Learning effective video representations for action recognition

Di Yang

UNIVERSITÉ CÔTE D'AZUR | ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Apprendre des Représentations Vidéo Efficaces Pour La Reconnaissance d'Actions

Di YANG

Centre Inria d'Université Côte d'Azur, Équipe STARS

Présentée en vue de l'obtention du grade de docteur en Informatique **d**'Université Côte d'Azur

Dirigée par : François BRÉMOND
Inria, France

Soutenue le : 1 Février 2024

Devant le jury, composé de :

Président du jury :
Matthieu CORD
Sorbonne Université & Valeo.ai, France

Rapporteurs :
Karteek ALAHARI
Inria, France

Jürgen GALL
University of Bonn, Allemagne

Examinateurs :
Wanli OUYANG
The Chinese University of Hong Kong & Shanghai AI Laboratory, Chine

Gianpiero FRANCESCA
Toyota Motor Europe, Belgique (Invité)

Inria

Learning Effective Video Representations for Action Recognition

Di Yang

Inria Center at Université Côte d'Azur
STARS Team

This dissertation is submitted for the degree of
Doctor of Philosophy

Inria April 2024

Acknowledgements

Writing this PhD thesis has been a journey filled with growth, learning, and discovery, and I am deeply grateful to the many individuals who have supported me along the way.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, François Brémond, for his unwavering guidance, insightful feedback, and mentorship throughout my doctoral research. His expertise and encouragement have been instrumental in shaping the direction and quality of this work.

I extend my gratitude to the members of my Ph.D. committee: Matthieu Cord, Jürgen Gall, Karteek Alahari and Wanli Ouyang, for their valuable insights, constructive criticism, and the time they dedicated to evaluating my research. Their collective expertise has enriched the academic rigor of this thesis. It is an honor to have you at my defense. I would like to especially thank my thesis reviewers Jürgen Gall and Karteek Alahari. Thank you for devoting your time and effort to review my manuscript.

I would like to acknowledge the financial support provided by Toyota Motor Europe, which enabled me to carry out this research. This support has been pivotal in realizing the goals of my Ph.D. research. I am especially thankful to Gianpiero Francesca and Lorenzo Garattoni for the support and advice. Thank you to give me the opportunity to join the Toyota projects and provide an internship position to understand the transformation of my research projects onto industrial application. These projects give me a very good starter for my Ph.D. study and I indeed obtain numerous inspirations from this project. I am also grateful to Quan Kong from Woven by Toyota for his suggestions and collaborations.

I am thankful to the Inria of University Côte d'Azur for providing me with the resources and the cluster NEF, infrastructure, and academic environment necessary for conducting this research. The support of the faculty, research staff, and fellow students has created a stimulating intellectual atmosphere that has greatly contributed to my growth as a researcher. I would like to give special thanks to Yaohui Wang and Antitza Dantcheva. Your initial suggestions led me to be a curious man and take up academic research seriously.

Many thanks to the music room at Inria and my friends who play music with me, especially my bandmates in ”For the moment”. I truly enjoyed my time with all of you.

My gratitude extends to my colleagues and research collaborators at STARS team, for example, Rui, Hao, Srijan, Valeriya, Snehashis, Mahmoud, Cyprien, David, Farhood, Ujjwal, Sabine, Jean-Paul, Monique and Sandrine. I thank all those who are with me and have made my stay memorable and who helped me along the long way to obtain this degree. I am truly grateful for the memories we created.

Last but not the least, I am profoundly grateful to my parents, Fengping Yang and Yu Zhang, for their unwavering support throughout my PhD journey in France. Their belief in me and constant encouragement were my anchor during late nights and weekends devoted to research. Without their support, this achievement wouldn’t have been possible.

In writing this acknowledgment, I am reminded of the truth in the African proverb, "If you want to go fast, go alone. If you want to go far, go together." To all those who have accompanied me on this journey, thank you for helping me go far.

