learning**, we choose 15% to 40% samples as the weakly supervision samples and set $\lambda_{2d} = 0$ and $\lambda_{3d} = 0$ when the samples do not have 3D pose annotations (gesture categories are used as weak supervision for 3D hand pose estimation). Similarly, we set $\lambda_c = 0$ for video sequences without gesture labels, where 3D pose annotations are used as weak supervision for gesture recognition as described in Section 3.2.4.

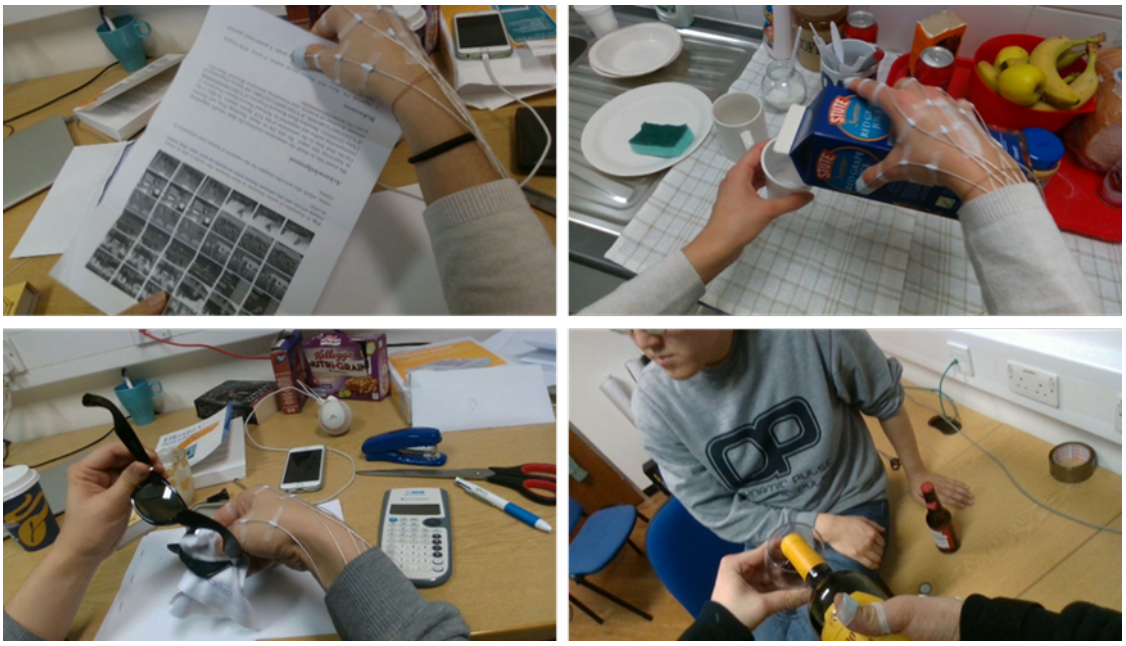


FIGURE 3.4: Sample snapshots from the FPHA [1] dataset.

Following [44], each input video is divided into K segments, and a short clip is randomly selected from each segment in training. For testing, each video is similarly divided into K segments, and one frame is selected from each segment to make sure that temporal space between adjacent frames is equal to T/K. The final classification scores are computed by the average over all clips from each video, and the pose estimation is presented on the image level.

Datasets: We perform extensive experiments on the large-scale and challenging dataset: First-Person Hand Action (FPHA) [1] for simultaneous gesture recognition and 3D hand pose estimation. To the best of our knowledge, this is the only publicly available dataset that provides labels of accurate 2D & 3D hand poses and gesture labels. The dataset consists of 1175 gesture videos with 45 gesture classes. The videos are performed by 6 actors under 3 different scenarios. A total of 105, 459 video frames are annotated with accurate hand pose and action classes. Fig. 3.4 shows some samples of the FPHA datasets. Both 2D and 3D annotations of the total 21 hand keypoints are provided for each frame. We follow the protocol in [1, 50] and utilize 600 video sequences for training and the remaining 575 video sequences for testing.

Evaluation Metrics: We adopt the widely used metrics for the evaluation of gesture recognition and 3D hand pose estimation. For gesture recognition, we directly evaluate the accuracy of video classification. For 3D pose estimation, we use the percentage of correct keypoints (PCK) score that evaluates the pose estimation accuracy with different error thresholds.

3.3.1 Experimental Results

Gesture Recognition: Table 3.1 shows the comparison with state-of-the-art gesture recognition methods. It can be seen that our method outperforms the state-of-the-art by up to 3%, showing its effectiveness in gesture recognition. Additionally, augmenting each of our proposed module (multi-scale relation, multi-order multi-stream, and collaborative learning strategy) results in an incremental improvement in the performance of gesture recognition.

TABLE 3.1: Comparisons to state-of-the-art gesture recognition methods: "Baseline" means 1-iteration network with no multi-order feature analysis and multi-scale relation.

Model	Input modality	Accuracy
Joule-depth [110]	Depth	60.17%
Novel View [4]	Depth	69.21%
HON4D [12]	Depth	70.61%
FPHA + LSTM[1]	Depth	72.06%
Two-stream-color [10]	Color	61.56%
Joule-color [110]	Color	66.78%
Two-stream-flow [10]	Color	69.91%
Two-stream-all [10]	Color	75.30%
[50] - HP	Color	62.54%
[50] - HP + AC	Color	74.20%
[50] - HP + AC + OC	Color	82.43%
Baseline	Color	72.17%
Baseline + multi-scale	Color	78.26%
Baseline + multi-scale + multi-order	Color	83.83%
Baseline + multi-scale + multi-order + 2-iterations	Color	85.22%

3D Hand Pose Estimation: We compare our method with prior works on FPHA as shown in the first graph in Fig. 3.5. Table 3.2 shows three 3D PCK results at

three specific error thresholds. It can be seen that our method outperforms the state-of-the-art with a large range between $0mm$ and $30mm$. Interestingly, even though we use color images, our results are better than [1] that uses depth images which demonstrates the advantage of our proposed method.

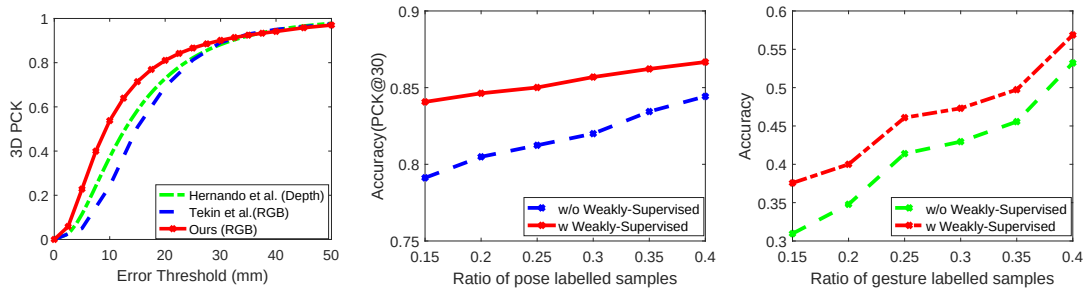


FIGURE 3.5: **Left:** Comparison on 3D hand pose estimation. **Middle and Right:** Comparison between our weakly supervised method and the baseline model.

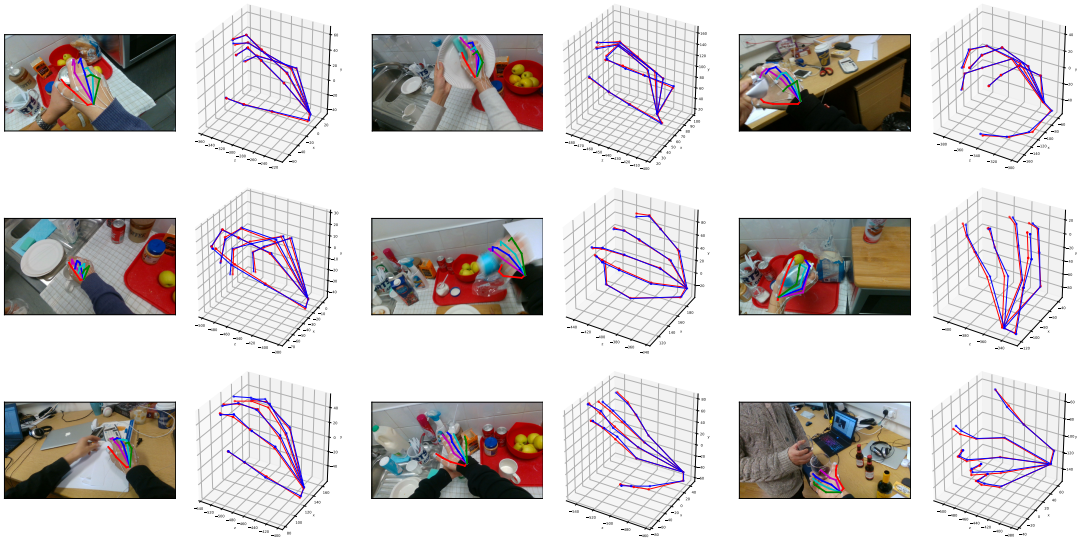


FIGURE 3.6: Qualitative illustration of our proposed method: It shows the predicted 2D poses shown on the original image. It also compares the predicted 3D poses (the blue-color structures) and the Ground Truth 3D poses (the red-color structures).

Qualitative results on 3D Hand Pose Estimation: Fig. 3.6 illustrates 3D pose estimations by our method. We compare the ground truth 3D poses (in blue-color structures) and the predicted 3D pose (in red-color structures) within the same 3D coordinate system. In addition, we also provide the predicted 2D poses in the original RGB image. As demonstrated in Fig. 3.6, our method is capable of accurately predicting 3D poses of different orientations with different backgrounds.

TABLE 3.2: Comparisons on 3D pose estimation: Numbers are the percentage of correct keypoint (PCK) over respective error threshold, more results available in Fig. 3.5 (left). Our results are based on the proposed 2-iterations multi-order structure.

Error Threshold(mm)	PCK@20	PCK@25	PCK@30
Hernando (Depth)[1]	72.13%	82.08%	87.87%
Tekin (RGB)[50]	69.17%	81.25%	89.17%
Ours (RGB)	81.03%	86.61%	90.11%

3.3.2 Weakly-supervised Learning

Weakly-supervised results on 3D hand pose estimation: We present multiple experiments on our weakly-supervised method by providing different ratios (15% to 40%) of samples with pose labels (gesture labels are provided for all training samples) and compare with the baseline that does not use gesture labels. Fig. 3.5 (middle) illustrates the 3D PCK@30 (percentage of correct keypoint when error threshold smaller than 30mm) results for both the baseline and our weakly-supervised method. It can be seen that the 3D hand pose estimation is improved significantly for all labeled ratios when weak supervision is included. This validates the beneficial role of joint-aware features in gestures for enhancing 3D hand pose estimation.

Weakly-supervised results on gesture recognition: We compare our weakly supervised method that uses pose labels as weak supervision for gesture recognition with the baseline, which does not use pose labels. We conduct experiments by providing different ratios of training samples with gesture labels, while the pose labels of all samples are given. As shown in Fig. 3.5 (right), our weakly-supervised learning improves gesture recognition significantly for all labeled ratios. This validates that joint-aware features inherent in hand poses can improve gesture recognition performance greatly.

3.3.3 Ablation Studies

Impact of the number of network iterations: Table 3.3 shows the 3D PCK

TABLE 3.3: Evaluation of our proposed network on gesture recognition and pose estimation with respect to different iteration numbers.

Iteration (itr) number	1-itr	2-itr	3-itr	4-itr	5-itr
Pose estimation (PCK@30)	87.2%	89.3%	89.8%	89.9%	**89.9**%
Gesture recognition accuracy	78.3%	80.9%	81.7%	81.9%	**82.0**%

TABLE 3.4: Evaluation of our proposed gesture recognition network with different combinations of motion features of different orders and slow-fast patterns. (All experiments below are conducted using the 2-iteration network. Δ indicates the accuracy improvement compared to the zero-order model.)

	Network setting	Accuracy	Δ
1	Zero-order	80.87%	
2	Zero-order + First-order slow-fast	82.61%	1.74%
	Zero-order + Second-order slow-fast	83.80%	2.93%
3	Zero-order + First and Second order slow	82.96%	2.09%
	Zero-order + First and Second order fast	82.09%	1.22%
4	Zero-order + First and Second order slow-fast	85.22%	4.35%

results and classification results of our method under different iterations of collaborative learning. It can be seen that our method improves with increasing iterations. This can be expected since hand pose estimation and gesture recognition learn in a collaborative manner and boost each other. It's worth noting that the improvement of 3D PCK and gesture recognition slows down with the increase of iterations. We use the two-iteration network in the experiment for the balance between accuracy and computational complexity. Note all these comparisons are based on the zero-order framework.

Effect of the multi-order module: We analyze the advantage of our proposed multi-order module by implementing four variants as shown in Table 3.4 (part 1, 2, and 4). It can be seen that adding first-order and second-order slow-fast features leads to an accuracy improvement of 1.7% and 2.9%, respectively. Our multi-order module (Zero-order + First and Second order slow-fast) achieves the best accuracy at 85.22%, demonstrating its effectiveness.

Effect of the slow feature and fast feature: We also evaluate the impact of the slow-fast features, and Table 3.4 (part 3) shows the results. It can be seen that the slow features and the fast features can improve the accuracy by 2.1% and 1.2%, respectively, and the best accuracy is obtained when both features are included.

Effect of the multi-scale relation: We also assess the effectiveness of our multi-scale relation module, and Table 3.1 shows experimental results. As illustrated by Table 3.1, removing the multi-scale relation module leads to around 6% accuracy drop as compared with the "Baseline" and "Baseline + multi-scale", showing the benefit of the proposed multi-scale relation.

3.4 Summary

In this chapter, we presented a collaborative learning method for joint gesture recognition and 3D hand pose estimation. Our model learns in a collaborative way to recurrently exploit the joint-aware feature to progressively boost the performance of each task. We have developed a multi-order multi-stream model to learn motion information in the intermediate feature maps and designed a multi-scale relation module to extract semantic information at hierarchical hand structure. To learn our model in scenarios that lack labeled data, we leverage one fully-labeled task's annotations as weak supervision for the other very sparsely labeled task. Our proposed collaborative learning network achieves state-of-the-art performance for both gesture recognition and 3D hand pose estimation tasks.

Chapter 4

Self-Supervised 3D Action Representation Learning with Skeleton Cloud Colorization

In the previous chapter, we introduced a weakly-supervised learning approach for gesture recognition and 3D hand pose estimation. Nonetheless, this method still relies on the label information of another task within the same sample for weak supervision. In this chapter, we present a self-supervised technique that exclusively uses the inherent attributes of the data as its supervisory signal, addressing the challenges associated with fewer labels.

4.1 Introduction

Human action recognition is a fast-developing area due to its relevance to a wide range of applications in human-computer interaction, video surveillance, game control, etc. According to the types of input data, human action recognition can be grouped into different categories such as RGB-based [9–11], depth-based [6, 12, 13], and 3D skeleton-based [3, 14–16], etc. Among these types of data modalities, 3D

skeleton sequences, which represent a human body by the locations of keypoints in the 3D space and characterize informative human motions, have attracted increasing attention in recent years. Compared with RGB videos or Depth videos, 3D skeleton data encodes high-level representations of human behaviors and is generally lightweight and robust to variations in appearances, surrounding distractions, viewpoint changes, etc. Additionally, with the development of depth sensors (*e.g.,* Microsoft Kinect, Asus Xtion, and Intel RealSense3D), skeleton sequences can be easily captured which triggers a large number of supervised methods that have been designed to learn spatio-temporal representations for skeleton-based action recognition.

Specifically, deep neural networks have been widely studied to model the spatio-temporal representation of skeleton sequences under supervised scenarios [3, 14–16]. For example, Recurrent Neural Networks (RNNs) have been explored for modeling skeleton actions since they can capture temporal relations well [3, 15, 17–19]. Convolutional Neural Networks (CNNs) have also been explored to build skeleton-based recognition frameworks by converting joint coordinates to 2D pseudo-images [14, 20–22]. Recently, graph convolutional networks (GCNs), which generalize convolutional neural networks (CNNs) to graphs structures, have achieved increasing attention and have been adopted in many studies [23–28] with outstanding performance [16, 73, 74]. However, most of these methods are fully-supervised which require a large number of labeled training samples that are often costly and time-consuming to collect. How to learn effective feature representations with minimal annotations becomes critically important. Recently, several work [88–90, 92, 111, 112] explores representation learning from unlabeled skeleton data for the task of skeleton action recognition, and the main technical line is to reconstruct skeleton data from the encoded features via certain encoder-decoder structures. Though remarkable progress has been achieved, self-supervised skeleton-based action representation learning remains a concerned problem.

In this work, we propose to represent a skeleton sequence as a 3D skeleton cloud, and design a self-supervised representation learning scheme that learns features

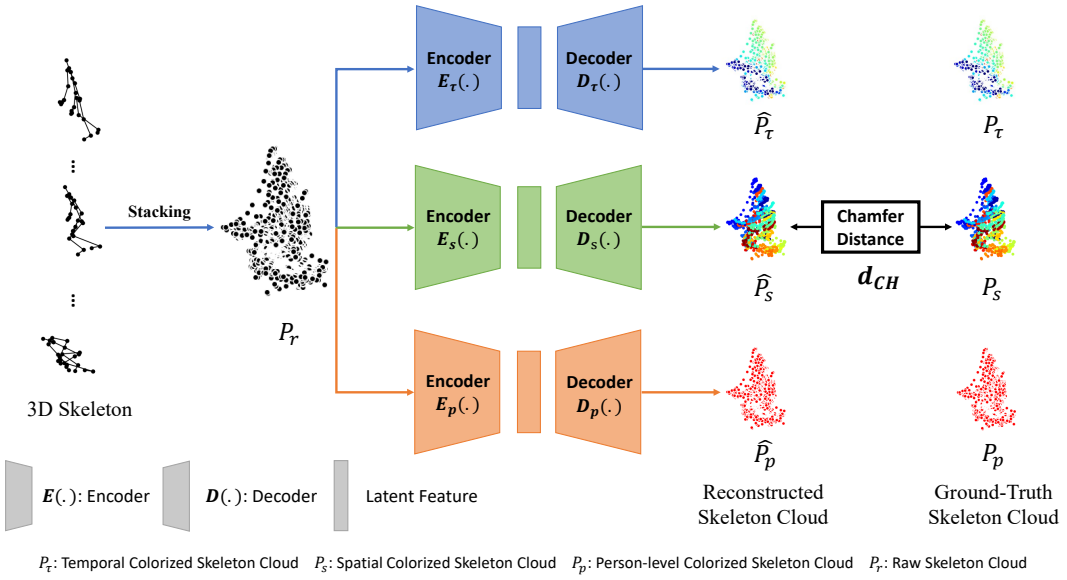


FIGURE 4.1: The pipeline of our proposed self-supervised representation learning with skeleton cloud colorization. Given a 3D skeleton sequence, we first stack it into a raw skeleton cloud P_r and then colorize it into 3 skeleton clouds P_τ, P_s, and P_p (construction details shown in Fig. 4.3 (a), Fig. 4.4 (a) and Fig. 4.5) according to spatial, temporal, and person-level information, respectively. With the three colorized clouds as self-supervision signals, three encoder-decoders (with the same structure but no weight sharing) learn discriminative skeleton representative features.

from spatial and temporal color labels. We treat a skeletal sequence as a spatial-temporal skeleton cloud by stacking the skeleton data of all frames together and colorizing each point (**fine-grained colorization**) in the cloud according to its temporal and spatial orders in the original skeleton sequence. Specifically, we learn spatial-temporal features from the corresponding joints' colors by leveraging a point-cloud based auto-encoder framework as shown in Fig. 4.1. By repainting the whole skeleton cloud, our network can achieve self-supervised skeleton representation learning successfully by learning both spatial and temporal information from skeleton sequences.

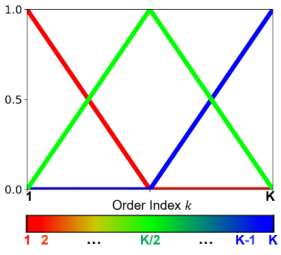
The above-mentioned colorization approaches primarily concentrate on learning spatial and temporal information at the single-frame and single-joint levels, ignoring the temporal dependence across frames and the spatial relationship between

different pairs of joints. To address this issue, we propose a new type of colorization (**coarse-grained colorization**). This method colorizes each point in the cloud according to the order of its corresponding multi-frame segment and body parts. On top of that, we design a two-stream auto-encoder framework that learns the skeleton representation from both fine-grained and coarse-grained color labels, while simultaneously aligning the learned representations between various levels of spatial and temporal colorization.