Abstract

Human action recognition is an active research field with significant contributions to applications such as home-care monitoring, human-computer interaction, and game control. However, recognizing human activities in real-world videos remains challenging in learning effective video representations that have a high expressive power to represent human spatio-temporal motion, view-invariant actions, complex composable actions, etc. To address this challenge, this thesis makes three contributions towards learning such effective video representations that can be applied and evaluated on real-world human action classification and segmentation tasks by transfer-learning. The first contribution is to improve the generalizability of human skeleton motion representation models. We propose a unified framework for real-world skeleton human action recognition. The framework includes a novel skeleton model that not only effectively learns spatio-temporal features on human skeleton sequences but also generalizes across datasets. The second contribution extends the proposed framework by introducing two novel joint skeleton action generation and representation learning frameworks for different downstream tasks. The first is a self-supervised framework for learning from synthesized composable motions for skeleton-based action segmentation. The second is a View-invariant model for self-supervised skeleton action representation learning that can deal with large variations across subjects and camera viewpoints. The third contribution targets general RGB-based video action recognition. Specifically, a time-parameterized contrastive learning strategy is proposed. It captures time-aware motions to improve performance of action classification in fine-grained and human-oriented tasks. Experimental results on benchmark datasets demonstrate that the proposed approaches achieve state-of-the-art performance in action classification and segmentation tasks. The proposed frameworks improve the accuracy and interpretability of human activity recognition and provide insights into the underlying structure and dynamics of human actions in videos. Overall, this thesis contributes to the field of video understanding by proposing novel methods for skeleton-based action representation learning, and general RGB video representation learning. Such representations benefit both action classification and segmentation tasks.

keywords: Video understanding, action recognition, motion generation

Résumé

La reconnaissance des actions humaines est un domaine de recherche actif avec d'importantes contributions à des applications telles que la surveillance des soins à domicile, l'interaction homme-ordinateur et le contrôle de jeu. Cependant, la reconnaissance des activités humaines dans les vidéos du monde réel reste un défi en termes d'apprentissage de représentations vidéo efficaces ayant un fort pouvoir expressif pour représenter le mouvement spatio-temporel humain, les actions invariantes par rapport à la vue, les actions composites complexes, etc. Pour relever ce défi, cette thèse apporte trois contributions visant à apprendre de telles représentations vidéo efficaces pouvant être appliquées et évaluées sur des tâches réelles de classification et de segmentation d'actions humaines par transfert d'apprentissage. La première contribution vise à améliorer la généralisabilité des modèles de représentation du mouvement du squelette humain. Nous proposons un cadre unifié pour la reconnaissance d'actions humaines basées sur le squelette dans le monde réel. Le cadre comprend un nouveau modèle de squelette qui non seulement apprend efficacement des caractéristiques spatio-temporelles sur les séquences de squelette humain, mais se généralise également à travers les ensembles de données. La deuxième contribution étend le cadre proposé en introduisant deux nouveaux cadres de génération d'actions squelettiques conjointes et d'apprentissage de représentation pour différentes tâches en aval. Le premier est un cadre auto-supervisé pour l'apprentissage à partir de mouvements composites synthétisés pour la segmentation d'actions basées sur le squelette. Le second est un modèle invariant de vue pour l'apprentissage auto-supervisé de la représentation d'actions squelettiques qui peut traiter de grandes variations entre les sujets et les points de vue de la caméra. La troisième contribution cible la reconnaissance générale d'actions vidéo basée sur RGB. Plus précisément, une stratégie d'apprentissage contrastif paramétré dans le temps est proposée. Elle capture les mouvements sensibles au temps pour améliorer les performances de la classification d'actions dans des tâches fines et axées sur l'humain. Les résultats expérimentaux sur des ensembles de données de référence démontrent que les approches proposées atteignent des performances de pointe dans les tâches de classification et de segmentation d'actions. Les cadres proposés améliorent la précision et l'interprétabilité de la reconnaissance des activités humaines et fournissent des informations sur la structure sous-jacente et la dynamique des actions humaines dans les vidéos. Dans l'ensemble, cette thèse contribue au domaine de la compréhension vidéo en proposant de nouvelles méthodes pour l'apprentissage de représentations d'actions basées sur le squelette et l'apprentissage de représentations vidéo

RGB générales. De telles représentations bénéficient à la fois des tâches de classification et de segmentation d'actions.

mots clés: Compréhension vidéo, reconnaissance d'actions, génération de mouvements, représentation de squelettes

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Video understanding, a cornerstone of computer vision and artificial intelligence, involves the interpretation and analysis of visual information contained within video sequences. With the proliferation of digital cameras, smartphones, and online video platforms, the volume of generated video data has skyrocketed, creating an urgent need for automated methods that can extract meaningful insights from these visual streams. Video understanding encompasses a range of tasks, from recognizing objects, scenes, and actions to comprehending complex narratives and interactions.