Inspired by the Mask Auto-Encoder (MAE), we also design a Masked Skeleton Cloud Repainting task. The primary purpose of this task is to pretrain the auto-encoder framework, aiming to facilitate the learning of more discriminative and informative self-supervised representations. To cater to the specific requirements of the skeleton action recognition task, we design five mask sampling strategies: random masking, temporal-only masking, segment masking, spatial-only masking, and body-part masking.

The contributions of this work are summarized as follows:

- We formulate self-supervised action representation learning as a 3D skeleton cloud repainting problem, where each skeleton sequence is treated as a skeleton cloud and can be directly processed with a point cloud auto-encoder framework.
- We propose a novel skeleton cloud colorization scheme that assigns colors to each point in the skeleton cloud based on its temporal and spatial orders in the skeleton sequence. The color labels 'fabricate' self-supervision signals, which significantly enhance the process of self-supervised skeleton action representation learning.
- We further extend the design of our skeleton colorization methods with the Masked Skeleton Cloud Repainting task and propose a more powerful coarse-fine alignment framework for better feature pre-training.



(a) Illustration of the colorization scheme. With the increase of order index k (where $k \in [1, K]$), the corresponding color changes from red to green to blue. (Best viewed in color)

$$r^x_{y,z} = \begin{cases} -2 \times (k/K) + 1, & \text{if } k <= K/2 \\ 0, & \text{if } k > K/2 \end{cases} \quad (1)$$

$$g^x_{y,z} = \begin{cases} 2 \times (k/K), & \text{if } k <= K/2 \\ -2 \times (k/K) + 2, & \text{if } k > K/2 \end{cases} \quad (2)$$

$$b^x_{y,z} = \begin{cases} 0, & \text{if } k <= K/2 \\ 2 \times (k/K) - 1, & \text{if } k > K/2 \end{cases} \quad (3)$$

(b) The general formula of the R, G, B color distribution of Figure 4.2 (a).

	Colorization	x	y	z	k	K
(i)	Temporal (τ)	τ	t	j	t	T
(ii)	Spatial (s)	s	t	j	j	J
(iii)	Coarse-Temporal (ct)	ct	s	j	s	S
(iv)	Coarse-Spatial (cs)	cs	t	p	p	$\mathcal{P}$

(c) The specific parameters (x, y, z, k, K) selection for the designed four types of colorization scheme. t: temporal index; T: total number of frame; j: joint index; J: total number of joint; s: segment index; S: total number of segment; p: part index; $\mathcal{P}$: total number of body part.

FIGURE 4.2: The overall definition of our designed skeleton cloud colorization schemes. **(a)** Illustration of the colorization scheme. **Top**: Definition of each color channel (RGB) when varying k (where $k \in [1, K]$). **Bottom**: The corresponding color of index k. **(b)** The general formula of the R, G, B color distribution of Fig. 4.2 (a). **(c)** The specific parameters selection for the designed four types of colorization scheme. The processings and visualizations of temporal, spatial, coarse-temporal, coarse-spatial colorization can be found in Fig. 4.3 and Fig. 4.4.

- Extensive experiments show that our method outperforms state-of-the-art unsupervised and semi-supervised skeleton action recognition methods by large margins, and its performance is also on par with supervised skeleton-based action recognition methods.

4.2 Method

In this section, we present our masked skeleton cloud colorization representation learning method that converts the skeleton sequence to a skeleton cloud and colorizes each point in the cloud by its spatial-temporal properties. The construction of the skeleton cloud is covered in Section 4.2.1, followed by a detailed description of the colorization process in Section 4.2.2 In Section 4.2.4, we present the repainting pipeline. We introduce the coarse-fine skeleton cloud colorization and masking strategy in Section 4.2.3 and Section 4.2.5, respectively. Finally, the training details are described in Section 4.2.6.

4.2.1 Data Processing

Given a skeleton sequence under a global coordinate system, the j^{th} skeleton joint in the t^{th} frame is denoted as $v_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}]$, where $t \in (1, \cdots, T)$ and $j \in (1, \cdots, J)$ represent the frame and joint indices respectively. Here, T signifies the total number of frames and J stands for the total body joints. Generally, skeleton data is defined as a sequence, and the set of joints in the t^{th} frame are represented as $V_t = \{v_{t,j} | j = 1, ..., J\}$. We propose to treat all the joints in a skeleton sequence as a whole by stacking all frames' data together, and Fig. 4.1 illustrates the stacking framework. We name the stacked data as the **skeleton cloud** and denote it by $P_r = \{v_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}] | t = 1, ..., T; j = 1, ..., J\}$. As a result, the resultant 3D skeleton cloud consists of $N = T \times J$ 3D points in total. We use P_r to represent the raw skeleton cloud so as to differentiate it from the colorized clouds to be described later.

4.2.2 Skeleton Cloud Colorization

Points within our skeleton cloud are positioned with 3D coordinates (x, y, z), akin to a conventional point cloud, which comprises unordered points. The spatial relation and temporal dependency of skeleton cloud points are crucial in skeleton-based action recognition, but they are largely neglected in the aforementioned raw skeleton cloud data. To address this, we propose an innovative skeleton cloud colorization method, designed specifically to leverage these spatial and temporal dependency of skeleton cloud points for skeleton-based action recognition.

Temporal Colorization. Temporal information plays a crucial role in action recognition. To assign each point in the skeleton cloud a temporal feature, we colorize the skeleton cloud points according to their relative time order (from 1 to T) in the original skeleton sequence. Different colorization schemes have been reported, and here we adopt the colorization scheme that uses 3 RGB channels [113], as illustrated in Fig. 4.2 (a). This colorization scheme works by linear mapping,

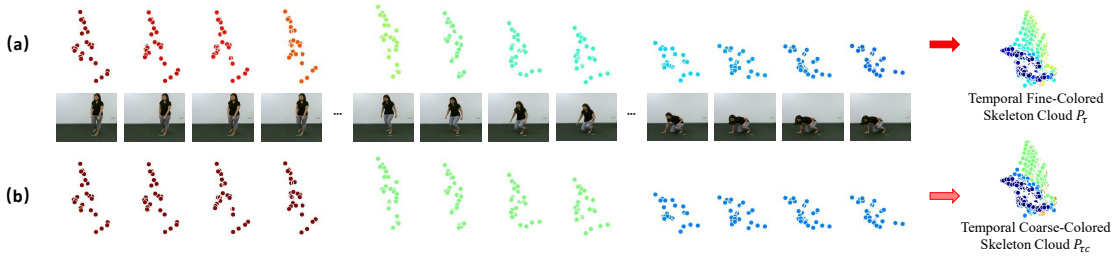


FIGURE 4.3: The pipelines of temporal colorization and temporal coarse-grained colorization. (a) Given a skeleton sequence, the temporal colorization colorizes points based on the relative temporal order t ($t \in [1, T]$) in the sequential data. (b) The coarse-grained temporal colorization colorizes points based on the index of segments s ($s \in [1, S]$). (Best viewed in color)

which can assign similar colors to points of adjacent frames and help the learning of temporal order information. The general formulation of the R, G, B color distribution for Fig. 4.2 (a) is shown in Fig. 4.2 (b), and the specific parameter selection can be found in Fig. 4.2 (c)(i). Based on this, the formulation for temporal colorization is:

$$r^{\tau}_{t,j} = \begin{cases} -2 \times (t/T) + 1, & \text{if } t <= T/2, \\ 0, & \text{if } t > T/2, \end{cases} \tag{4.1}$$

$$g^{\tau}_{t,j} = \begin{cases} 2 \times (t/T), & \text{if } t <= T/2, \\ -2 \times (t/T) + 2, & \text{if } t > T/2, \end{cases} \tag{4.2}$$

$$b^{\tau}_{t,j} = \begin{cases} 0, & \text{if } t <= T/2, \\ 2 \times (t/T) - 1, & \text{if } t > T/2. \end{cases} \tag{4.3}$$

With this colorizing scheme, we can assign unique colors to points from different frames based on the frame index t, as illustrated in Fig. 4.3 (a). More specifically, with this temporal-index based colorization scheme, each point will have a 3-channels feature that can be visualized with red, green, and blue channels (RGB channels) to represent its temporal information. Together with the original 3D coordinate information, the temporally colorized skeleton cloud can be represented as $P_\tau = \{v^{\tau}_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}, r^{\tau}_{t,j}, g^{\tau}_{t,j}, b^{\tau}_{t,j}] | t = 1, ..., T; j = 1, ..., J\}$.

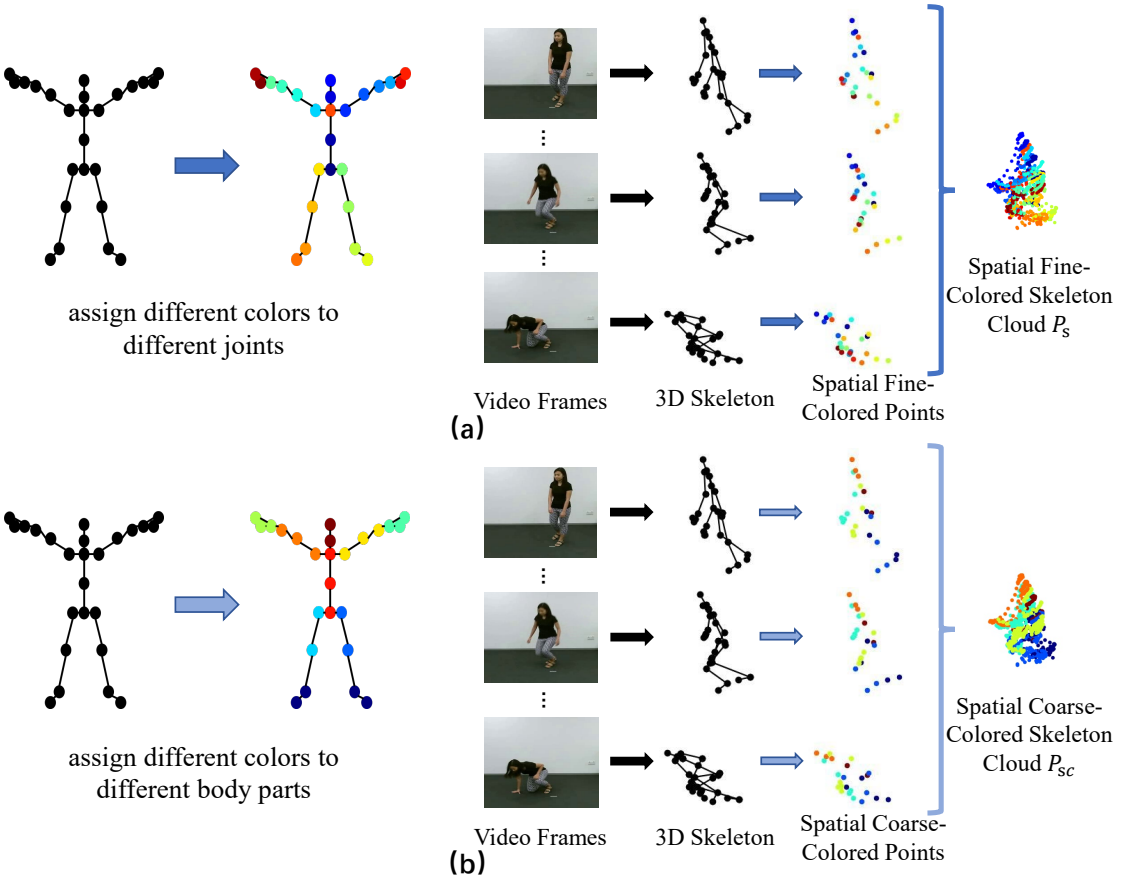


FIGURE 4.4: The pipelines of spatial colorization and spatial coarse-grained colorization. (a) Given a skeleton sequence, the temporal colorization colorizes points based on the relative spatial order j ($j \in [1, J]$) in the sequential data. (b) The spatial colorization colorizes points based on the index of body part p ($p \in [1, P]$). (Best viewed in color)

Spatial Colorization. In addition to temporal data, spatial information plays a crucial role in action recognition. To emphasize this, we employ a similar colorization scheme to colorize spatial information as illustrated in Fig. 4.2. The scheme assigns different colors to different points according to their spatial orders $j \in [1, J]$ (J is the total number of joints in the skeleton cloud of a person), as shown in Fig. 4.4 (a). The specific parameter selection for spatial colorization can be found in Fig. 4.2 (c)(ii). Based on this, the formulation for temporal colorization can be represented as:

$$r^s_{t,j} = \begin{cases} -2 \times (j/J) + 1, \text{ if } j <= J/2, \\ 0, \text{ if } j > J/2, \end{cases} \tag{4.4}$$

$$g^s_{t,j} = \begin{cases} 2 \times (j/J), \text{ if } j <= J/2, \\ -2 \times (j/J) + 2, \text{ if } j > J/2, \end{cases} \tag{4.5}$$

$$b^s_{t,j} = \begin{cases} 0, \text{ if } j <= J/2, \\ 2 \times (j/J) - 1, \text{ if } j > J/2. \end{cases} \tag{4.6}$$

The spatially colorized skeleton cloud is denoted as P_s, which is defined as $P_s = \{v^s_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}, r^s_{t,j}, g^s_{t,j}, b^s_{t,j}] | t = 1, ..., T; j = 1, ..., J\}$. With the increase of the spatial order index of the joint in the skeleton, points will be assigned with different colors that change from red to blue and to green gradually, which is able to represent the linear order information for the joint spatial order and facilitate the learning of spatial order information.

Person-level Colorization. Human actions encompass abundant information regarding person-to-person interactions as in NTU RGB+D [18], which is important to the skeleton action recognition. We therefore propose a person-level colorization scheme for action recognition.

We focus on human interactions involving two people and apply different colors to the points of different people. Specifically, we encode the first person's joints with red and the second person's joints with blue, as illustrated in Fig. 4.5. The person-level colored clouds can thus be denoted by $P_p = \{v^p_{t,j,n} = [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 1, 0, 0] | t = 1, ..., T; j = 1, ..., J; n = 1\} \cup \{v^p_{t,j,n} = [x_{t,j,n}, y_{t,j,n}, z_{t,j,n}, 0, 0, 1] | t = 1, ..., T; j = 1, ..., J; n = 2\}$, where $n = 1$ and $n = 2$ mean that the points belong to the first and the second people, respectively.

Given a raw skeleton cloud, the three colorization schemes thus construct three colorized skeleton clouds P_τ, P_s and P_p that capture temporal dependency, spatial relations, and human interaction information, respectively.

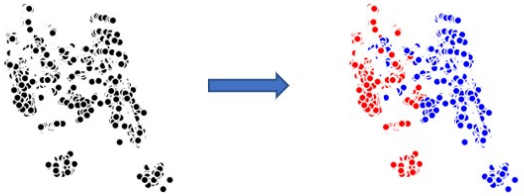


FIGURE 4.5: Person-level colorization. The first person's points will be assigned the red color, and person two will colorize to blue.

4.2.3 Coarse-Fine Skeleton Cloud Colorization

As mentioned above, in our framework, the skeleton cloud is colorized according to each point's temporal and spatial order information. Besides the frame-level and joint-level colorization (fine-grained colorization), coarse-grained colorization can also contribute to spatial-temporal feature learning for self-supervised skeleton action recognition. This is because some actions are often performed at the body part level. For these actions, all the joints from the same informative body part tend to represent similar spatial information. Additionally, some actions contain long-range temporal dependence, which means that several consecutive frames contain similar representations. These observations imply that coarse-grained skeleton cloud colorization is also useful for self-supervised skeleton representation learning.

Spatial Coarse-Grained Colorization. Based on the human physical structure, the human skeleton can be segmented into multiple sections, comprising $\mathcal{P}$ parts, as outlined in [17, 114]. This allows us to undertake a coarse-grained spatial colorization aligned with the divisions of body parts. In order to assign each point in the skeleton cloud a coarser spatial feature, we colorize the skeleton cloud points in accordance with the order of the body parts they correspond to (from 1 to $\mathcal{P}$). The detailed parameter selection for the coarse-grained spatial colorization can be found in Fig. 4.2 (c)(iii). The value distributions of R, G, and B channels can be formulated as follows:

$$r_{t,p}^{cs} = \begin{cases} -2 \times (p/P) + 1, & \text{if } p <= P/2, \\ 0, & \text{if } p > P/2, \end{cases} \tag{4.7}$$

$$g_{t,p}^{cs} = \begin{cases} 2 \times (p/P), \text{ if } p <= P/2, \\ -2 \times (p/P) + 2, \text{ if } p > P/2, \end{cases} \tag{4.8}$$

$$b_{t,p}^{cs} = \begin{cases} 0, \text{ if } p <= P/2, \\ 2 \times (p/P) - 1, \text{ if } p > P/2. \end{cases} \tag{4.9}$$

We use P_{sc} to stand for the spatial coarse-grained colorized skeleton cloud. In contrast to the spatial colorization that assigns different colors to different joints (as shown in Fig. 4.4 (a)), the Spatial Coarse-Grained Colorization assigns different colors to points from different body parts, as shown in Fig. 4.4 (b). In this way, the proposed spatial coarse-grained colorized skeleton cloud P_{sc} contains the spatial relation information within different body parts.

Temporal Coarse-Grained Colorization. Given that consecutive frames often possess similar representations, human skeleton sequences can be segmented into multiple sections, denoted as SS segments. Here, we introduce a coarse-grained temporal colorization at the segment level. Specifically, we employ the colorization scheme (Fig. 4.2 (a)) to colorize segment-level information. For the coarse-grained temporal colorization, the detailed parameter selections are presented in Fig. 4.2 (c)(iv). Based on this, the formulation for temporal coarse-grained colorization is:

$$r_{s,j}^{ct} = \begin{cases} -2 \times (s/S) + 1, \text{ if } s <= S/2, \\ 0, \text{ if } s > S/2, \end{cases} \tag{4.10}$$

$$g_{s,j}^{ct} = \begin{cases} 2 \times (s/S), \text{ if } s <= S/2, \\ -2 \times (s/S) + 2, \text{ if } s > S/2, \end{cases} \tag{4.11}$$

$$b_{s,j}^{ct} = \begin{cases} 0, \text{ if } s <= S/2, \\ 2 \times (s/S) - 1, \text{ if } s > S/2. \end{cases} \tag{4.12}$$

Here, we use the $P_{\tau c}$ to stand for the temporal coarse-grained colorized skeleton cloud. In contrast to temporal colorization, which assigns different colors to individual frames (as shown in Fig. 4.3 (a)), Temporal Coarse-Grained Colorization assigns different colors to points from different segments, as shown in Fig. 4.3 (b). Consequently, the proposed temporal coarse-grained colorized skeleton cloud $P_{\tau c}$ encapsulates the temporal dependency information at the segment level.

Given a raw skeleton cloud, the two coarse-grained colorization schemes thus construct two colorized skeleton clouds $P_{\tau c}$ and P_{sc} that capture temporal dependencies at the segment level and spatial relationships among distinct body parts, respectively.

4.2.4 Repainting Pipeline

Inspired by the success of self-supervised learning, our goal is to extract the temporal, spatial, and interactive information by learning to repaint the raw skeleton cloud P_r in a self-supervised manner. As illustrated in Fig. 4.1, we use colorized skeleton clouds (temporal-level P_τ, spatial-level P_s, and person-level P_p) as three kinds of self-supervision signals, respectively. The framework consists of an encoder $E(.)$ and a decoder $D(.)$. Since we have three colorization schemes, we have three pairs of encoders ($E_\tau(.)$, $E_s(.)$, and $E_p(.)$) and decoders ($D_\tau(.)$, $D_s(.)$, and $D_p(.)$). Below we use the temporal colorization stream as an exemplar to elaborate on the model architecture and training process.