Human action recognition has become an active research field and an important topic of video understanding in recent years, with significant contributions towards many current applications, such as video surveillance, human-computer interaction, game control, and robotics (see Fig. 1.1). The ability to recognize and interpret human actions in videos has important implications for a wide range of domains, including healthcare, sports analysis, security, and entertainment. However, recognizing human activities in real-world videos remains a challenging task that requires effective representation learning methods. Specifically, given an input video, we need to first encode this video into a vector that represents its features using an encoder. Then, the representation vector is used as the input of an action classifier for video-level action classification tasks or frame-level segmentation tasks. As an important intermediary between the original video and the action category, good video representations can help to learn the accurate mapping of videos and their corresponding action labels. Hence, what are good video representations? How to learn such representations? How to evaluate the learned representations? This thesis mainly focuses on the representation learning stage of action recognition and explores to answer these questions.

Fig. 1.1 **Real-word Action Classification System and Applications.** For instance, when applied within the context of robotics for elderly care, this technology holds the promise of enabling robots to not only comprehend human actions but also to anticipate needs, to offer timely aid, and to establish meaningful human-robot interactions.

1.1 Goals

The primary goal of this thesis is to propose novel approaches for learning effective action representations and to understand human activities in real-world videos. We aim to improve the accuracy and interpretability of human activity recognition and segmentation models and to provide insights into the underlying structure and dynamics of human actions in videos. Specifically, this thesis involves mainly the Action Representation Learning task and two downstream tasks in human action recognition: Action Classification and Temporal Action Segmentation. Below, we provide the problem statements and their definitions.

Action Classification in Trimmed Videos: Action classification in trimmed videos refers to the downstream task of recognizing the action label of human activity in a short video segment on top of the learned action representations. It is considered as a relatively simpler task than action detection and segmentation in untrimmed videos since the temporal extent of the action is usually well-defined. However, it still poses significant challenges, such as intra-class variations, inter-class similarities, and occlusions.

Action Segmentation in Untrimmed Videos: Action detection and segmentation in untrimmed videos refer to the downstream task of detecting and localizing actions within a long video sequence after action representation learning. It is considered a more challenging task than action classification in trimmed videos since the temporal extent of the action is not well-defined and the video may contain multiple actions with different temporal durations. The main challenges in this task are handling long-term temporal dependencies,

co-occurrence, dense labeling and interactions, and dealing with a large amount of irrelevant background in the video.

Action Representation Learning: An important challenge in human action recognition is the design of effective representation learning methods that can capture spatio-temporal features of human actions in videos. Effective representations for human-centric action understanding need to be generic and to be able to clearly represent the human motion details. However, most existing approaches rely on global features or simple motion representations, which may not capture complex motion patterns and temporal dependencies in videos. Therefore, learning discriminative and robust representations of human actions is crucial for improving the performance of human action recognition. In this thesis, we focus on proposing effective representation learning approaches and we analyze the performance improvements on the action classification and action segmentation tasks for evaluating the learned representations.

1.2 Applications

Action recognition in videos, leveraging the spatial-temporal information inherent in human motions, proves to be a versatile technology with applications in numerous fields. The ability to interpret and classify human movements from motion data opens the door to innovative solutions and improved understanding of human behavior.

Human-computer Interaction and Human-robot Interaction: In the realm of human-computer interaction, video action recognition enables natural and intuitive interactions between humans and computers. Gesture-based controls, sign language interpretation, and facial expressions analysis are some applications where recognizing human actions enhances the user experience. Moreover, action recognition plays a crucial role in human-robot interaction scenarios. Robots equipped with the ability to interpret human movements can better understand user commands, assist in daily tasks, and ensure safe collaboration in shared spaces.

Human Activity Understanding in Smart Homes: In homecare settings, action recognition assists in assessing the well-being of elderly individuals, ensuring timely assistance, and detecting unusual events or emergencies. Smart home systems leverage action recognition to understand and respond to residents' especially old people's activities. It contributes to

home automation by identifying actions such as cooking, sleeping, or exercising, allowing for context-aware smart home functionalities.

Expressive Animation and Entertainment: The technology is employed in entertainment for creating expressive animations that mimic human movements accurately. This includes applications in character animation, avatar control, and the film and animation industry.

Sports Analysis: In sports analysis, video action recognition is employed to analyze athletes' movements, track player positions, and evaluate performance. It provides coaches and analysts with valuable insights for strategic planning, player development, and performance optimization. These diverse applications underscore the versatility and significance of skeleton-based action recognition in enhancing various aspects of human life, from healthcare and rehabilitation to entertainment and security.