Model Architecture. As mentioned in Section 4.2.2, the obtained skeleton cloud format is similar to that of a normal point cloud. Therefore, we adopt DGCNN [115] (designed for point cloud classification and segmentation) as the backbone of our framework and use the modules before the fully-connected (FC) layers to build our encoder. In addition, we adopt the decoder of FoldingNet [116] as the decoder in our network architecture. Since the input and output of FoldingNet are all $N \times 3$ matrices with 3D positions (x, y, z) only, we expand the feature dimension to 6, thereby enabling the repainting of both position and color information. Assuming

that the input is the raw point set P_r and the obtained repainted point set is $\widehat{P_\tau} = D_\tau(E_\tau(P_r))$, the repainting error between the ground truth temporal colorization P_τ and the repainted $\widehat{P_\tau}$ is computed by using the Chamfer distance:

$$d_{CH}(P_\tau, \widehat{P_\tau}) = Max\left\{A, B\right\}, where, \tag{4.13}$$

$$A = \frac{1}{|P_\tau|} \sum_{v_\tau \in P_\tau} \min_{\widehat{v_\tau} \in \widehat{P_\tau}} \left\| v_\tau - \widehat{v_\tau} \right\|_2, \tag{4.14}$$

$$B = \frac{1}{\left|\widehat{P_\tau}\right|} \sum_{\widehat{v_\tau} \in \widehat{P_\tau}} \min_{v_\tau \in P_\tau} \left\| \widehat{v_\tau} - v_\tau \right\|_2, \tag{4.15}$$

where the term $\min_{\widehat{v_\tau} \in \widehat{P_\tau}} \left\| v_\tau - \widehat{v_\tau} \right\|_2$ enforces that any 3D point v_τ in temporally colorized skeleton cloud P_τ has a matched 3D point $\widehat{v_\tau}$ in the repainted point cloud $\widehat{P_\tau}$. The term $\min_{v_\tau \in P_\tau} \left\| \widehat{v_\tau} - v_\tau \right\|_2$ enforces the matching vice versa. The max operation enforces that the distance from P_τ to $\widehat{P_\tau}$ and vice versa need to be small concurrently.

By using the Chamfer distance, the encoder $E_\tau(.)$ and decoder $D_\tau(.)$ are forced to recover temporal color features for points v_r in the raw skeleton cloud P_r. Similarly, the encoder $E_s(.)$ and decoder $D_s(.)$ are trained to learn spatial color features, and the encoder $E_p(.)$ and decoder $D_p(.)$ are pushed to distinguish the person index and learn interactive information.

Two-Stream Pipeline: As mentioned in Section 4.2.3, we introduce two coarse-grained colorization clouds ($P_{\tau c}$ and P_{sc}) which capture the coarse-grained temporal dependency and spatial relation information. As shown in Fig. 4.6 and Fig. 4.7, together with the original temporal and spatial colorizations, we design a two-stream auto-encoder framework for learning both the fine-grained and coarse-grained spatial-temporal information. Additionally, we do the feature alignment at the latent feature level to align the representations learned from varying spatial

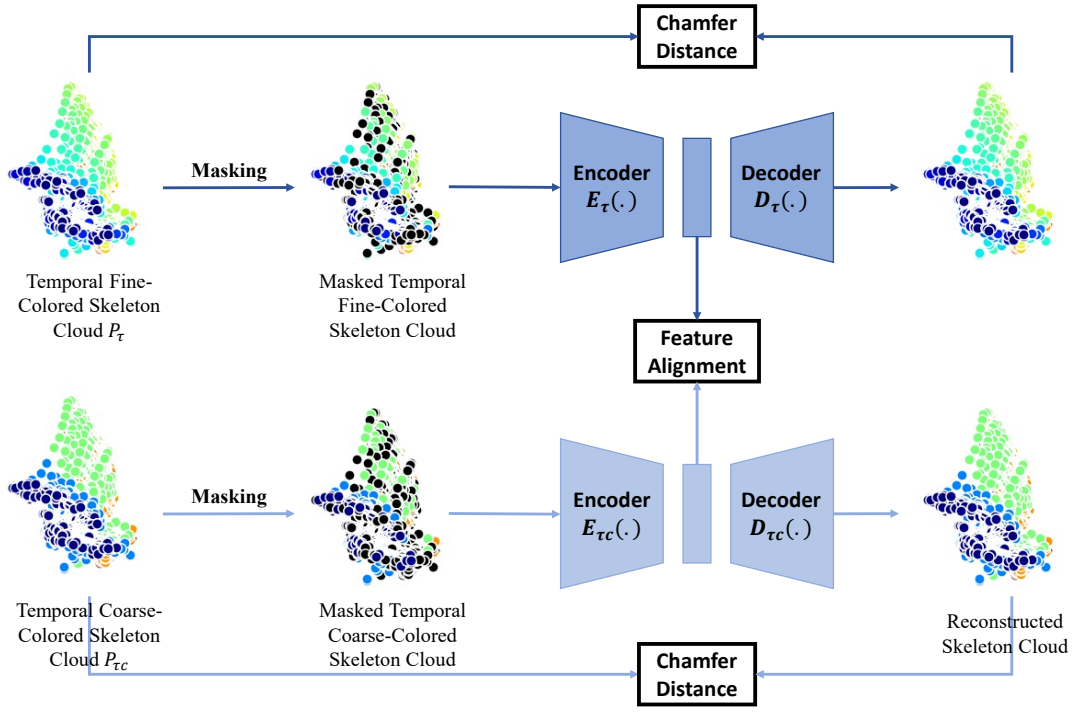


FIGURE 4.6: Illustration of the Coarse-Fine Alignment framework for temporal colorization, which incorporates temporal fine-grained colorization and temporal coarse-grained colorization repainting. The feature alignment is to focus the main auto-encoder (fine-grained one) to learn both the fine and coarse temporal information.

and temporal colorization granularities. This alignment guides the fine-grained encoders (E_τ and E_s) in capturing both fine and coarse spatial-temporal information.

Using Fig. 4.6 as an example, we utilize the Chamfer Distance to measure the repainting error both between the ground truth temporal colorization P_τ and the repainted $\widehat{P_\tau}$, and between the ground truth temporal coarse-grained colorization $P_{\tau c}$ and the repainted $\widehat{P_{\tau c}}$. For feature alignment, we apply the Mean Squared Error (MSE). The latent features for the two streams are defined as $F_\tau = E_\tau(P_\tau)$ and $F_{\tau c} = E_{\tau c}(P_{\tau c})$, respectively. The feature alignment loss, denoted by L_{fa}, is determined using the equation:

$$L_{fa} = \frac{1}{n}\sum_{i=1}^{n}(P_{\tau c}^i - P_\tau^i), \tag{4.16}$$

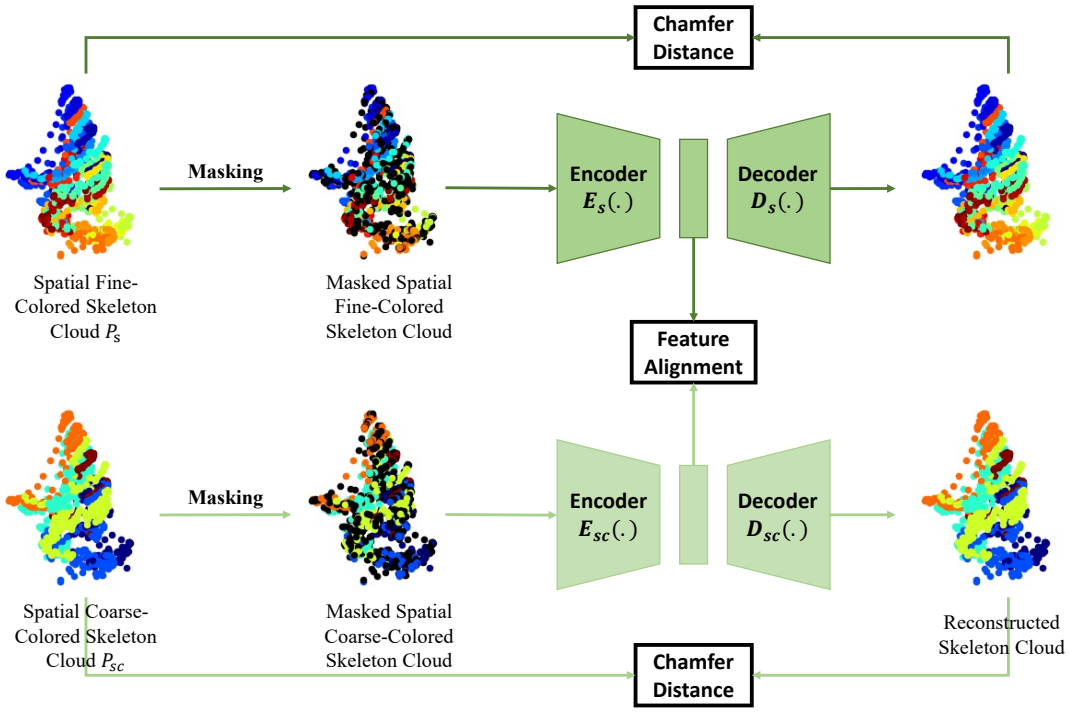


FIGURE 4.7: Illustration of the Coarse-Fine Alignment framework for spatial colorization, which contains spatial fine-grained colorization and spatial coarse-grained colorization repainting pipelines. We introduce an alignment loss at the latent feature level, which enables the latent feature of the fine-grained auto-encoder to perceive both fine and coarse spatial information.

where n indicates the dimension of the latent features. Thus, the final self-supervised models comprise both these two coarse-fine colorization alignment frameworks and the single-stream person-level colorization models.

4.2.5 Masked Skeleton Cloud Modeling

As mentioned in Section 4.2.4, it is non-trivial to repaint P_r to colorize skeleton clouds. Drawing inspiration from BERT [117] and MAE [118], we investigate a masked skeleton cloud modeling strategy for skeleton representation learning based on the proposed auto-encoder framework. Specifically, we formulate a Masked Skeleton Cloud Repainting task that aims at reconstructing the geometric structure and color information of the masked points from partially visible points. The proposed task is based on the proposed auto-encoder, where an encoder is utilized

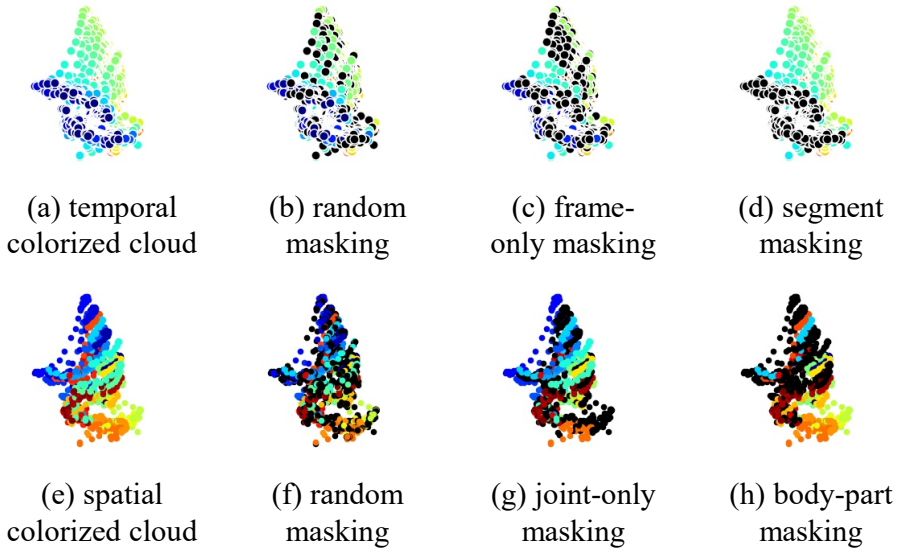


FIGURE 4.8: Mask Sampling Strategy. **(b)** and **(f)**: Random masking that is spacetime-agnostic. **(c)** Frame-only masking: mask all points from the randomly selected frames. **(d)** Segment masking: mask all points from the randomly selected continuous frames. **(g)** Joint-only masking: randomly mask points from selected joints, broadcasted to all frames. **(h)** Body-part masking: randomly mask points from selected body parts, broadcasted to all frames. (Best viewed in color)

to map the visible skeleton points, together with the corresponding color information, into latent representations. The decoder then recreates the geometric information and corresponding color information from these latent representations. By modeling the masked parts, the model attains a comprehensive spatial-temporal understanding of the skeleton sequence. Below, we provide more details of the masking strategies designed for the colorized skeleton cloud.

Mask Sampling Strategy. Five distinct masking strategies are employed in our approach: random masking, frame-only masking, segment masking, joint-only masking, and body-part masking, as shown in Fig. 4.8. (1) In *random masking*, we randomly and uniformly select a subset of skeleton points. Given the importance of temporal correlation and joint interaction for skeleton action recognition, we design the specific masking strategy based on the temporal and spatial dimension (Fig. 4.8 (b) and (f)). (2) In *frame-only masking*, we mask a subset of skeleton points, which belong to randomly selected frames (Fig. 4.8 (c)). (3) In *segment*

masking, we randomly choose a location as the center of the sampled sequence and mask all points from the selected continuous frames (Fig. 4.8 (d)). (4) In *joint-only masking*, we mask a subset of skeleton points, which belong to randomly selected joints (Fig. 4.8 (g)). (5) In *body-part masking*, we mask a subset of skeleton points, which belong to randomly selected body parts (Fig. 4.8 (h)).

Implementation of masking strategy. For the masked points, we initialize all the position and color values to zeros. Consequently, the unmasked point is represented by $[x, y, z, r, g, b]$, while the masked point is initialized as $[0, 0, 0, 0, 0, 0]$.

4.2.6 Training objectives

Self-Supervised Repainting. In this stage, we aim to repaint from three perspectives. When it comes to temporal and spatial colorization, the training objective contains two reconstruction losses (Chamfer distance) on fine-grained and coarse-grained colorization levels and alignment loss (mean squared error, MSE) between latent features, as shown in Fig. 4.6 and Fig. 4.7.

Hence, for temporal colorization, the aggregate loss function is formulated as follows:

$$L_t = d_{CH}(P_\tau, \widehat{P_\tau}) + d_{CH}(P_{\tau c}, \widehat{P_{\tau c}}) + \frac{1}{n}\sum_{i=1}^{n}(P_{\tau c}^i - P_\tau^i), \tag{4.17}$$

where d_{CH} represents the Chamfer distance applied to both the coarse and fine streams. The third component accounts for the alignment of features across varying levels of temporal colorization granularities.

Similarly, the overall loss for the spatial colorization is represented as:

$$L_s = d_{CH}(P_s, \widehat{P_s}) + d_{CH}(P_{sc}, \widehat{P_{sc}}) + \frac{1}{n}\sum_{i=1}^{n}(P_{sc}^i - P_s^i). \tag{4.18}$$

In the case of person-level colorization, where coarse-grained colorization is absent, the training target consists solely of a single Chamfer distance. This is illustrated

in Fig. 5.3. Therefore, the objective function for person-level colorization is established as:

$$L_p = d_{CH}(P_p, \widehat{P_p}) \tag{4.19}$$

Skeleton Action Recognition. As described in Section 4.2.3, we introduce the alignment loss to focus the latent feature of each auto-encoder to contain both coarse and fine spatial-temporal information. To maintain the computation cost the same as that of [119], we only use the fine-grained encoders (*i.e.*, $E_\tau(.)$, $E_s(.)$) in this stage. Together with $E_p(.)$, we obtain three encoders that capture meaningful temporal, spatial, and interaction features, respectively. Leveraging the feature representations from these three encoders, we include a simple linear classifier $f(.)$ on top of the encoder to perform action recognition as in [88–90, 111, 112]. We adopt different settings to train the classifier, including unsupervised, semi-supervised, and supervised settings.

In the unsupervised setting, the encoder is only trained by the skeleton cloud repainting method, and then we train the linear classifier with the encoder fixed by following previous unsupervised skeleton representation learning work [88–90, 111, 112]. In the semi-supervised and supervised settings, the encoder is initially trained with self-supervised representation learning and then fine-tuned in conjunction with the linear classifier as in [111, 120]. The standard cross-entropy loss is employed as the classification loss L_{cls}.

4.3 Experiments

We conducted extensive experiments over several publicly accessible datasets, including NTU RGB+D [18], NTU RGB+D 120 [98], PKU-MMD [121], Northwestern-UCLA [122], UWA3D [13]. Fig. 4.9 shows some samples of these six datasets. The experiments aim to evaluate whether our skeleton cloud colorization scheme can

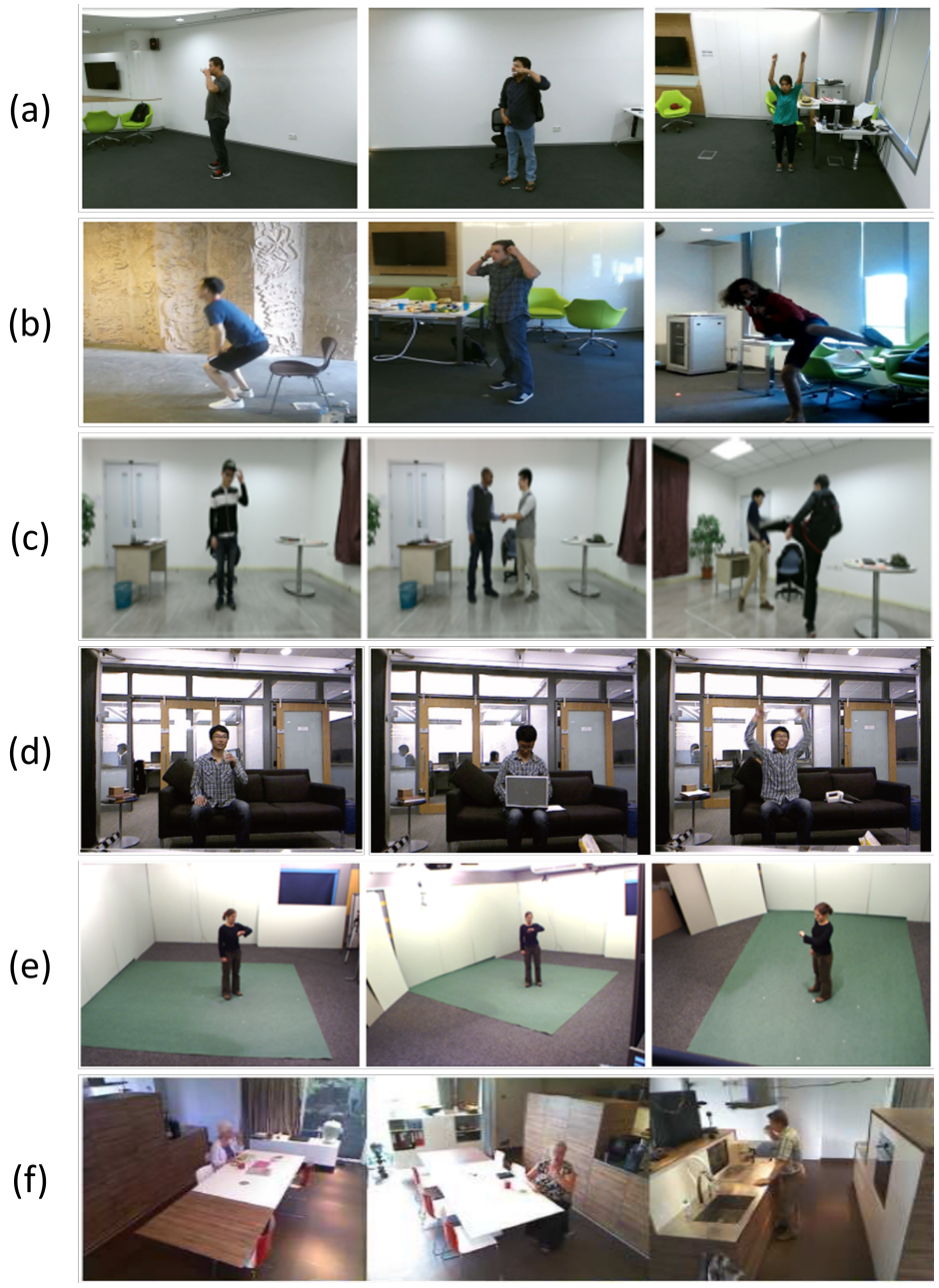


FIGURE 4.9: Sample snapshots from the used datasets. (a) NTU RGB+D dataset; (b) NTU RGB+D 120 dataset; (c) PKU-MMD dataset; (d) NW-UCLA dataset; (e) UWA 3D dataset; (f) Toyota Smarthome dataset.

learn effective self-supervised feature representations for the task of skeleton action recognition. We therefore evaluate different experimental settings, including unsupervised, semi-supervised, and supervised, as well as transfer learning.

4.3.1 Datasets

NTU RGB+D [18]. NTU RGB+D consists of 56880 skeleton action sequences which is the most widely used dataset for skeleton-based action recognition research. In this dataset, action samples are performed by 40 volunteers and categorized into 60 classes. Each sample captures an action that involves at most two subjects, recorded by three Microsoft Kinect v2 cameras from various viewpoints. The authors of this dataset recommend two benchmarks: (1) cross-subject (CS or C-Subject) benchmark, where training data comes from 20 subjects and testing data comes from the remaining 20 subjects; (2) cross-view (CV or C-View) benchmark, where training data comes from camera views 2 and 3, while testing data comes from camera view 1.

NTU RGB+D 120 [98]. NTU RGB+D 120 dataset is currently the largest dataset with 3D joint annotations for human action recognition. The dataset contains 114480 action samples in 120 action classes. Samples are captured by 106 volunteers through three camera views. This dataset contains 32 setups, and each setup represents a unique location and background. The author of this dataset recommends two benchmarks for evaluation: (1) the cross-subject (C-Subject) benchmark: training data comes from 53 subjects, and the testing data comes from the other 53 objects; (2) the cross-setup (C-Setup) benchmark: training data comes from samples with even setup IDs, and testing data comes from samples with odd setup IDs.

PKU-MMD [121]. PKU-MMD is a new large-scale benchmark for continuous multi-modality 3D human action understanding and covers a wide range of complex human activities with well-annotated information. It contains almost 20, 000 action instances and 5.4 million frames in 51 action categories. Each sample consists of 25 body joints. PKU-MMD consists of two subsets, *i.e.*, parts I and II. Part I is an easier version for skeleton action recognition, while part II is more challenging, with more skeleton noise caused by the large view variation. We conduct experiments under the cross-subject protocol on the two subsets respectively.

Northwestern-UCLA (NW-UCLA) [122]. This dataset is captured by three Kinect v1 cameras, and it contains 1494 samples performed by 10 subjects. It contains 10 action classes, and each body has 20 skeleton joints. Following the evaluation protocol in [122], the training set consists of samples from the first two cameras ($V1$, $V2$) and the rest of the samples from the third camera ($V3$) form the testing set.

Multiview Activity II (UWA3D) [13]. UWA3D dataset contains 30 human actions performed 4 times by 10 subjects. 15 joints are recorded, and each action is observed from four views: front (V1), left side (V2), right sides (V3), and top view (V4). The total number of action sequences is 1075. The dataset is challenging due to many views and self-occlusions.

Toyota Smarthome (Smarthome) [123]. Toyota Smarthome is a real-world dataset for daily living action classification and contains 16,115 videos of 31 classes. This dataset poses a unique combination of challenges: high intra-class variation, high-class imbalance, and activities with similar motion and high duration variance. We conduct the transfer learning experiments following the cross-subject (CS) and cross-view1 (CV1) evaluation protocols.

Note that NW-UCLA, UWA3D, and Smarthome datasets only contain single-person actions, so we do not conduct person-level colorization experiments on these three datasets.

4.3.2 Ablation Study

We conduct ablation studies on different datasets to verify the effectiveness of different components of our method.

Effectiveness of Our Skeleton Colorization: We verify the effectiveness of our skeleton cloud colorization on all three learning settings, including unsupervised learning, semi-supervised learning, and fully-supervised learning. We compare our method against three baselines: 1) *Baseline-U*: it only trains the linear classifier

and freezes the encoder, which is randomly initialized; 2) *Baseline-Semi*: the encoder is initialized with random weight instead of pre-training by our self-supervised representation learning; 3) *Baseline-S*: the same as *Baseline-Semi*. We train the encoder and linear classifier jointly with action labels. The input for these three baselines is the raw skeleton cloud without color label information.

Additionally, rather than merely comparing with models that rely on a randomly initialized encoder, we incorporate two commonly used video self-supervised methods to pre-train the encoder, resulting in two stronger baselines: 1) *Motion Prediction*: During self-supervised pre-training, we mask the points from the final 10 frames, allowing the network to predict the motion information of these frames. 2) *Masked Autoencoder*: we randomly mask 25% points during the self-supervised pre-training and push the network to reconstruct the masked information. It is worth noting that both of these two strong baselines are conducted without color information.

We also explore three configurations of skeleton cloud colorization: 1) *'T-Stream' (TS)* that uses temporally colorized skeleton cloud as self-supervision; 2) *'S-Stream' (SS)* that uses spatially colorized skeleton cloud as self-supervision; 3) *'P-Stream' (PS)* that uses the person-level colorized cloud as self-supervision. '2s' means the combination of temporal and spatial steams, and '3s' stands for the combination of temporal, spatial, and person streams.

TABLE 4.1: Comparisons of different network configurations' results with the semi-supervised setting on NTU RGB+D dataset. ('*TS*': Temporal Stream; '*SS*': Spatial Stream; '*PS*': Person Stream; '3s' means three-stream fusion; and the number in parentheses denotes the number of labeled samples per class)

Method	Semi-1		Semi-5		Semi-10		Semi-20		Semi-40	
	CS (7)	CV (7)	CS (33)	CV (31)	CS (66)	CV (62)	CS (132)	CV (124)	CS (264)	CV (248)
Baseline-Semi	27.1	28.1	46.0	50.6	55.1	60.7	60.9	69.1	64.2	73.7
Motion prediction-Semi	36.2	38.3	52.9	56.2	58.5	64.4	64.0	70.8	69.3	76.7
Masked autoencoder-Semi	35.5	38.6	52.4	55.6	59.4	64.3	63.9	71.7	69.5	76.9
'*TS*' Colorization	42.9	46.3	60.1	63.9	66.1	73.3	72.0	77.9	75.9	82.7
'*SS*' Colorization	40.2	43.1	54.6	60.0	60.1	68.1	64.2	73.1	69.1	77.6
'*PS*' Colorization	37.9	40.1	51.2	56.0	56.8	63.2	61.9	70.2	65.8	74.6
'*TS* + *SS*' Colorization	48.1	51.5	64.7	69.3	70.8	78.2	75.2	81.8	79.2	86.0
'*TS* + *PS*' Colorization	46.9	50.5	63.9	68.1	69.8	77.0	74.9	81.3	78.3	85.4
'*SS* + *PS*' Colorization	43.2	46.7	58.3	64.4	64.2	72.0	69.0	77.1	73.2	81.5
3s-Colorization	**48.3**	**52.5**	**65.7**	**70.3**	**71.7**	**78.9**	**76.4**	**82.7**	**79.8**	**86.8**

TABLE 4.2: Comparisons of different network configurations' results with the semi-supervised setting on NW-UCLA dataset. ('*TS*': Temporal Stream; '*SS*': Spatial Stream; '2s' means two-stream fusion)

Method	Semi-1	Semi-5	Semi-10	Semi-15	Semi-30	Semi-40
Baesline-Semi	34.3	46.4	54.9	61.8	69.1	70.2
Motion prediction	38.8	51.6	58.5	64.6	69.7	73.0
Mask autoencoder	37.8	49.7	55.3	64.6	71.1	73.8
'*TS*' Colorization	40.6	55.9	71.3	74.3	81.4	83.6
'*SS*' Colorization	39.1	54.2	66.3	70.2	79.1	80.8
2s-Colorization	**41.9**	**57.2**	**75.0**	**76.0**	**83.0**	**84.9**

Tables 4.1, 4.2, and 4.3 show experimental results. It can be seen that all three colorization strategies (*i.e.,* temporal-level, spatial-level, and person-level) achieve significant performance improvement as compared with the baseline, demonstrating the effectiveness of our proposed colorization technique. Furthermore, though the person-level colorization stream does not perform as well as the other two streams on the NTU RGB+D, it improves the overall performance while collaborating with the other two. Moreover, when compared with the two stronger baselines (*i.e.,* motion prediction and masked autoencoder), our proposed color repainting strategy still clearly outperforms them by a larger margin.

Selection of Segment Size and Body Part Scale: As introduced in Section 4.2.3, the coarse-grained skeleton colorization is based on the segment level and body part level.

Table 4.4 shows the classification results of our method with different segment sizes and body part scales on NTU RGB+D C-Subject protocol, under the unsupervised setting. For the segment size, we can see that assigning each 5 consecutive frames on temporal color is the most beneficial for temporal feature learning.

The human skeleton can be divided into ten body parts [114] (scale 1, *i.e.,* Neck, Trunk, Right arm, Right hand, Left arm, Left hand, Right leg, Right foot, Left leg, Left foot) or six body parts [17] (scale 2, *i.e.,* Torso, Right upper limb, Left upper limb, Right lower limb, and Left lower limb) based on the human physical structure.

TABLE 4.3: Comparisons of different network configurations' results with unsupervised and supervised settings on NTU RGB+D and NW-UCLA dataset.('*TS*': Temporal Stream; '*SS*': Spatial Stream; '*PS*': Person Stream; '3s' means three-stream fusion)

Dataset	NTU-CS	NTU-CV	NW-UCLA
Unsupervised Setting			
Baseline-U	61.8	68.4	78.6
Motion Prediction-U	65.7	75.0	82.1
Masked Autoencoder-U	65.9	75.3	83.6
'*TS*' Colorization	71.6	79.9	90.1
'*SS*' Colorization	68.4	77.5	87.0
'*PS*' Colorization	64.2	72.8	–
'*TS* + *SS*' Colorization	74.6	82.6	**91.1**
'*TS* + *PS*' Colorization	73.3	81.4	–
'*SS* + *PS*' Colorization	69.6	78.6	–
3s-Colorization	**75.2**	**83.1**	–
Supervised Setting			
Baseline-S	76.5	83.4	83.8
'*TS*' Colorization	84.2	93.1	92.7
'*SS*' Colorization	82.3	91.5	90.4
'*PS*' Colorization	81.1	90.3	–
'*TS* + *SS*' Colorization	86.3	94.2	**94.0**
'*TS* + *PS*' Colorization	86.4	94.1	–
'*SS* + *PS*' Colorization	85.0	93.0	–
3s-Colorization	**88.0**	**94.9**	–

TABLE 4.4: Ablation studies on the Segment Size and Body Part Scale. The experiments conduct on NTU RGB+D and NW-UCLA datasets under the unsupervised setting.

Segment Step Size	NTU-CS	NTU-CV	NW-UCLA
2	72.2	82.3	90.1
4	72.8	82.2	89.2
5	**73.2**	**82.6**	**91.0**
8	73.1	82.5	90.8
10	72.5	81.8	90.3
20	72.5	82.1	89.9

Spatial scale	NTU-CS	NTU-CV	NW-UCLA
1 (10 parts)	**72.5**	**82.1**	**87.9**
2 (6 parts)	71.3	81.9	86.8

As shown in Table 4.4, the performance of coarse-grained spatial colorization with scale 1 (10 parts) is better.

Effectiveness of Masking Strategy and Coarse-Fine Alignment framework We conduct experiments on NTU RGB+D (C-Subject protocol) to verify

TABLE 4.5: o

n NTU RGB+D and NW-UCLA datasets.]Linear evaluation results compared with Skeleton Colorization [31] on NTU RGB+D and NW-UCLA datasets. 'Δ' represents the gain compared to [31] with the same stream data. (C-F stands for coarse-fine; '*TS*':Temporal Stream; '*SS*':Spatial Stream; '*PS*':Person Stream; '2s' means two-stream fusion; '3s' means three-stream fusion)

Method	**NTU RGB+D**				**NW-UCLA**	
	C-Subject		C-View			
	Acc.	Δ	Acc.	Δ	Acc.	Δ
'*TS*' Colorization	71.6		79.9		90.1	
'*TS*' Masked Colorization	72.1	↑ 0.5	81.1	↑ 1.2	90.5	↑ 0.4
'*TS*' C-F Masked Colorization (Ours)	73.2	↑ 1.6	82.6	↑ 2.7	91.0	↑ 0.9
'*SS*' Colorization	68.4		77.5		87.0	
'*SS*' Masked Colorization	69.6	↑ 1.2	80.1	↑ 2.6	87.3	↑ 0.3
'*SS*' C-F Masked Colorization (Ours)	72.5	↑ 4.1	82.1	↑ 4.6	87.9	↑ 0.9
'*PS*' Colorization	64.2		72.8		–	
'*PS*' Masked Colorization	67.9	↑ 3.7	77.1	↑ 4.3	–	–
2s-Colorization	–		–		91.1	
2s-Masked Colorization	–	–	–	–	91.4	↑ 0.3
2s-C-F Masked Colorization (Ours)	–	–	–	–	92.0	↑ 0.9
3s-Colorization	75.2		83.1		–	
3s-Masked Colorization	77.2	↑ 2.0	85.8	↑ 2.7	–	–
3s-C-F Masked Colorization (Ours)	**79.1**	↑ 3.9	**87.2**	↑ 4.1	–	–

the effectiveness of our Coarse-Fine Alignment framework and masking strategy under the unsupervised setting. All 'masked' experiments in this ablation study are conducted with 25% random masking.

As shown in Table 4.5, the masking task enhances performance across all three colorization streams, and the proposed Coarse-Fine Alignment framework is able to cause further improvements, which demonstrates the benefit of these techniques.

Effectiveness of Different Masking Strategies and Masking Ratios: To find a proper masking strategy for our method, we conduct experiments varying both the masking types and ratios on the temporal colorization and spatial colorization streams, respectively. This ablation study is conducted on the NTU RGB+D dataset C-Subject protocol under the unsupervised setting. The experimental results are presented in Tables 4.6 and 4.7.

It can be seen that random masking does not work well in both scenarios. We hypothesize that the spatial and temporal relationship is important for skeleton action recognition, and that random sampling can be an overly difficult task in

TABLE 4.6: Ablation study on temporal masking strategy, including the temporal random masking (Fig. 4.8 (b)), frame-only masking (Fig. 4.8 (c)), and segment masking (Fig. 4.8 (d)). The experiments conduct on NTU RGB+D under the unsupervised setting.

(b) Random Masking				(c) Frame-only Masking				(d) Segment Masking			
Mask Ratio	NTU-CS	NTU-CV	NW-UCLA	Mask Frame Number	NTU-CS	NTU-CV	NW-UCLA	Mask Segment Length	NTU-CS	NTU-CV	NW-UCLA
0.25	73.2	82.6	91.0	5	71.7	82.4	89.5	5	72.8	82.6	89.7
0.50	72.8	83.1	89.5	10	73.4	82.7	89.5	10	73.5	83.1	90.1
0.75	72.8	82.8	90.1	15	72.8	83.1	**91.2**	15	**74.0**	**83.5**	**91.2**
				20	73.4	82.8	90.1	20	73.4	83.3	89.9
				30	73.5	82.3	89.7	30	73.8	83.3	89.5

our scenarios. In terms of temporal colorization, segment masking with a 15-frame segment length results in the best performance. For spatial colorization, 10-joint masking performs the best. Therefore, we leverage these two masking strategies in the main experiments.

TABLE 4.7: Ablation study on spatial masking strategy, including the spatial random masking (Fig. 4.8 (f)), joint-only masking (Fig. 4.8 (g)), and body-part masking (Fig. 4.8 (h)). The experiments conduct on NTU RGB+D and NW-UCLA under the unsupervised setting.

(f) Random Masking				(g) Joint-only Masking				(h) Body-part Masking			
Mask Ratio	NTU-CS	NTU-CV	NW-UCLA	Mask Joint Number	NTU-CS	NTU-CV	NW-UCLA	Mask Part Number	NTU-CS	NTU-CV	NW-UCLA
0.25	72.5	82.1	87.9	5	72.5	82.3	87.9	2	72.2	82.2	87.5
0.50	72.4	81.9	87.4	10	**72.8**	**82.6**	**88.3**	4	72.2	82.3	87.7
0.75	71.7	81.4	86.2	15	72.3	82.3	87.5	6	72.0	82.4	**88.3**
				20	72.5	82.2	–	8	72.0	81.7	87.1

Noting that we only have one style for person-level colorization, in which color information is not related to temporal and spatial information, we apply random masking (25%) for the person stream in the main experiments.

4.3.3 Comparison with the State-of-the-art Methods

We conduct extensive experiments under four settings, including unsupervised learning, semi-supervised learning, supervised learning, and transfer learning. Additionally, we investigate three configurations of skeleton cloud colorization. '2s' means the combination of temporal and spatial steams, and '3s' stands for the combination of temporal, spatial, and person streams.

TABLE 4.8: Comparisons to state-of-the-art self-supervised skeleton action recognition methods on NTU RGB+D, NTU RGB+D 120, PKU-MMD, UWA3D, and NW-UCLA datasets. (C-F stands for coarse-fine; '2s' means three-stream fusion; '3s' means three-stream fusion)

Method	Backbone	Stream	NTU RGB+D		NTU RGB+D 120		PKU-MMD		UWA3D		NW-UCLA
			C-Subject	C-View	C-Subject	C-Setup	I	II	V3	V4	
LongT GAN [88] (AAA1 2018)	GRU	1	39.1	52.1	35.6	39.7	68.7	26.5	53.4	59.9	74.3
P&C FS-AEC [90] (CVPR 2020)	GRU	1	50.6	76.3	–	–	–	–	59.5	63.1	83.8
P&C FW-AEC [90] (CVPR 2020)	GRU	1	50.7	76.1	41.1	44.1	59.9	25.5	59.9	63.1	84.9
MS^2L [111] (ACMMM 2020)	GRU	1	52.6	–	–	–	64.9	27.6	–	–	76.8
PCRP [124] (TMM 2021)	GRU	1	53.9	63.5	41.7	45.1	–	–	–	–	87.0
AS-CAL [91] (Information Sciences 2021)	LSTM	1	58.5	64.8	–	–	–	–	–	–	–
GLTA-GCN [125] (ICME 2022)	GCN	2	61.2	81.2	49.1	51.1	–	–	61.5	68.2	–
MCAE-MP [126] (NeurIPS 2021)	MCAE	1	65.6	82.4	52.8	54.7	–	–	–	–	84.9
Taxonomy-SSL [127] (WACV 2022)	GRU	1	67.0	76.3	59.1	61.5	–	–	–	–	86.1
CRRL [128] (TIP 2022)	GRU	1	67.6	73.8	57.0	56.2	–	–	–	–	83.8
ST-CL(×4) [129] (TMM 2021)	GCN	4	68.1	69.4	54.2	55.6	–	–	46.0	44.0	81.2
EnGAN-PoseRNN [89] (WACV 2019)	RNN	3	68.6	77.8	–	–	–	–	–	–	–
H-Transformer [130] (ICME 2021)	Transformer	1	69.3	72.8	–	–	–	–	–	–	83.9
CP-STN [131] (ACML 2021)	GCN	1	69.4	76.6	55.7	54.7	–	–	–	–	–
SeBiReNet [112] (ECCV 2020)	GRU	2	–	79.7	–	69.3	–	–	53.9	61.6	80.3
SKT [132] (ICME 2022)	GCN	1	72.6	77.1	62.6	64.3	–	–	–	–	–
2s-Colorization [119] (ICCV 2021)	DGCNN	2	–	–	–	–	–	–	70.0	70.6	91.1
3s-Colorization [119] (ICCV 2021)	DGCNN	3	75.2	83.1	64.3	67.5	87.2	47.1	–	–	–
GL-Transformer [94] (ECCV 2022)	Transformer	1	76.3	83.8	66.0	68.7	–	–	–	–	90.4
Skeleton-Contrastive [133] (ACMMM 2021)	GRU+CNN	2	76.3	85.2	67.1	67.9	80.9	36.0	–	–	–
3s-CrosSCLR [92] (CVPR 2021)	GCN	3	77.8	83.4	67.9	66.7	84.9	21.1	–	–	–
3s-HiCLR [97] (AAAI 2023)	GCN	3	78.8	83.1	67.3	69.9	–	–	–	–	–
3s-AimCLR [93] (AAAI 2022)	GCN	3	78.9	83.8	68.2	68.8	–	–	–	–	–
2s-C-F Masked Colorization (Ours)	DGCNN	2	–	–	–	–	–	–	**71.7**	**73.8**	**92.0**
3s-C-F Masked Colorization (Ours)	DGCNN	3	**79.1**	**87.2**	**69.2**	**70.8**	**89.2**	**49.8**	–	–	–

4.3.3.1 Unsupervised Learning

In the unsupervised setting, the feature extractor (*i.e.,* the encoder $E(.)$) is trained with our proposed skeleton cloud colorization self-supervised representation learning approach. Then the feature representation is evaluated by the simple linear classifier $f(.)$, which is trained on top of the frozen encoders $E(.)$. Such experimental setting for unsupervised learning has been widely adopted and practiced in prior studies [88, 89, 111, 112]. Here for a fair comparison, we use the same setting as this prior work.