1.3 Motivation

The dynamic nature of human action recognition, its interdisciplinary relevance, and its potential for real-world impact make it a compelling and important area of research and development. Targeting action recognition, recent works have made promising progress by adopting spatio-temporal Convolutional Neural Networks (CNNs) [77, 18, 63, 51, 50, 140, 96, 182] or Transformers [4] to effectively extract features from RGB videos and optical flows [79, 53]. Moreover, skeleton-based human action recognition methods have also achieved promising results as they rely on 2D or 3D positions of human key joints only. They are able to filter out noise caused, for instance, by background clutter or changing light conditions and to focus on the action being performed. However, in real-world applications (e.g., Toyota Smarthome, UAV-Human, Kinetics), where human-centric activities are often complex and fine-grained, there are still many scientific challenges to deal with, such as model generalizability, fine-grained motions, long-term dependencies, interpretable representation, scalability, real-time processing, multi-modal inputs, etc. In this context, we focus on the settings of both action classification and segmentation tasks and we tackle the mentioned specific challenges on real-world videos by leveraging RGB videos and skeleton sequences.

To do so, we provide scientific contributions by proposing novel approaches in three directions: improving skeleton-based action representation and recognition models, improving RGB-based action representation models, and designing generic models for video-skeleton

combined learning. In the following sections, we introduce the target scientific challenges and our contributions.

1.4 Scientific Challenges

In this thesis, we address real-world challenges by proposing novel approaches towards understanding human activities in videos. Our proposed models include supervised and self-supervised learning models for action representation learning. These models aim to improve the accuracy and interpretability of human action classification and action segmentation. In order to provide insights into the underlying structure and dynamics of human actions in videos, we introduce here the main challenges for learning action representations.

Model Generalizability: Adapting action representations learned from one domain or dataset to another domain (domain adaptation) or to specific tasks (transfer-learning) is challenging due to domain shifts and differences in action distributions. Action recognition methods based on skeleton data have recently witnessed increasing attention and progress. Such methods show advantages in learning effective motion features compared to using RGB data. Hence, making good use of skeleton data could help to learn clear motion features of a video. However, the model generalizability of such methods is limited. State-of-the-art approaches [197, 145, 105, 22] adopting Graph Convolutional networks (GCNs) can effectively extract features on human skeletons relying only on the pre-defined human topology but they have difficulties to generalize across domains, especially with different human topological structures. To address this, we propose a unified framework (see Chapter 3) including a topology-free model and a large-scale pre-training dataset to significantly improve the generalizability of the skeleton-based action representations.

Effective Latent Action Representation: To better learn the mapping of the video and its action, clear motion coded in the latent video representation that is robust to occlusions and viewpoints is important to be disentangled and used for action classifiers. In this context, we aim to improve the skeleton action representation ability in three aspects: occlusion robustness, viewpoints robustness, and compositionality.

1. **Occlusion-robust representation:** The challenge has to do with occlusion in real-world videos that hinder the visibility of all joints. Such occlusions impede the representation of such scenes by models that have been trained on full-body pose data,

obtained in laboratory conditions with specific sensors. We propose an occlusion-robust skeleton representation learning approach leveraging sub-graph contrastive learning [201].

2. **View-invariant representation:** Current approaches for skeleton action representation learning often focus on constrained scenarios, where videos and skeleton data are recorded in laboratory settings. When dealing with estimated skeleton data in real-world videos, such methods perform poorly due to the large variations across subjects and camera viewpoints. To address this, we introduce a skeleton representation learning approach that learns the consistency of multi-view actions generated from our novel generation module (see Chapter 4).

3. **Complex composable representation:** action segmentation requires recognizing composable actions in untrimmed videos. Current approaches decouple this problem by first extracting local visual features from skeleton sequences and then processing them by a temporal model to classify frame-wise actions. However, their performances remain limited as the visual features cannot sufficiently express composable actions. In Chapter 4, we present a joint generative and contrastive model to learn skeleton action representation with high compositionality.

Time-aware Action Representation: In the RGB video-based action representation learning domain, self-supervised approaches aimed at maximizing similarities between different temporal segments of one video, in order to enforce feature persistence over time. This leads to a loss of pertinent information related to temporal relationships, making actions such as 'enter' and 'leave' to be indistinguishable. We claim that the representation of subtle and interaction motions aware of time variance is important to capture motion variance, and we present an effective solution in Chapter 5.