We compare our skeleton cloud colorization method with prior unsupervised methods on NTU RGB+D, NTU RGB+D 120, PKU-MMD, NW-UCLA, and UWA3D datasets, as shown in Table 4.8. It can be seen that our proposed coarse-fine masked colorization method can achieve state-of-the-art performances on all five datasets. PKU-MMD part II, with its increased skeleton noise due to view variation, and NTU RGB+D 120, being the largest dataset with multiple classes, present more challenging test cases. Our proposed method performs well on these two datasets, demonstrating the effectiveness of our proposed technique.

TABLE 4.9: Comparisons of action recognition results with semi-supervised learning approaches on NTU RGB+D Cross-Subject (CS) Protocol. (C-F stands for coarse-fine; '3s' means three-stream fusion; * means the reproduced results with our labeled/unlabeled splitting; and the number in parentheses denotes the number of labeled samples per class)

Method	Backbone	Streams	Semi-1	Semi-5	Semi-10	Semi-20	Semi-40
			CS (7)	CS (33)	CS (66)	CS (132)	CS (264)
S^4L(Inpainting) [134] (ICCV 2019)	GRU	1	–	48.4	58.1	63.1	68.2
Pseudolabels [135] (ICML 2013)	GRU	1	–	50.9	57.2	62.4	68.0
VAT [136] (TPAMI 2018)	GRU	1	–	51.3	60.3	65.6	70.4
VAT + EntMin [137] (NeurIPS 2005)	GRU	1	–	51.7	61.4	65.9	70.8
ASSL [120] (ECCV 2020)	GRU	1	–	57.3	64.3	68.0	72.3
MS^2L [111] (ACMMM 2020)	GRU	1	33.1	–	65.2	–	–
LongT GAN [88] (AAAI 2018)	GRU	1	35.2	–	62.0	–	–
Skeleton-Contrastive [133] (ACMMM 2021)	GRU+CNN	4	35.7	59.6	65.9	70.8	–
AL+K [138] (arxiv 2020)	GRU	1	41.8	57.8	62.9	–	–
SKT [132] (ICME 2022)	GCN	1	43.2	–	67.6	–	–
MAC [139] (TPAMI 2022)	GCN	2	–	63.3	74.2	78.4	81.1
GL-Transformer [94] (ECCV 2022)	Transformer	1	–	64.5	68.6	–	–
3s-Colorization [119] (ICCV 2021)	DGCNN	3	48.3	65.7	71.7	76.4	79.8
3s-CrosSCLR [92]* (CVPR 2021)	GCN	3	49.9±1.30	66.5±0.50	73.3±0.43	77.0±0.33	81.0±0.30
3s-AimCLR [93]* (AAAI 2022)	GCN	3	45.5±1.72	**68.1±0.48**	76.1±0.43	77.8±0.38	81.5±0.51
3s-C-F Masked Colorization (Ours)	DGCNN	3	**52.3±0.57**	**68.1±0.19**	**76.5±0.27**	**78.7±0.27**	**82.5±0.28**

TABLE 4.10: Comparisons of action recognition results with semi-supervised learning approaches on NTU RGB+D Cross-View (CV) Protocol. (C-F stands for coarse-fine; '3s' means three-stream fusion; * means the reproduced results with our labeled/unlabeled splitting; and the number in parentheses denotes the number of labeled samples per class)

Method	Backbone	Streams	Semi-1	Semi-5	Semi-10	Semi-20	Semi-40
			CV (7)	CV (33)	CV (66)	CV (132)	CV (264)
S^4L(Inpainting) [134] (ICCV 2019)	GRU	1	–	55.1	63.6	71.1	76.9
Pseudolabels [135] (ICML 2013)	GRU	1	–	56.3	63.1	70.4	76.8
VAT [136] (TPAMI 2018)	GRU	1	–	57.9	66.3	72.6	78.6
VAT + EntMin [137] (NeurIPS 2005)	GRU	1	–	58.3	67.5	73.3	78.9
ASSL [120] (ECCV 2020)	GRU	1	–	63.6	69.8	74.7	80.0
Skeleton-Contrastive [133] (ACMMM 2021)	GRU+CNN	4	38.1	65.7	72.5	78.2	–
SKT [132] (ICME 2022)	GCN	1	44.9	–	71.3	–	–
GL-Transformer [94] (ECCV 2022)	Transformer	1	–	68.5	74.9	–	–
3s-Colorization [119] (ICCV 2021)	DGCNN	3	52.5	70.3	78.9	82.7	86.8
MAC [139] (TPAMI 2022)	GCN	2	–	70.4	78.5	84.6	89.6
3s-CrosSCLR [92]* (CVPR 2021)	GCN	3	52.0±0.48	69.5±1.55	77.4±0.30	82.9±0.14	87.3±0.20
3s-AimCLR [93]* (AAAI 2022)	GCN	3	46.9±2.08	71.8±0.80	80.7±0.71	84.0±0.29	88.6±0.28
3s-C-F Masked Colorization (Ours)	DGCNN	3	**53.1±0.97**	**74.2±1.05**	**81.3±0.30**	**85.7±0.46**	**89.7±0.10**

4.3.3.2 Semi-Supervised Learning

We evaluate semi-supervised learning with the same protocol as in [111, 120, 131, 138], for a fair compaison. Under the semi-supervised setting, the encoder $E(.)$ is first pre-trained with colorized skeleton clouds and then jointly trained with the linear classifier $f(.)$ with a small ratio of action annotations. Following [111, 120, 131, 138], we derive labeled data by uniformly sampling 1% (Semi-1), 5% (Semi-5), 10% (Semi-10), 20% (Semi-20), 40% (Semi-40) data from the training set of NTU

TABLE 4.11: Comparisons of action recognition results with semi-supervised learning approaches on NW-UCLA dataset. (C-F stands for coarse-fine; '2s' means two-stream fusion; *v./c.* denotes the number of labeled videos per class)

Method	Backbone	Stream	Semi-1	Semi-5	Semi-10	Semi-15	Semi-30	Semi-40
S^4L(Inpainting) [134] (ICCV 2019)	GRU	1	–	35.3	–	46.6	54.5	60.6
Pseudolabels [135] (ICML 2013)	GRU	1	–	35.6	–	48.9	60,6	65.7
VAT [136] (TPAMI 2018)	GRU	1	–	44.8	–	63.8	73.7	73.9
VAT + EntMin [137] (NeurIPS 2005)	GRU	1	–	46.8	–	66.2	75.4	75.6
ASSL [120] (ECCV 2020)	GRU	1	–	52.6	–	74.8	78.0	78.4
LongT GAN [88] (AAAI 2018)	GRU	1	18.3	–	59.9	–	–	–
MS^{2}L [111] (ACMMM 2020)	GRU	1	21.3	–	60.5	–	–	–
2s-Colorization [119] (ICCV 2021)	DGCNN	2	41.9	57.2	75.0	76.0	83.0	84.9
GL-Transformer [94] (ECCV 2022)	Transformer	1	–	58.5	74.3	–	–	–
AL+K [138] (arxiv 2020)	GRU	1	–	**63.9**	–	76.8	80.3	85.0
MAC [139] (TPAMI 2022)	GCN	2	–	63.0	–	**78.8**	79.9	81.6
2s-C-F Masked Colorization (Ours)	DGCNN	2	**42.8**	61.6	**76.7**	**78.8**	**84.8**	**86.6**

TABLE 4.12: Comparisons of action recognition results with semi-supervised learning approaches on NTU RGB+D 120 dataset C-Subject protocol. (C-F stands for coarse-fine; '3s' means three-stream fusion)

Method	Semi-1	Semi-5	Semi-10	Semi-20	Semi-40	Semi-50
AS-CAL[91] (Information Sciences 2021)		–	42.3	–	–	52.6
CP-STN [131] (ACML 2021)	**48.8**	51.5	56.1	60.6	66.5	67.8
3s-C-F Masked Colorization (Ours)	46.5	**55.0**	**62.2**	**67.3**	**72.3**	**75.4**

TABLE 4.13: Comparisons of action recognition results with semi-supervised learning approaches on UWA3D dataset. (C-F stands for coarse-fine; '2s' means two-stream fusion)

Method	Semi-5	Semi-10	Semi-20	Semi-50
AL [138] (arxiv 2020)	26.9	37.1	41.2	55.8
AL+K [138] (arxiv 2020)	28.3	36.0	51.8	59.5
2s-Colorization [119] (ICCV 2021)	32.8	40.9	52.2	61.5
2s-C-F Masked Colorization (Ours)	**36.5**	**44.2**	**54.5**	**63.8**

RGB+D dataset, 1% (Semi-1), 5% (Semi-5), 10% (Semi-10), 20% (Semi-20), 40% (Semi-40), 50% (Semi-50) data from the training set of NTU RGB+D 120 dataset, 1% (Semi-1), 10% (Semi-10) data from the training set of PKU-MMD dataset, 1% (Semi-1), 5% (Semi-5), 10% (Semi-10), 15% (Semi-15), 30% (Semi-30), 40% (Semi-40) data from the training set of NW-UCLA dataset, and 5% (Semi-5), 10% (Semi-10), 20% (Semi-20), 50% (Semi-50) data from the training set of UWA-3D dataset.

Tables 4.9, 4.10, 4.11, 4.12, 4.13 and 4.14 show experimental results for these

TABLE 4.14: Comparisons to state-of-the-art semi-supervised skeleton action recognition method on PKU-MMD dataset. (C-F stands for coarse-fine; '3s' means three-stream fusion)

Method	PKU-MMD part I		PKU-MMD part II	
	Semi1	Semi10	Semi1	Semi10
LongT GAN [88] (AAAI 2018)	35.8	69.5	12.4	25.7
MS^2L [111] (ACMMM 2021)	36.4	70.3	13.0	26.1
Skeleton-Contrastive [133] (ACMMM 2021)	37.7	72.1	–	–
3s-CrosSCLR [92] (CVPR 2021)	49.7	82.9	10.2	28.6
3s-AimCLR [93] (AAAI 2022)	57.5	86.1	15.1	33.4
3s-C-F Masked Colorization (Ours)	**57.8**	**86.5**	**15.5**	**34.2**

five datasets. It can be seen that most of the experimental results on all semi-settings of these five datasets achieve state-of-the-art performance. Even though the proposed framework does not outperform the state-of-the-art methods on four semi-settings (*i.e.,* semi-1 and semi-10 on NTU RGB+D C-Subject, semi-1 on NTU RGB+D 120 C-Subject, and semi-5 on NW-UCLA), we still can achieve competitive performance.

Additionally, for a much fairer comparison, we randomly select a specific percentage of training samples five times and subsequently perform five-fold experiments. These experiments involve our proposed methods and two state-of-the-art methods [92, 93], both of which provide pre-trained models on the NTU RGB+D dataset. The experimental results can be found at the bottom of Table 4.9 and Table 4.10, we can find that our proposed method consistently outperforms [92, 93] on the averaged results, further showing the effectiveness of our proposed methods.

4.3.3.3 Supervised Learning

Following the supervised evaluation protocol in [92, 93, 111], we pre-train the encoder with our self-supervised masked skeleton colorization method and fine-tune the encoder and classifier by using labeled training data. The experimental results are presented in Table 4.15. We find that the proposed coarse-fine masked skeleton colorization method outperforms the state-of-the-art self-supervised methods on all four datasets. Despite our framework not being specifically designed for a supervised setting, its performance is comparable to state-of-the-art supervised skeleton

TABLE 4.15: Comparisons to state-of-the-art Supervised and Unsupervised Pretrain skeleton action recognition methods on NTU RGB+D, NTU RGB+D 120, PKU-MMD, and NW-UCLA datasets.

Method	NTU RGB+D		NTU RGB+D 120		PKU-MMD		NW-UCLA
	C-Subject	C-View	C-Subject	C-Setup	I	II	
Supervised method							
Actionlet ensemble [140] (TPAMI 2013)	–	–	–	–	–	–	76.0
HBRNN-L [17] (CVPR 2015)	59.1	64.0	–	–	–	–	78.5
Part-Aware LSTM [18] (CVPR 2016)	62.9	70.3	25.5	26.3	–	–	–
ST-LSTM [15] (ECCV 2016)	69.2	77.7	55.7	57.9	–	–	–
Ensemble TS-LSTM [141] (ICCV 2017)	74.6	81.3	–	–	–	–	89.2
VA-RNN-Aug [142] (TPAMI 2019)	79.8	88.9	–	–	–	–	90.7
GCA-LSTM [52] (CVPR 2017)	74.4	82.8	–	–	–	–	–
ST-GCN [73] (AAAI 2018)	81.5	88.3	70.7	73.2	84.1	48.2	–
AS-GCN [143] (CVPR 2019)	86.8	94.2	–	–	–	–	–
2s-AGCN [144] (CVPR 2019)	88.5	95.1	82.5	84.2	93.5	56.8	–
2s-AGC-LSTM [145] (CVPR 2019)	89.2	95.0	–	–	–	–	93.3
4s-Shift-GCN [75] (CVPR 2020)	90.7	96.5	85.9	87.6	–	–	94.6
PA-ResGCN-B19 [146] (ACMMM 2020)	90.9	96.0	87.3	88.3	–	–	–
MS-G3D [147] (CVPR 2020)	91.5	96.2	86.9	88.4	**95.0**	**59.2**	–
CTR-GCN [76] (ICCV 2021)	92.4	96.8	88.9	90.6	–	–	96.5
Info-GCN [77] (CVPR 2022)	**93.0**	**97.1**	**89.8**	**91.2**	–	–	**97.0**
Unsupervised Pretrain method							
Li *et al.* [87] (NeurIPS 2018)	63.9	68.1	–	–	–	–	62.5
MS^2L [111] (ACMMM 2020)	78.6	–	–	–	85.2	45.7	86.8
SKT [132] (ICME 2022)	83.1	91.2	–	–	–	–	–
3s-CrosSCLR [92] (CVPR 2021)	86.2	92.5	80.5	80.4	–	–	–
3s-AimCLR [93] (AAAI 2022)	86.9	92.8	80.1	80.9	–	–	–
2s-Colorization [119] (ICCV 2021)	–	–	–	–	–	–	94.0
3s-Colorization [119] (ICCV 2021)	88.0	94.9	78.5	79.3	91.5	54.0	–
2s-C-F Masked Colorization (Ours)	–	–	–	–	–	–	**95.0**
3s-C-F Masked Colorization (Ours)	**89.1**	**95.9**	**81.2**	**82.4**	**93.3**	**57.7**	–

action recognition methods on NTU RGB+D, NTU RGB+D 120, PKU-MMD, and NW-UCLA, and UWA3D datasets.

4.3.3.4 Transfer Learning

To further evaluate whether the proposed skeleton colorization method is able to gain knowledge to related tasks, we investigate the transfer learning performance of our model.

Transfer to PKU MMD II: Following the setting in [111], we conduct transfer learning experiments, which use NTU RGB+D and PKU-MMD I as the source datasets and PKU-MMD II as the target dataset. Initially, we train the model using either the NTU RGB+D or PKU-MMD I Cross-Subject protocol. The pretrained model is subsequently fine-tuned on the PKU-MMD II dataset. Table 4.16

TABLE 4.16: Comparison of the transfer learning performance on PKUMMD part II dataset. (C-F stands for coarse-fine; '3s' means three-stream fusion)

Method	Transfer to PKU-MMD II	
	PKU-MMD I	NTU RGB+D 60
LongT GAN [88] (AAAI 2018)	43.6	44.8
MS^{2}L [111] (ACMMMM 2020)	44.1	45.8
Skeleton-Contrastive [133] (ACMMM 2020)	45.1	45.9
3s-Colorization [119] (ICCV 2021)	50.0	51.0
3s-C-F Masked Colorization (Ours)	**58.0**	**58.1**

TABLE 4.17: Comparison of the transfer learning performance on the Toyota Smarthome dataset. (C-F stands for coarse-fine; '2s' means two-stream fusion)

Methods	Smarthome	
	CS	CV1
UNIK [148] (BMVC 2021)	63.1	22.9
ViA [149] (IJCV 2024)	64.5	36.1
2s-Colorization [119] (ICCV 2021)	70.4	42.1
2s-C-F Masked Colorization (Ours)	**72.6**	**44.7**

shows the transfer learning results. We can find that our proposed method achieves much better results than [88, 111, 133], showing that the feature extracted by our proposed method from a source dataset can improve classification accuracy on a different target set.

Transfer to Smarthome [123]: We also conduct transfer learning experiments that use the NTU RGB+D as the source dataset and Smarthome as the target dataset. The experiments are conducted on the cross-subject (CS) and cross-view1 (CV1) evaluation protocols. The experimental results are shown in Table 4.17. It can be seen that our proposed method achieves state-of-the-art performance, demonstrating the effectiveness of our proposed method when transferred to related datasets.

It should be noted that the Toyota Smarthome skeleton data consists of only 13 skeleton joints. To ensure the input data from Toyota Smarthome matches the size of the NTU RGB+D dataset, we applied interpolation and zero padding.

4.4 Summary

In this work, we tackle the challenge of self-supervised representation learning for skeleton action recognition by designing a novel coarse-fine masked skeleton cloud colorization method that is capable of learning skeleton representations from unlabeled data. Specifically, we obtain colored skeleton cloud representations by stacking skeleton sequences to 3D skeleton clouds and colorizing each point according to its temporal and spatial orders in the skeleton sequences. Besides, we introduce a two-stream pretraining framework that leverages coarse-grained colorization and fine-grained colorization to learn multi-scale spatial-temporal features. Additionally, we introduce a Masked Skeleton Cloud Repainting task to pretrain the designed auto-encoder framework. Extensive experiments over different datasets demonstrate that our proposed method achieves superior unsupervised and semi-supervised action recognition performance.

Chapter 5

One-Shot Action Recognition via Multi-Scale Spatial-Temporal Skeleton Matching

In Chapters 3 and 4, we explored weakly-supervised and self-supervised learning methods, tackling the issue of scarce labeling in skeleton action recognition. These approaches typically evaluate performance when the training and testing datasets consist of the same action categories. This chapter shifts our focus to a more complex objective: one-shot skeleton action recognition. Here, we deal with the unique challenge of recognizing actions from new, unseen classes with only a single example provided.

5.1 Introduction

Human action recognition is a fast-developing research area due to its diverse applications in human-computer interaction, video surveillance, game control, etc. In recent years, human action recognition with skeleton data has attracted increasing attention as skeleton data encodes high-level representations of human actions

and is generally lightweight and robust to variations in appearances, surrounding distractions, viewpoint changes, etc. As of today, most existing studies studies predominantly rely on extensive labeled training data for learning effective human action representations. While facing skeleton data of a new category, these studies necessitate the collection of hundreds of action samples from the new category to adapt or fine-tune existing models. How to achieve single-shot recognition for new action categories becomes critically important for circumventing the tedious and laborious data collection and labeling procedure.

One-shot skeleton action recognition is an extremely challenging task. Not only is it constrained by the one-shot data of unseen new classes, but it is also complicated by the considerable variations in human actions. Take the action "put on glasses" as an example. Different people could perform it by using their left hand, right hand, or both hands. The same person could also perform it at different paces with different motion dynamics. Different approaches have been explored to address this challenging task, and most existing work [98–101] represents the anchor and target samples with certain pooled feature vectors and computes the sample distance based on the similarity of their pooled feature vectors. However, the adoption of such global feature similarity discards the very useful spatial structures and temporal order of the skeleton sequences. In addition, most existing work learns skeleton action representations at a single scale of the original body joints, which tends to lose useful action features under the one-shot scenario. Drawing from [147, 150–153], it is evident that human actions are multi-scales in both spatial and temporal spaces. For instance, multiple joints on arms and legs collaborate in walking, and consecutive frames of human actions contain strong temporal correlations. Skeleton action representations should therefore capture the rich semantic correlations at different spatial and temporal scales. In contrast to [147, 150–153], which employ multi-scale information for recognition and prediction, we leverage the extracted multi-scale features for match and design the innovative matching strategies specifically for one-shot skeleton action recognition challenge.

We propose to capture spatial-temporal features by leveraging spatial structure and

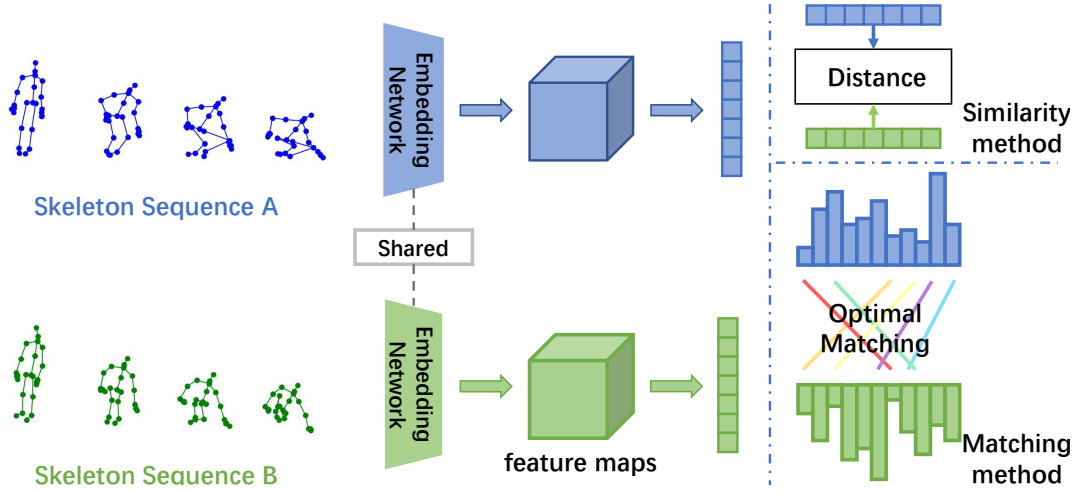


FIGURE 5.1: Skeleton action recognition based on feature similarity or feature matching: Feature similarity computes the distance between feature vectors which discards the very useful spatial skeleton structures and temporal information. The proposed feature matching compares two skeleton sequences by computing a matching flow between their feature distributions which can capture useful spatial and temporal information effectively. The colored line emphasizes channels paired based on their high matching scores.

temporal order inherent in skeleton sequences, as illustrated in Fig. 5.1. Inspired by the theory of optimal transport [154, 155], we measure the semantic relevance of two skeleton sequences by computing an optimal matching flow between their feature maps. Specifically, we adopt Earth Mover's Distance (EMD) [156] as the optimal matching metric for acquiring the optimal matching flow. EMD is the metric for computing the distance between two representations, enabling us to determine the similarity between the feature representations of two skeleton samples. In our one-shot skeleton action recognition scenario, given the distances between all skeleton joint pairs, EMD maximizes the impact of relevant joints and minimizes the effect of irrelevant joints between two skeleton sequences. In addition, we model skeleton sequences across multiple spatial scales (joint-scale, part-scale, and limb-scale) and temporal scales as illustrated in Fig. 5.2 . We conduct multi-scale matching using EMD to capture the scale-wise semantic relevance of the skeleton. It is also important to note that human actions can be performed at different motion magnitudes and paces, e.g., 'hand waving' may be performed by hand (joint-level), forearm (part-level), or the whole arm (limb-level) at different paces. Therefore,

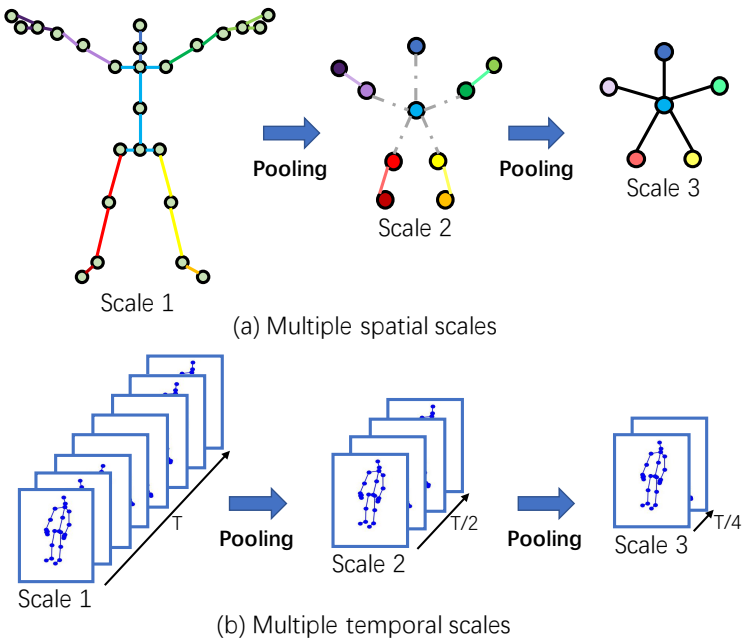


FIGURE 5.2: The proposed multi-scale skeleton modeling at spatial dimension in (a) and temporal dimension in (b): Given the original spatial scale at *Scale 1* , we first divide the skeleton nodes into multiple groups with similar semantic skeleton structures and then perform average pooling to each group to generate skeleton graphs of coarser scales (i.e., *Scale 2* and *Scale 3*.) The nodes whose links are of the same color belong to the same group with similar semantics. Along the temporal dimension, we perform average pooling over features of adjacent frames to obtain temporal features of coarser scales at *Scale 2* and *Scale 3*.

we design cross-scale matching that learns semantic relevance by measuring feature consistency across various spatial and temporal scales.

The contributions of this work are summarized as follows:

- We formulate one-shot skeleton action recognition as an optimal matching problem and design an effective network framework for one-shot skeleton action recognition.
- We propose a multi-scale matching strategy that can capture scale-wise skeleton semantic relevance at multiple spatial and temporal scales. On top of that, we design a novel cross-scale matching scheme that can model the within-class variation of human actions in motion magnitudes and motion paces. To the best of our knowledge, this is the first work that exploits

multi-scale representations and cross-scale matching to capture multi-scale skeleton semantic relevance and maintain consistency across motion scales in one-shot skeleton action recognition.

- Extensive experiments on three public datasets (NTU RGB+D, NTU RGB+D 120, and PKU-MMD) show that our method outperforms the state-of-the-art consistently by large margins.

5.2 Method

Our objective is to train a model capable of recognizing human skeleton data of novel classes with only a single labeled sample. It is a very challenging task due to the very rich intra-class spatial-temporal variations in human skeleton action. To address this, we propose a one-shot skeleton action recognition framework, as illustrated in Fig. 5.3. Specifically, we design a novel optimal matching technique that captures the useful spatial structure and temporal order information, which is largely neglected in most existing one-shot skeleton action recognition studies.

In the following, we first present the problem formulation of the one-shot skeleton action recognition task. Then, we introduce the embedding network and elaborate on the process of constructing multi-spatial and multi-temporal scale skeletons. Finally, we provide a detailed description of our proposed optimal matching technique.

5.2.1 Problem Formulation

Inspired by prior studies in few-shot learning [35, 157–160] and few-shot video action recognition [161–166], we formulate the one-shot skeleton action recognition task as a meta-learning problem [37] that consists of a meta-training phase and a meta-testing phase. In an n-way and 1-shot problem, each episode consists of a support set S and a query set Q, where S contains 1 labeled sample for each of n

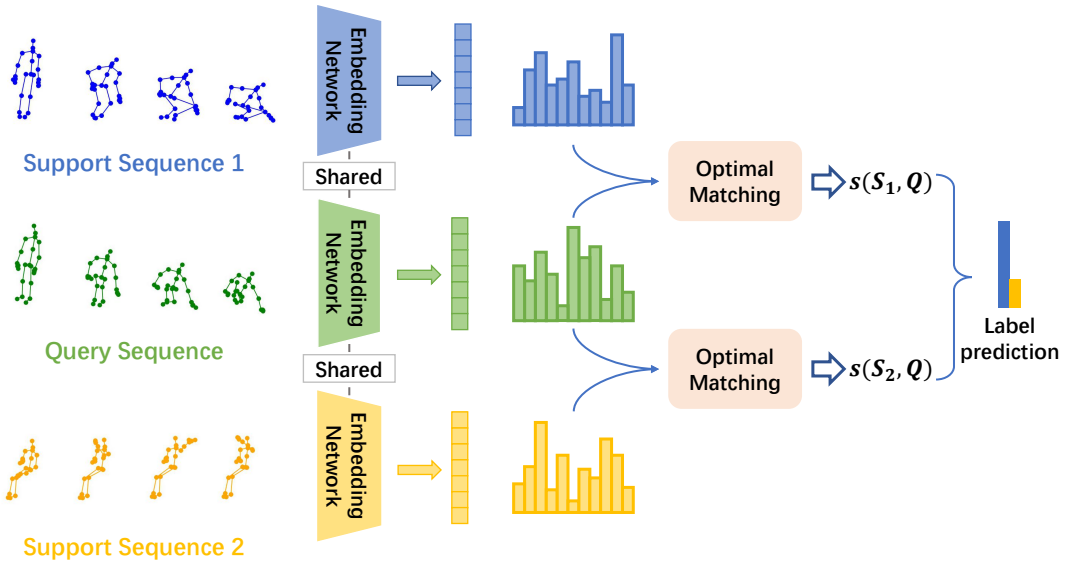


FIGURE 5.3: The pipeline of the proposed method: Given a *Query Sequence* and two support skeleton sequences *Support Sequence 1* and *Support Sequence 2*, skeleton representations are first extracted with a weight-sharing embedding network which are further aligned progressively with the proposed *Optimal Matching*. The semantic relevance score (denoted by $s(\cdot, \cdot)$) between the query and the two support instances can then be computed for action prediction. The pipeline is illustrated with a 2-way 1-shot task. Different *Optimal Matching* strategies are provided in Fig. 5.4.

unseen classes and Q is employed to evaluate the generalization performance. The algorithm's objective is to assign each query sample to the appropriate support classes. Specifically, multiple n-way and 1-shot tasks are randomly sampled from the *meta-training set* D_{train} (with seen classes), and employed to train a model in an episodic manner. In *meta-testing phase*, n-way and 1-shot tasks are sampled from the *meta-testing set* D_{test} (with unseen classes) for evaluation.

5.2.2 Skeleton Feature Embedding

Following prior studies on few-shot learning [35, 157–160] and few-shot video action recognition [161–166], we first pre-train an embedding network on the whole *meta-training set* D_{train} using the cross-entropy loss for standard classification before proceeding to episodic training. We adopt the GCN-based model [16] as the baseline network which has an adaptive spatial-temporal graph for extracting the

relation among body joints. The GCN-based model, however, only processes the features of the original scale. However, such single-scale modeling often misses meaningful skeleton information, especially when only a single labeled sample is available as described in Section 5.1. Inspired by [147, 150–153] which handle multi-scale features, we represent human skeleton data at multi-spatial and multi-temporal scales.

Multi-Spatial Scale Skeleton: We model skeleton actions at multiple spatial scales. Specifically, we employ three distinct spatial scales including the body-joint scale (s_1), the part-level scale (s_2), and the limb-level scale (s_3), as illustrated in Fig. 5.2 (a). We first build GCN blocks on the first scale to capture joint-wise feature representations and then perform average pooling. For skeleton-based representations, the pooling requires meaningful neighborhoods and we simply put joints of the same spatial scale into one group.

Multi-Temporal Scale Skeleton: Recognizing that consecutive frames capture continuous motions and poses reflecting analogous abstract states, we represent skeleton data across multiple temporal scales. After processing several GCN blocks at the original temporal scale, we incorporate two average pooling layers along the temporal dimension to perform temporal pooling as illustrated in Fig. 5.2 (b). Specifically, by applying average pooling to the features of consecutive frames from the original scale, we consolidate them into a unified feature to represent a 'new frame' in coarser scales, such as scale 2 or scale 3 As we apply average pooling with strides of 2 and 4, the features of two consecutive frames are pooled to produce the scale 2 features, while those of four consecutive frames are pooled to yield the scale 3 features.

We implement 3 spatial scales and 3 temporal scales for illustration, where each skeleton structure captures unique perspectives of skeleton representations. To extract the multi-scale skeleton representation, each stream within the multi-spatial and multi-temporal scale skeleton structures is individually optimized through the cross-entropy loss.

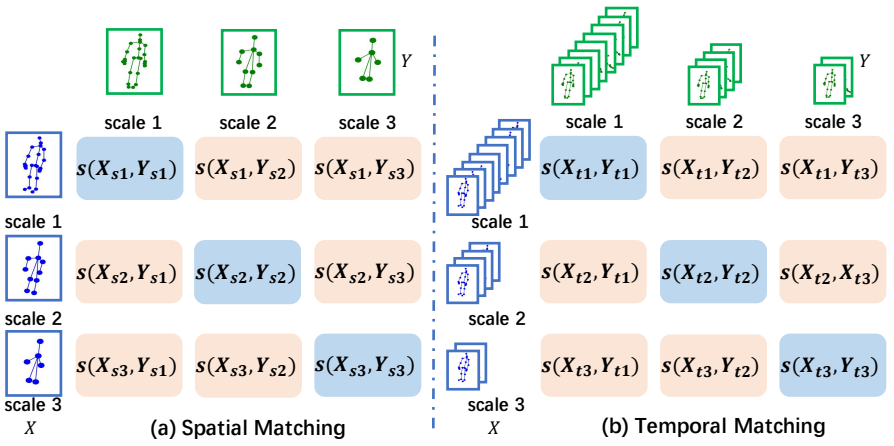


FIGURE 5.4: Illustration of optimal matching in Spatial Matching in (a) and Temporal Matching in (b): In each of the two sub-figures, the three matching strategies along the diagonal (in blue color) illustrate the proposed multi-scale matching, and the rest 6 of the diagonal (in orange color) show the proposed cross-scale matching. Here, $s(\cdot, \cdot)$ represents the semantic relevance score between two skeleton features. X and Y stand for two skeleton sequences.

5.2.3 Optimal Matching Strategy

Unlike [98–101] that compute distances over sequence-level embeddings, we strive to capture discriminative local information of each body joint and design a skeleton optimal matching scheme to compute semantic relevance based on optimal transport theory as discussed in Section 5.1. Specifically, we adopt the Earth Mover's Distance (EMD) [156] as the optimal transport matching metric. This approach aims to search for the minimal cost transport plan between two joints' feature distributions by maximizing the influence caused by relevant joints and minimizing the impact between irrelevant joints.

Here, we first utilize the single-scale model as an illustrative example to show how we formulate the one-shot skeleton action recognition as the optimal matching problem by employing the Earth Mover's Distance (EMD). The skeleton representation embedded by the single-scale model can be represented as $\mathbf{X} \in R^{C \times N \times T}$, where C is the number of output channels, N denotes the number of skeleton joints, and T denotes the number of frames. Given two feature maps $\mathbf{X}, \mathbf{Y} \in R^{C \times N \times T}$, we first flatten them into two sets of local joint representations $\mathcal{X} = \{x_i | i = 1, 2, ...NT\}$

and $\mathcal{Y} = \{y_j \mid j = 1, 2, \ldots NT\}$, where x_i and y_j ($x_i, y_j \in R^C$) denote the local joint representations at the corresponding spatial and temporal positions, respectively. Then we define the EMD between two sets of local representations as the minimum "transport cost" from suppliers ($\mathcal{X}$) to demanders ($\mathcal{Y}$). Assuming each supplier x_i possesses r_i units for transport and each demander y_j requires c_j units, the overall optimal transport matching problem can be encapsulated as:

$$OT(r, c) = \left\{\pi \in \mathcal{R}^{NT \times NT} | \pi \mathbb{1} = \mathbf{r}, \pi^T \mathbb{1} = \mathbf{c}\right\}, \tag{5.1}$$

where π is the optimal matching flow between these two distributions, which can also be viewed as the optimal matching plan of two skeleton sequences. r_i and c_j are called the weights of nodes, which control the total matching flows generated by each node, and $\mathbf{r}$ and $\mathbf{c}$ are vectorized representations of $\{r_i\}$ and $\{c_j\}$. EMD seeks an optimal matching flow π between "suppliers" $\mathcal{X}$ and "demanders" $\mathcal{Y}$, such that the overall matching cost can be minimized.

Additionally, the transporting cost per unit is determined by calculating the pairwise distance between supplier node $\mathbf{x}_i$ and demander node $\mathbf{y}_j$ from two skeleton features:

$$d_{ij} = 1 - \frac{\mathbf{x}_i^T \mathbf{y}_j}{\|\mathbf{x}_i\| \, \|\mathbf{y}_j\|}, \tag{5.2}$$

where nodes with similar local representations tend to generate small transporting costs between each other. Then we can define the EMD as the optimal transportation problem, which is represented as:

$$D_{emd}(\mathbf{X}, \mathbf{Y}) = \min_{\pi \in OT(r,c)} \sum_{i=1}^{NT} \sum_{j=1}^{NT} d_{ij} \pi_{ij}. \tag{5.3}$$

The weight of each node (*e.g.*, r_i and c_j) plays a significant role in optimal matching problems. Intuitively, the node with a larger weight is more important when matching two sets. Therefore, in order to assign the more important node a higher weight, we follow [160] to generate weight r_i by a cross-reference mechanism that

employs the dot product between a joint representation and the average joint representation within the other skeleton features:

$$r_i = max\left\{x_i^T \cdot \frac{\sum_{j=1}^{NT} y_j}{NT}, 0\right\}, \tag{5.4}$$

where x_i and y_j denote the feature vectors from two skeleton feature maps, and function $max(\cdot)$ ensures the weight is always non-negative. Above we take r_i as an example, and c_j can be calculated in the same manner. Upon acquiring the optimal matching flow π, the semantic relevance score s between two skeleton representations can be calculated as follows:

$$s(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{NT}\sum_{j=1}^{NT}(1 - d_{ij})\pi_{ij}. \tag{5.5}$$

These semantic relevance scores allow studying the composition of the overall relevance, enabling us to assign high relevance to semantically similar joints no matter whether they are in the same spatial order or temporal frame. Consequently, we can tackle the problems that the semantic relevance of two skeleton sequences can occur at different temporal positions or different spatial joints.

Multi-Scale Matching. As mentioned in Section 5.2.2, human skeleton data can be represented in multi-spatial scales and multi-temporal scales, and each scale's representation contains unique semantic information. Therefore, we propose to capture the pair-wise skeleton semantic relevance at multiple scales, including multi-spatial scale matching and multi-temporal scale matching, to acquire an optimal matching flow from multiple spatial and multiple temporal scales.

For the multi-spatial scale scenario, there are three pairs of feature embeddings, which can be represented as $\mathbf{X}_{s1} \in R^{C\times N\times T}$, $\mathbf{X}_{s2} \in R^{C\times N_2\times T}$, and $\mathbf{X}_{s3} \in R^{C\times N_3\times T}$, respectively. Here, N_2 denotes the number of nodes for the second-scale spatial graph, and N_3 stands for the number of third-scale spatial graph nodes. The

semantic relevance score between two skeleton sequences is defined as:

$$s_{ms}(\mathbf{X}, \mathbf{Y}) = s(\mathbf{X}_{s1}, \mathbf{Y}_{s1}) + s(\mathbf{X}_{s2}, \mathbf{Y}_{s2}) + s(\mathbf{X}_{s3}, \mathbf{Y}_{s3}). \tag{5.6}$$

This enables us to seek the optimal matching flow using EMD, and measure the semantic relevance at multiple spatial scales as shown in Fig. 5.4 (a) (highlighted in blue color).

For the multi-temporal scale, there are also three pairs of feature embeddings, which can be represented as $\mathbf{X}_{t1} \in R^{C\times N\times T}$, $\mathbf{X}_{t2} \in R^{C\times N\times T/2}$, and $\mathbf{X}_{t3} \in R^{C\times N\times T/4}$, respectively. The semantic relevance score between two skeleton sequences becomes:

$$s_{mt}(\mathbf{X}, \mathbf{Y}) = s(\mathbf{X}_{t1}, \mathbf{Y}_{t1}) + s(\mathbf{X}_{t2}, \mathbf{Y}_{t2}) + s(\mathbf{X}_{t3}, \mathbf{Y}_{t3}). \tag{5.7}$$

With the application of Eq. (5.7), the semantic relevance between two skeleton sequences is measured at multiple temporal scales, as shown in Fig. 5.4 (b) (highlighted in blue color).

Cross-Scale Matching. As discussed in Section 5.1, different instances of the same action class may be performed at different magnitudes (spatial scales). For instance, consider the action class 'hand waving'; it can be performed by merely moving the palm (joint-level), by moving the forearm (part-level), or by mobilizing the entire arm (limb-level). Additionally, the same-category samples can also be performed at different speeds (temporal scale). Thus, there is also semantic relevance between different scales' skeleton representations for matching. To address this cross-scale matching challenge, we further investigate how to measure the semantic relevance between skeleton sequences across different scales, including cross-spatial scale and cross-temporal scale matching, considering the possibility of different spatial magnitudes and temporal speeds of the same action.

For cross-spatial scale matching, the three spatial scales' skeleton representations are $\mathbf{X}_{s1} \in R^{C\times N\times T}$, $\mathbf{X}_{s2} \in R^{C\times N_2\times T}$, and $\mathbf{X}_{s3} \in R^{C\times N_3\times T}$. It can be seen that all these three scales' representations contain T frame features. Thus, we first

perform 1D average pooling ($AvgPool$) on the spatial dimension to generate these three scales' features in the same shape ($R^{C\times T}$). Next, we formulate the semantic relevance score as an optimal matching problem over T frame features, to obtain the optimal matching flow between different spatial-scale representations The cross-spatial scale semantic relevance score can be represented as:

$$s_{cs}(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{3}\sum_{j=1,j\neq i}^{3} s(AvgPool(\mathbf{X}_{si}), AvgPool(\mathbf{Y}_{sj})). \tag{5.8}$$

This process is shown in Fig. 5.4 (a) (highlighted in orange color). By following this approach, we address the challenge of matching skeleton sequences that exhibit varying degrees of motion magnitude

In addition to this, we handle the challenge of matching skeleton sequences that exhibit varying speeds of motion by designing cross-temporal scale matching. The three temporal scales' skeleton features are represented as $\mathbf{X}_{t1} \in R^{C\times N\times T}$, $\mathbf{X}_{t2} \in R^{C\times N\times T/2}$, and $\mathbf{X}_{t3} \in R^{C\times N\times T/4}$. It can be seen that these three scales' representations all contain N joint features. Similarly, the 1D average pooling ($AvgPool$) can be performed on the temporal dimension to pool $\mathbf{X}_{t1}$, $\mathbf{X}_{t2}$, and $\mathbf{X}_{t3}$ into the same shape ($R^{C\times N}$). Consequently, we apply the Earth Mover's Distance to measure the semantic relevance score across different temporal scales, which can be expressed as

$$s_{ct}(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{3}\sum_{j=1,j\neq i}^{3} s(AvgPool(\mathbf{X}_{ti}), AvgPool(\mathbf{Y}_{tj})), \tag{5.9}$$

This process is shown in Fig. 5.4 (b) (highlighted in orange color). By applying Eq. (5.9), we can effectively match action sequences that exhibit different motion speeds.

Summary. As mentioned above, we first introduce the single-scale semantic relevance score $s(\mathbf{X},\mathbf{Y})$ (Eq. (5.5)) that considers the useful spatial structure and temporal order information during matching. To address the problem that different instances of the same action class samples may be performed at different paces with

different motion dynamics, we introduce 4 types of semantic relevance scores including multi-spatial scale ($s_{ms}(\mathbf{X}, \mathbf{Y})$, Eq. (5.6)), multi-temporal scale ($s_{mt}(\mathbf{X}, \mathbf{Y})$, Eq. (5.7)), cross-spatial scale ($s_{cs}(\mathbf{X}, \mathbf{Y})$, Eq. (5.8)), and cross-temporal scale ($s_{ct}(\mathbf{X}, \mathbf{Y})$, Eq. (5.9)). Our proposed model averages these multi-scale and cross-scale relevance scores to generate a final semantic relevance score, which is subsequently utilized to predict the action category.

5.2.4 Objective Loss, Model Training, and Inference

The majority of current few-shot learning techniques [35, 157–160] implement a pre-training stage prior to meta-learning. The effectiveness of this stage within the realm of few-shot learning has been validated by [167, 168]. Therefore, the proposed method is trained in two sequential stages: The First is the pre-training stage. The embedding network is trained on the *meta-training set* D_{train} in a standard supervised learning way (Section 5.2.2). The embedding network and our optimal matching method (Section 5.2.3) are further optimized in an end-to-end manner, following [156, 169]. Both training stages leverage the Softmax cross-entropy loss as the classification loss L_{cls} for optimization. Given an unseen query sequence q and its support set S (both sampled from *meta-testing set* D_{test}) at test, the goal is to determine which support set classes q belongs to.

5.3 Experiments

5.3.1 Datasets

We conduct experiments on three datasets: **NTU RGB+D** [18], **NTU RGB+D 120** [98], and **PKU-MMD** [121]. Detailed descriptions of these datasets are provided in Section 4.3.1. It is worth noting that the PKU-MMD consists of two subsets, part I and part II. In this work, we conduct experiments on the part I subset.

5.3.2 Training and Evaluation Protocol

Training Protocol. As described in Section 5.2.1, we formulate the one-shot skeleton action recognition problem as a meta-learning problem [35, 37, 160, 162, 163]. Therefore, we adopt the meta-training phase as our training protocol. All experiments are trained under the *5*-way, *1*-shot setting, and we set the number of query samples for each class to 15.

Evaluation Protocol 1. Similarly, we conduct the evaluation on *5*-way, *1*-shot setting, and adopt this procedure as Evaluation Protocol 1 in the following experiments.

Evaluation Protocol 2. To compare with the existing one-shot skeleton action recognition techniques [98–101, 103, 104] in a fair way, we also follow the official one-shot protocol described in [98] for the dataset NTU RGB+D 120. Specifically, the testing set consists of 20 novel classes, and we pick one sample from each novel class as the exemplar and leave the rest (except for the 20 exemplars) to test the recognition performance. For datasets NTU RGB+D and PKU-MMD, we adopt a similar protocol in experiments.

5.3.3 Implementation Details

Pre-training Stage. We use the SGD optimizer with Nesterov momentum (0.9) as the optimizer. The learning rate is set as 0.1 and is divided by 10 at the 30^{th} epoch and the 40^{th} epoch. The training process is ended at the 50^{th} epoch. For NTU RGB+D and NTU RGB+D 120, the batch size is set as 64. The batch size for the PKU-MMD dataset is 32.

Meta-learning Stage. The learning rate starts at 0.001 and decays every 10 epochs by 0.5. We train for 100 epochs using the SGD optimizer, and each epoch consists of 100 episodes from the *meta-training set* D_{train}. For the meta-testing phase, we sample 1000 episodes from *meta-testing set* D_{test}. (D_{train} and D_{test} are defined in Section 5.2.1).

Embedding Network. We use Adaptive Graph Convolutional Network [16] (AGCN) as our single-scale embedding network which has 9 GCN blocks. For multi-spatial and multi-temporal networks, the first 6 blocks are shared to capture the single-scale features. Then each scale feature is processed by the other 3 individually and in parallel.

We implement our method with the PyTorch framework. All experiments are conducted on Nvidia 2080Ti GPUs with CUDA 11.

5.3.4 Dataset Splitting

For dataset NTU RGB+D 120, we adopt the one-shot skeleton action setting as described in [98] which splits the full dataset into a training set and a testing set. The action classes of the two sets are distinct which include 100 classes for training and 20 for testing. For dataset NTU RGB+D, the training set and testing set are determined by the selection of 50 classes and 10 classes from the 100 training and 20 testing classes of the NTU RGB+D 120, respectively. In a similar fashion, for dataset PKU-MMD, we divide the action categories into a training set and a testing set which include 41 classes for training and 10 classes for testing.

As no hold-out validation set is defined in the one-shot skeleton action setting, and all three datasets contain the cross-subject setting for supervised action recognition. Therefore, we divide the training class data into the training set and validation set based on the cross-subject principle for the one-shot skeleton action recognition task. The original testing sets for these three datasets are retained as the testing sets for one-shot skeleton action recognition.

5.3.5 Evaluating on One-Shot Skeleton Action Recognition

We conduct extensive experiments with five optimal matching strategies that include single-scale ('**S-scale**'), multi-spatial scale ('$\boldsymbol{M_s}$'), multi-temporal scale ('$\boldsymbol{M_t}$'),

TABLE 5.1: One-shot skeleton recognition experiment under the Evaluation Protocol 1. (S-scale: single-scale matching; M-scale: multi-scale matching; M&C-scale: multi-scale and cross-scale matching)

Method	NTU	NTU 120	PKU-MMD
ProtoNet [35]	78.3	80.3	84.7
FEAT [159]	77.8	80.0	83.8
Subspace [157]	77.9	80.5	84.2
Dynamic Filter [158]	79.3	80.4	84.9
S-scale (Ours)	80.4	81.2	85.7
M-scale (Ours)	82.6	83.5	88.2
M&C-scale (Ours)	**83.7**	**84.5**	**89.3**

cross-spatial scale ('$\boldsymbol{C_s}$'), and cross-temporal scale ('$\boldsymbol{C_t}$'). We combine '$\boldsymbol{M_s}$' and '$\boldsymbol{M_t}$' to form a new multi-scale strategy '**M-scale**'. In addition, we combine '$\boldsymbol{M_s}$', '$\boldsymbol{M_t}$', '$\boldsymbol{C_s}$', and '$\boldsymbol{C_t}$' to form another new strategy '**M&C-scale**' that matches spatial and temporal features simultaneously at multiple scales and also cross scales.

We compare our method with two groups of state-of-the-art methods on one-shot skeleton action recognition. The first group consists of state-of-the-art few-shot image classification methods including Subspace [157], ProtoNet [35], Dynamic Filter [158], FEAT [159]. All these methods use the same embedding network as our method for fair comparisons. We re-implement [35, 157–159] based on publicly available codes and conduct experiments on the NTU RGB+D, NTU RGB+D 120, and PKU-MMD datasets. The second group consists of state-of-the-art one-shot skeleton action recognition techniques including APSR [98], TCN [99], SL-DML [100], Skeleton-DML [101], uDTW [104], and JEANIE [103]. For those one-shot skeleton action recognition works [98–101, 103, 104], the results in Tabs. 5.1, 5.2, and 5.4 are from the original papers. We compare our method with the first group methods under both evaluation protocols, while the second group with Evaluation Protocol 2 only.

Evaluation Protocol 1: We evaluate the *5*-way *1*-shot setting on all three datasets and Table 5.1 shows experimental results. It can be seen that our proposed single-scale optimal matching outperforms state-of-the-art few-shot learning methods on all three datasets. In addition, our proposed multi-scale and cross-scale

TABLE 5.2: One-shot skeleton recognition experiments under the Evaluation Protocol 2. (S-scale: single-scale matching; M-scale: multi-scale matching; M&C-scale: multi-scale and cross-scale matching)

Method	NTU	NTU 120	PKU-MMD
Attention Network [52]	–	41.0	–
Fully Connected [52]	–	42.1	–
Average Pooling [3]	–	42.9	–
APSR [98]	–	45.3	–
TCN [99]	–	46.5	–
SL-DML [100]	–	50.9	–
Skeleton-DML [101]	–	54.2	–
uDTW [104]	72.4	49.0	–
JEANIE [103]	80.0	57.0	–
ProtoNet [35]	74.8	60.4	78.1
FEAT [159]	74.3	61.5	75.9
Subspace [157]	75.6	60.9	75.6
Dynamic Filter [158]	75.9	60.6	78.8
S-scale (Ours)	77.4	63.2	82.8
M-scale (Ours)	81.6	67.6	85.7
M&C-scale (Ours)	**82.7**	**68.7**	**86.9**

matching strategies further improve one-shot skeleton action recognition by large margins, demonstrating the effectiveness of our proposed method on the one-shot skeleton action recognition task.

Evaluation Protocol 2: Following the one-shot setting in [98], we also conduct experiments on NTU RGB+D, NTU RGB+D 120, and PKU-MMD datasets under Evaluation Protocol 2. Similar to the experiments on Evaluation Protocol 1, our proposed method outperforms the state-of-the-art one-shot skeleton action recognition and few-shot learning methods by large margins (up to 8% on NTU120 and PKU-MMD). Table 5.2 shows more details of the experiments.

5.3.6 Ablation Studies

Matching Strategies: We draw comparisons between our proposed optimal matching method and the *global matching scheme* that adopts global average pooling to generate feature vector, as well as *a local matching scheme* that computes

TABLE 5.3: Comparison of different feature embedding approaches and distance metrics under Evaluation Protocol 2. (EMD: earth mover's distance)

Embedding	Metric	NTU	NTU 120	PKU-MMD
Global	Euclidean	71.6	56.1	78.1
Global	Cosine	74.8	60.4	74.8
Local	Euclidean	74.7	58.0	80.7
Local	Cosine	75.8	61.3	80.8
Local	EMD	**77.4**	**63.2**	**82.8**

joint-to-joint distances (either Euclidean or Cosine distance) with local-level representations. For fair comparisons, we adopt the same backbone and training scheme for all compared methods, and Table 5.3 presents the results of these experiments. We can observe that models with local-level representations perform better than models that rely on the global-level representations (in globally pooled feature vectors). In addition, our method which works with the optimal matching flow between all pairs of joints outperforms all compared methods. It should be noted that all these experiments were carried out on the single-scale model following evaluation protocol 2.

Reducing Training Classes: While evaluating one-shot action recognition methods, one interesting question is how many training classes are required to achieve fair recognition performance. We examine this issue under Evaluation Protocol 2 by following prior studies on NTU RGB+D 120 dataset [98, 100, 101, 103, 104]. Table 5.4 shows experimental results. It can be seen that our method outperforms the state-of-the-art by large margins under different numbers of training classes. With a training set of 60 classes, our method is on par with state-of-the-art methods that are trained by using 100-class training set. This clearly shows the effectiveness of our proposed optimal matching.

Effect of Multi-Scale and Cross-Scale Matching Manners: We conduct experiments on different combinations of our proposed optimal matching strategies. Table 5.5 shows experimental results under evaluation protocol 2. We can see that including any of our proposed matching strategies (M_s, M_t, C_s, and C_t)

TABLE 5.4: Experiments on different sizes of the auxiliary training set for one-shot skeleton recognition on NTU RGB+D 120 dataset. (S-scale: single-scale matching; M-scale: multi-scale matching; M&C-scale: multi-scale and cross-scale matching)

Train Classes	20	40	60	80	100
APSR [98]	29.1	34.8	39.2	42.8	45.3
SL-DML [100]	36.7	42.4	49.0	46.4	50.9
Skeleton-DML [101]	28.6	37.5	48.6	48.0	54.2
uDTW [104]	32.2	39.0	41.2	45.3	49.0
JEANIE [103]	38.5	44.1	50.3	51.2	57.0
ProtoNet [35]	36.7	45.7	54.4	55.5	60.4
FEAT [159]	37.6	44.2	52.2	55.4	61.5
Subspace [157]	37.7	45.3	54.2	55.5	60.9
Dynamic Filter [158]	34.5	43.0	54.0	52.9	60.6
S-scale (Ours)	37.9	46.5	54.6	58.7	63.2
M-scale (Ours)	41.2	52.6	59.0	62.4	67.6
M&C-scale (Ours)	**44.1**	**55.3**	**60.3**	**64.2**	**68.7**

TABLE 5.5: Evaluation of different combinations of optimal matching manners under the Evaluation Protocol 2: The second line shows the baseline result under the single-scale matching manner. (M_t: multi-temporal scale matching; M_s: multi-spatial scale matching; C_t: cross-temporal scale matching; C_s: cross-spatial scale matching)

M_t	M_s	C_t	C_s	NTU	NTU 120	PKU-MMD
				77.4	63.2	82.8
✓				79.3	65.0	85.0
	✓			79.6	65.1	84.9
✓	✓			81.6	67.6	85.7
✓		✓		80.7	66.7	85.7
	✓		✓	80.8	67.3	86.0
✓	✓	✓	✓	**82.7**	**68.7**	**86.9**

improves the one-shot skeleton action recognition clearly. Including all four matching strategies performs simply the best over all three datasets, demonstrating the effectiveness of our proposed optimal matching technique.

Effect of Multiple Spatial Scales and Multiple Temporal Scales: We conduct an analysis how models with different spatial and temporal scales (as the backbone) perform for the one-shot skeleton action recognition task. Table 5.6 shows the one-shot learning performance with different combinations of scales. We can observe that the model performs the best when combining scales 1, 2, and

TABLE 5.6: Evaluation of our proposed multi-scale matching with different scale combinations. The experiments were conducted on NTU RGB+D 120 dataset under the Evaluation Protocol 2.

scales	Multi-Spatial Scale	Multi-Temporal Scale
1	63.2	63.2
1,2	64.3	64.4
1,3	64.6	64.2
1,2,3	**65.1**	**65.0**

3. In addition, models employing two scales (scales 1 and 2 or scales 1 and 3) also outperform the model using scale 1 only, showing the benefits of the proposed multi-scale representations.

5.4 Summary

In this study, we approach the challenge of one-shot skeleton action recognition by formulating it as a matching problem We obtain the multi-spatial and multi-temporal scale features by designing hierarchical pooling that represents the same skeleton sequence at various spatial and temporal scales. Moreover, based on the multi-scale skeleton features, we propose a multi-scale skeleton matching strategy and a cross-scale skeleton matching method to measure the semantic relevance between two skeleton sequences for one-shot skeleton action recognition. The experiments demonstrate that our proposed method achieves superior one-shot skeleton action recognition performance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis delves into the fewer-label scenario for skeleton action recognition. It presents a range of methodologies, spanning weak-supervised, self-supervised, and one-shot learning, to tackle the challenges associated with fewer label data.

In light of the shared joint-aware feature foundation in both gesture recognition and hand pose estimation, in our first work presented in Chapter 3, we propose a weakly supervised learning scheme that is capable of leveraging hand pose (or gesture) annotations to learn powerful gesture recognition (or pose estimation) models. Specifically, we present a collaborative learning method for joint gesture recognition and 3D hand pose estimation. Our model learns in a collaborative way to recurrently exploit the joint-aware feature to progressively boost the performance of each task. Additionally, we develop a multi-order multi-stream model to learn motion information in the intermediate feature maps and designed a multi-scale relation module to extract semantic information of the hierarchical hand structure. This weakly supervised learning strategy greatly relieves the data annotation burden, especially considering the very limited annotated 3D pose data and the wide availability of annotated hand gesture data. Our proposed collaborative learning

network achieves state-of-the-art performance for both gesture recognition and 3D hand pose estimation tasks.

Recently, some self-supervised learning methods in the image domain have shown promising results, yet their potential in skeleton action recognition remains underexplored. In our second work presented in Chapter 4, we tackle self-supervised representation learning for skeleton action recognition. We introduce a novel coarse-fine masked skeleton cloud colorization method that is capable of learning skeleton representations from unlabeled data. Specifically, we obtain colored skeleton cloud representations by stacking skeleton sequences to 3D skeleton clouds, and then assigning colors to each point based on its temporal and spatial order in the skeleton sequences. Furthermore, we incorporate a dual-stream pre-training framework that capitalizes on both coarse and fine-grained colorization, facilitating the acquisition of multi-scale spatial-temporal features. Additionally, we introduce a Masked Skeleton Cloud Repainting task to pre-train the designed auto-encoder framework. Extensive experiments across different datasets demonstrate that our proposed method achieves superior unsupervised and semi-supervised action recognition performance.

The third part of this thesis delves into a more challenging fewer-label scenario in skeleton action recognition, the one-shot learning scenarios, which aim to recognize new action categories from a single reference example. In the third work presented in Chapter 5, we study the one-shot learning in skeleton action recognition and formulate one-shot skeleton action recognition as an optimal matching problem. We introduce a multi-scale matching strategy that can capture scale-wise skeleton semantic relevance at multiple spatial and temporal scales. On top of that, we design a novel cross-scale matching scheme that can model the within-class variation of human actions in motion magnitudes and motion paces. To the best of our knowledge, this is the first work that exploits multi-scale representations and cross-scale matching to capture multi-scale skeleton semantic relevance and maintain consistency across motion scales in one-shot skeleton action recognition.

Extensive experiments on three public datasets show that our method outperforms the state-of-the-art consistently by large margins.

6.2 Future Work

Our current investigation is centered on weakly-supervised and self-supervised techniques that address the challenge of scant labels in intra-dataset/format evaluations, alongside the one-shot learning approach, which uniquely enables the recognition of actions from previously unseen classes with only a single example.

This exploration naturally gives rise to two critical inquiries:
(1) How should we address evaluations that require navigating across diverse datasets or formats?
(2) What strategies can we employ in scenarios devoid of any label data for novel, unseen classes?

To address these pivotal concerns, our forthcoming studies will delve into leveraging zero-shot learning techniques and cross-domain methodologies as potential solutions.

Zero-shot Learning for Skeleton Action Recognition. In Chapter 5, we have studied the challenge scenario that there is only one sample of the new category during practice. On the other hand, zero-shot learning represents an even more challenging scenario with limited label information. Here, only semantic data such as names, attributes, or descriptions of new classes are accessible for the new categories. There is a growing demand for zero-shot skeleton-based action recognition in real-world applications because it eliminates the extensive process of gathering and annotating new actions. Currently, only a limited number of studies [170, 171] have addressed zero-shot skeleton-based action recognition, which is still a nascent research area. Therefore, zero-shot learning in skeleton action recognition is worthy of exploration in future work.

Cross-domain Skeleton Action Recognition. Recent developments in skeleton methodologies primarily focus on training and evaluating within the common intra-dataset scenario. However, in real-world applications, it is not easy to hold such an assumption, because labeling a dataset with the same distribution as the target data is a laborious task. Ideally, it would be more efficient to leverage an existing public annotated skeleton or even an image dataset as the source. Yet, a notable domain gap usually exists between these source and target datasets. This variance stems from multiple sources, including the type of devices utilized (like 3D sensors [172, 173] or image-based pose estimation tools [174, 175]), diverse camera setups (with altered viewpoints), and contrasting environments (ranging from controlled laboratory settings to dynamic real-world conditions). These elements lead to distinct skeleton styles across datasets in terms of joint types, quality, viewpoints, and action categories. Despite the domain gap being a critical issue in skeleton-based action recognition, it remains largely unaddressed in current research. Thus the cross-domain problem in skeleton action recognition is worthy of exploration in future work.

List of Author's Awards, Patents, and Publications

Journal Articles

- **Siyuan Yang**, Jun Liu, Shijian Lu, Er Meng Hua, Yongjian Hu, and Alex C. Kot, "Self-Supervised 3D Action Representation Learning with Skeleton Cloud Colorization", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

- **Siyuan Yang**, Jun Liu, Shijian Lu, Er Meng Hua, and Alex C. Kot, "One-Shot Action Recognition via Multi-Scale Spatial-Temporal Skeleton Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

- Jianhong Pan, **Siyuan Yang**, Qiuhong Ke, Zhipeng Fan, Hossein Rahmani, and Jun Liu, "Progressive Channel-Shrinking Network", *IEEE Transactions on Multimedia.*

- Yang Yu, Rongrong Ni, **Siyuan Yang**, Yao Zhao, Alex C. Kot, "Narrowing Domain Gaps with Bridging Samples for Generalized Face Forgery Detection", *IEEE Transactions on Multimedia.*

- Yang Yu, Xiaolong liu, Rongrong Ni, **Siyuan Yang**, Yao Zhao, Alex C. Kot, "PVASS-MDD: Predictive Visual-audio Alignment Self-supervision for Multimodal Deepfake Detection", *IEEE Transactions on Circuits and Systems for Video Technology.*

Conference Proceedings

- **Siyuan Yang**, Jun Liu, Shijian Lu, Er Meng Hua, and Alex C. Kot, "Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis," in *Proceedings of the European Conference on Computer Vision (ECCV), 2020 (Spotlight).*

- **Siyuan Yang**, Jun Liu, Shijian Lu, Er Meng Hua, and Alex C. Kot, "Skeleton cloud colorization for unsupervised 3d action representation learning," in *Proceedings of the International Conference on Computer Vision (ICCV), 2021.*

- Chenyu Yi*, **Siyuan Yang***, Haoliang Li, Yap-Peng Tan, and Alex C. Kot, "Benchmarking the Robustness of Spatial-Temporal Models Against Corruptions," in *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021.*

- Chenyu Yi, **Siyuan Yang**, Haoliang Li, Yap-Peng Tan, and Alex C. Kot, "Temporal Coherent Test Time Optimization for Robust Video Classification," in *Proceedings of International Conference on Learning Representations (ICLR), 2023.*

- Hao Cheng, **Siyuan Yang**, Joey Tianyi Zhou, Lanqing Guo, and Bihan Wen, "Frequency Guidance Matters in Few-Shot Learning," in *Proceedings of the International Conference on Computer Vision (ICCV), 2023.*

*: equal contribution

Bibliography

[1] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018. xv, 30, 31, 32, 33

[2] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634. 1

[3] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal lstm network with trust gates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 3007–3021, 2017. 1, 2, 11, 37, 38, 87

[4] H. Rahmani and A. Mian, "3d action recognition from novel viewpoints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016. 31

[5] Z. Jiang, V. Rozgic, and S. Adali, "Learning spatiotemporal features for infrared action recognition with 3d convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2017, pp. 115–123.

[6] Y. Wang, Y. Xiao, F. Xiong, W. Jiang, Z. Cao, J. T. Zhou, and J. Yuan, "3dv: 3d dynamic voxel for action recognition in depth video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 511–520. 1, 37

[7] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-based action recognition using motion information and spiking neural networks." in *IJCAI*, 2021, pp. 1743–1749.

[8] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 286–293. 1

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308. 1, 7, 9, 10, 37

[10] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576. 7, 9, 18, 23, 31

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2740–2755, 2018. 1, 37

[12] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723. 1, 31, 37

[13] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition,"

in *European conference on computer vision*. Springer, 2014, pp. 742–757. 1, 37, 54, 57

[14] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3288–3297. 1, 2, 11, 37, 38

[15] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 816–833. 2, 8, 11, 18, 38, 67

[16] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2, 12, 37, 38, 76, 85

[17] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1110–1118. 2, 11, 38, 46, 59, 67

[18] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016. 45, 54, 56, 67, 83

[19] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2117–2126. 2, 11, 38

[20] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *2015 3rd IAPR Asian Conference on Pattern Recognition*. IEEE, 2015, pp. 579–583. 2, 11, 38

[21] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *2017 IEEE International Conference on Multimedia & Expo Workshops*. IEEE, 2017, pp. 597–600. 11

[22] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, and P. Ogunbona, "Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 595–604. 2, 11, 38

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. 2, 38

[24] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[25] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[26] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015.

[27] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[28] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2688–2697. 2, 38

[29] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2015, pp. 685–694. 3

[30] J. Peyre, J. Sivic, I. Laptev, and C. Schmid, "Weakly-supervised learning of visual relations," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5179–5188.

[31] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018. 3

[32] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020. 3

[33] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6707–6717.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738. 3, 13

[35] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4077–4087, 2017. 3, 75, 76, 83, 84, 86, 87, 89

[36] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[37] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016. 3, 75, 84

[38] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," 2008. 7, 9

[39] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[41] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558. 7, 9

[42] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459. 7, 9, 10

[43] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *European conference on computer vision*, 2018, pp. 305–321. 7, 9, 10

[44] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36. 7, 9, 18, 23, 30

[45] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 843–10 852. 8

[46] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 49–59. 18

[47] M. Rad, M. Oberweger, and V. Lepetit, "Domain transfer for 3d pose estimation from color images without manual annotations," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 69–84.

[48] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4903–4911. 8, 18, 21

[49] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018. 8, 18

[50] B. Tekin, F. Bogo, and M. Pollefeys, "H+o: Unified egocentric recognition of 3d hand-object poses and interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. 8, 18, 30, 31, 33

[51] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9. 8, 18

[52] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1656. 67, 87

[53] X. S. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "A neural network based on spd manifold learning for skeleton-based hand gesture recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. 8, 18

[54] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018. 8

[55] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1293–1301. 8

[56] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *European conference on computer vision*, 2018, pp. 666–682. 9, 18

[57] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3d human pose estimation in the wild: A weakly-supervised approach," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017. 9, 21, 29

[58] X. Chen, K.-Y. Lin, W. Liu, C. Qian, and L. Lin, "Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 895–10 904. 9

[59] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017. 10

[60] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497. 10

[61] 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*, vol. 35, no. 1, pp. 221–231, 2012. 10

[62] A. Diba, M. Fayyaz, V. Sharma, M. M. Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool, "Spatio-temporal channel correlation networks for action classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–299. 10

[63] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6202–6211. 10

[64] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy 10

[65] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, vol. 2, no. 3, 2021, p. 4. 10

[66] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE international conference on computer vision*, 2021, pp. 6836–6846. 10

[67] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid, "Multiview transformers for video recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3333–3343. 10

[68] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211. 10

[69] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows,"

in *Proceedings of the IEEE international conference on computer vision*, 2021, pp. 10 012–10 022. 10

[70] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017. 11

[71] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 786–792. 11, 12

[72] T. Soo Kim and A. Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 20–28. 11

[73] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI conference on artificial intelligence*, 2018. 12, 38, 67

[74] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 12, 38

[75] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 183–192. 12, 67

[76] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368. 12, 67

[77] H.-g. Chi, M. H. Ha, S. Chi, S. W. Lee, Q. Huang, and K. Ramani, "Infogcn: Representation learning for human skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 186–20 196. 12, 67

[78] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Computer Vision and Image Understanding*, vol. 208, p. 103219, 2021. 12

[79] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[80] Y. Zhang, B. Wu, W. Li, L. Duan, and C. Gan, "Stst: Spatial-temporal specialized transformer for skeleton-based action recognition," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3229–3237. 12

[81] Z. Gao, P. Wang, P. Lv, X. Jiang, Q. Liu, P. Wang, M. Xu, and W. Li, "Focal and global spatial-temporal transformer for skeleton-based action recognition," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 382–398. 12

[82] W. Xin, Q. Miao, Y. Liu, R. Liu, C.-M. Pun, and C. Shi, "Skeleton mixformer: Multivariate topology representation for skeleton-based action recognition," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 2211–2220.

[83] Q. Wang, S. Shi, J. He, J. Peng, T. Liu, and R. Weng, "Iip-transformer: Intra-inter-part transformer for skeleton-based action recognition," in *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 936–945. 12

[84] D. Ahn, S. Kim, H. Hong, and B. C. Ko, "Star-transformer: a spatio-temporal cross attention transformer for human action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3330–3339. 12

[85] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852. 13

[86] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2203–2212. 13

[87] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Unsupervised learning of view-invariant action representations," in *Advances in Neural Information Processing Systems*, 2018, pp. 1254–1264. 13, 67

[88] N. Zheng, J. Wen, R. Liu, L. Long, J. Dai, and Z. Gong, "Unsupervised representation learning with long-term dynamics for skeleton based action recognition," in *Thirty-Second AAAI conference on artificial intelligence*, 2018. 13, 38, 54, 63, 64, 65, 66, 68

[89] J. N. Kundu, M. Gor, P. K. Uppala, and V. B. Radhakrishnan, "Unsupervised feature learning of human actions as trajectories in pose embedding manifold," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, 2019, pp. 1459–1467. 13, 63

[90] K. Su, X. Liu, and E. Shlizerman, "Predict & cluster: Unsupervised skeleton based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9631–9640. 13, 38, 54, 63

[91] H. Rao, S. Xu, X. Hu, J. Cheng, and B. Hu, "Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition," *Information Sciences*, vol. 569, pp. 90–109, 2021. 13, 63, 65

[92] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang, "3d human action representation learning via cross-view consistency pursuit," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021, pp. 4741–4750. 13, 38, 63, 64, 66, 67

[93] T. Guo, H. Liu, Z. Chen, M. Liu, T. Wang, and R. Ding, "Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 762–770. 13, 63, 64, 66, 67

[94] B. Kim, H. J. Chang, J. Kim, and J. Y. Choi, "Global-local motion transformer for unsupervised skeleton-based action learning," in *European conference on computer vision*, 2022, pp. 209–225. 14, 63, 64, 65

[95] H. Zhang, Y. Hou, W. Zhang, and W. Li, "Contrastive positive mining for unsupervised 3d action representation learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 36–51. 14

[96] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758. 14

[97] J. Zhang, L. Lin, and J. Liu, "Hierarchical consistent contrastive learning for skeleton-based action recognition with growing augmentations," *The Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023. 14, 63

[98] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020. 14, 54, 56, 72, 78, 83, 84, 85, 86, 87, 88, 89

[99] A. Sabater, L. Santos, J. Santos-Victor, A. Bernardino, L. Montesano, and A. C. Murillo, "One-shot action recognition in challenging therapy scenarios," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2777–2785, 2021. 14, 86, 87

[100] R. Memmesheimer, N. Theisen, and D. Paulus, "Sl-dml: Signal level deep metric learning for multimodal one-shot action recognition," in *2020 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 4573–4580. 14, 86, 87, 88, 89

[101] R. Memmesheimer, S. Häring, N. Theisen, and D. Paulus, "Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022. 14, 72, 78, 84, 86, 87, 88, 89

[102] N. Ma, H. Zhang, X. Li, S. Zhou, Z. Zhang, J. Wen, H. Li, J. Gu, and J. Bu, "Learning spatial-preserved skeleton representations for few-shot action recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 174–191. 15

[103] L. Wang and P. Koniusz, "Temporal-viewpoint transportation plan for skeletal few-shot action recognition," 2022. 15, 84, 86, 87, 88, 89

[104] L. Wang and P. Koniusz, "Uncertainty-dtw for time series and sequences," in *European Conference on Computer Vision*. Springer, 2022, pp. 176–195. 15, 84, 86, 87, 88, 89

[105] A. Zhu, Q. Ke, M. Gong, and J. Bailey, "Adaptive local-component-aware graph convolutional network for one-shot skeleton-based action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6038–6047. 15

[106] M. Abavisani, H. R. V. Joze, and V. M. Patel, "Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2019. 18

[107] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," in *12th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 2017, pp. 438–445. 18

[108] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 20, 22, 24

[109] J. Liu, H. Ding, A. Shahroudy, L.-Y. Duan, X. Jiang, G. Wang, and A. C. Kot, "Feature boosting network for 3d pose estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 494–501, 2019. 21, 29

[110] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for rgb-d activity recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2015. 31

[111] L. Lin, S. Song, W. Yang, and J. Liu, "Ms2l: Multi-task self-supervised learning for skeleton based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2490–2498. 38, 54, 63, 64, 65, 66, 67, 68

[112] Q. Nie and Y. Liu, "View transfer on human skeleton pose: Automatically disentangle the view-variant and view-invariant information for pose representation learning," *International Journal of Computer Vision*, pp. 1–22, 2020. 38, 54, 63

[113] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "Potion: Pose motion representation for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7024–7033. 42

[114] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multi-scale graph neural networks for 3d skeleton based human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 214–223. 46, 59

[115] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics*, vol. 38, no. 5, pp. 1–12, 2019. 48

[116] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215. 48

[117] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2. 51

[118] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009. 51

[119] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Skeleton cloud colorization for unsupervised 3d action representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 13 423–13 433. 54, 63, 64, 65, 67, 68

[120] C. Si, X. Nie, W. Wang, L. Wang, T. Tan, and J. Feng, "Adversarial self-supervised learning for semi-supervised 3d action recognition," in *European conference on computer vision*, 2020. 54, 64, 65

[121] J. Liu, S. Song, C. Liu, Y. Li, and Y. Hu, "A benchmark dataset and comparison study for multi-modal human action analytics," *ACM Transactions on*

Multimedia Computing, Communications, and Applications, vol. 16, no. 2, pp. 1–24, 2020. 54, 56, 83

[122] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2649–2656. 54, 57

[123] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome: Real-world activities of daily living," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2019. 57, 68

[124] S. Xu, H. Rao, X. Hu, J. Cheng, and B. Hu, "Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition," *IEEE Transactions on Multimedia*, 2021. 63

[125] H. Qiu, Y. Wu, M. Duan, and C. Jin, "Glta-gcn: Global-local temporal attention graph convolutional network for unsupervised skeleton-based action recognition," in *IEEE International Conference on Multimedia and Expo*, 2022, pp. 1–6. 63

[126] Z. Xu, X. Shen, Y. Wong, and M. S. Kankanhalli, "Unsupervised motion representation learning with capsule autoencoders," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3205–3217, 2021. 63

[127] A. B. Tanfous, A. Zerroug, D. Linsley, and T. Serre, "How and what to learn: Taxonomizing self-supervised learning for 3d action recognition." in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2888–2897. 63

[128] P. Wang, J. Wen, C. Si, Y. Qian, and L. Wang, "Contrast-reconstruction representation learning for self-supervised skeleton-based action recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 6224–6238, 2022. 63

[129] X. Gao, Y. Yang, Y. Zhang, M. Li, J.-G. Yu, and S. Du, "Efficient spatio-temporal contrastive learning for skeleton-based 3d action recognition," *IEEE Transactions on Multimedia*, 2021. 63

[130] Y.-B. Cheng, X. Chen, J. Chen, P. Wei, D. Zhang, and L. Lin, "Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2021, pp. 1–6. 63

[131] Y. Zhan, Y. Chen, P. Ren, H. Sun, J. Wang, Q. Qi, and J. Liao, "Spatial temporal enhanced contrastive and pretext learning for skeleton-based action representation," in *Asian Conference on Machine Learning*. PMLR, 2021, pp. 534–547. 63, 64, 65

[132] H. Zhang, Y. Hou, and W. Zhang, "Skeletal twins: Unsupervised skeleton-based action representation learning," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2022, pp. 1–6. 63, 64, 67

[133] F. M. Thoker, H. Doughty, and C. G. Snoek, "Skeleton-contrastive 3d action representation learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1655–1663. 63, 64, 66, 68

[134] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 1476–1485. 64, 65

[135] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013. 64, 65

[136] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018. 64, 65

[137] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in neural information processing systems*, 2005, pp. 529–536. 64, 65

[138] J. Li and E. Shlizerman, "Sparse semi-supervised action recognition with active learning," *arXiv preprint arXiv:2012.01740*, 2020. 64, 65

[139] X. Shu, B. Xu, L. Zhang, and J. Tang, "Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2022. 64, 65

[140] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3d human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 914–927, 2013. 67

[141] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1012–1020. 67

[142] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1963–1978, 2019. 67

[143] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595–3603. 67

[144] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 67

[145] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1227–1236. 67

[146] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1625–1633. 67

[147] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof trace for generic face anti-spoofing," in *European Conference on Computer Vision*. Springer, 2020, pp. 406–422. 67, 72, 77

[148] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, and F. Bremond, "Unik: A unified framework for real-world skeleton-based action recognition," *The British Machine Vision Conference*, 2021. 68

[149] ——, "View-invariant skeleton action representation learning via motion retargeting," *International Journal of Computer Vision*, pp. 1–16, 2024. 68

[150] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2272–2281. 72, 77

[151] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[152] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis," in *European Conference on Computer Vision*. Springer, 2020, pp. 769–786.

[153] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1113–1122. 72, 77

[154] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, pp. 2292–2300, 2013. 73

[155] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338. 73

[156] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000. 73, 78, 83

[157] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145. 75, 76, 83, 86, 87, 89

[158] C. Xu, Y. Fu, C. Liu, C. Wang, J. Li, F. Huang, L. Zhang, and X. Xue, "Learning dynamic alignment via meta-filter for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5182–5191. 86, 87, 89

[159] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8808–8817. 86, 87, 89

[160] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020. 75, 76, 79, 83, 84

[161] M. Bishay, G. Zoumpourlis, and I. Patras, "Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition," *30th British Machine Vision Conference*, p. 154, 2019. 75, 76

[162] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles, "Few-shot video classification via temporal alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 618–10 627. 84

[163] H. Zhang, L. Zhang, X. Qi, H. Li, P. H. S. Torr, and P. Koniusz, "Few-shot action recognition with permutation-invariant attention," in *European conference on computer vision*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020, pp. 525–542. 84

[164] X. Wang, S. Zhang, Z. Qing, M. Tang, Z. Zuo, C. Gao, R. Jin, and N. Sang, "Hybrid relation guided set matching for few-shot action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 19 948–19 957.

[165] A. Thatipelli, S. Narayan, S. Khan, R. M. Anwer, F. S. Khan, and B. Ghanem, "Spatio-temporal relation modeling for few-shot action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 19 958–19 967.

[166] L. Zhu and Y. Yang, "Compound memory networks for few-shot video classification," in *European conference on computer vision*, 2018, pp. 751–766. 75, 76

[167] Chen and et al., "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019. 83

[168] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *ICCV*, 2021, pp. 9062–9071. 83

[169] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12 203–12 213. 83

[170] B. Jasani and A. Mazagonwalla, "Skeleton based zero shot action recognition in joint pose-language semantic space," *arXiv preprint arXiv:1911.11344*, 2019. 93

[171] P. Gupta, D. Sharma, and R. K. Sarvadevabhatla, "Syntactically guided generative embeddings for zero-shot skeleton action recognition," in *IEEE International Conference on Image Processing*. IEEE, 2021, pp. 439–443. 93

[172] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *International conference on advanced robotics*. IEEE, 2015, pp. 388–394. 94

[173] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 1–10. 94

[174] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299. 94

[175] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2500–2509. 94