Multi-modal Action Representation: Modeling and combining data in different structures (*e.g.*, RGB, skeleton, text, audio) by an effective and unified model could broaden the view of the model and further improve the action recognition accuracy of a single model. We have studied the learning of effective motion features from single skeleton data or RGB data, and we further deduce that the combination of both is important to model fine-grained details to distinguish similar actions. In this thesis, we explore the way of combining skeleton motion with visual RGB features [37] and with semantic text features (see Chapter 6).

1.5 Thesis Outline

In this thesis, we first design a unified framework including a skeleton estimation model, a skeleton processing model and a pre-training dataset for skeleton-based action classification. Then we propose two self-supervised skeleton action representation learning frameworks based on generated data. Subsequently, we introduce a self-supervised action representation for RGB videos. Finally, we present a Visual-Motion-Text multi-modal pre-training approach for action recognition. These contributions are organized in the following chapters.

Chapter 2 revisits literature with a particular focus on action representation learning and its downstream tasks: action classification, and action segmentation.

In Chapter 3, we introduce UNIK, a novel skeleton-based action recognition method that is not only effective to learn spatio-temporal features on human skeleton sequences but also able to generalize across datasets. This is achieved by learning an optimal dependency matrix from the uniform distribution based on a multi-head attention mechanism. The high-quality skeletons can be obtained by our proposed SSTA-PRS human pose refinement system, which is also presented in this chapter. Subsequently, we introduce the Posetics dataset. To study the cross-domain generalizability of skeleton-based action recognition in real-world videos, we re-evaluate state-of-the-art approaches, as well as the proposed UNIK, in light of a novel Posetics dataset. This dataset is created from Kinetics-400 videos by estimating, refining, and filtering poses. We provide an analysis of performance improvement on smaller benchmark datasets after pre-training on Posetics for the action classification task.

In Chapter 4, we focus on latent skeleton motion learning and interpretation. We try to know the content coded in the features, interesting for action recognition, generation model for data augmentation for human-focused activity understanding. Specifically, we introduce (i) Latent Action Composition (LAC), a novel self-supervised framework aiming at learning from synthesized composable motions for skeleton-based action segmentation. LAC is composed of a novel generation module towards synthesizing new sequences based on an learnable orthogonal basis. We also introduce (ii) ViA, a novel View-Invariant Action representation learning framework. ViA leverages motion retargeting between different human performers as a pretext task, in order to disentangle the latent action-specific 'Motion' features on top of the visual representation of a 2D or 3D skeleton sequence. Such 'Motion' features are invariant to skeleton geometry and camera view and they allow ViA to facilitate both cross-subject and cross-view action classification tasks.

Chapter 5 presents Latent Time Navigation (LTN), a time-parameterized contrastive learning strategy that is streamlined to capture fine-grained motions. Specifically, we maxi-

mize the representation similarity between different video segments from one video, while maintaining their representations *time-aware* along a subspace of the latent representation code including an orthogonal basis to represent temporal changes.

In Chapter 6, we extend the Posetics dataset as PoseCap-1M and introduce Transferable Motion Representation (T-MOR) learning approach, to capture and analyze fine-grained human actions leveraging human skeleton motion data. This model not only focuses on visual-level learning with natural language supervision, but also learns the subtle human motion dynamics crucial for complex, human-centric action recognition using multi-modal contrastive learning.

Chapter 7 discusses future work and concludes this thesis.

1.6 Contributions

We list all publication contributions, as well as software that we developed in the course of this thesis. We will detail the contributions of five publications in Chapters 3-5 and additionally introduce the work in progress in Chapter 6.

1.6.1 Publications

- Di Yang, Rui Dai, Yaohui Wang, Rupayan Mallick, Luca Minciullo, Gianpiero Francesca, Francois Bremond. Selective Spatio-Temporal Aggregation Based Pose Refinement System: Towards Understanding Human Activities in Real-World Videos. *In Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2021.* [200] (Chapter 3)

- Di Yang*, Yaohui Wang*, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, Francois Bremond. UNIK: A Unified Framework for Real-world Skeleton-based Action Recognition. *In Proc. British Machine Vision Conference (BMVC) 2021 (Oral presentation).* [202] (Chapter 3)

- Di Yang, Yaohui Wang, Antitza Dantcheva, Quan Kong, Lorenzo Garattoni, Gianpiero Francesca, Francois Bremond. LAC - Latent Action Composition for Skeleton-based Action Segmentation. *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV) 2023.* [204] (Chapter 4)

- Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, Francois Bremond. View-invariant Skeleton Action Representation Learning via Self-

supervised Motion Retargeting. *International Journal of Computer Vision (IJCV) 2024.* [203] (Chapter 4)

- Di Yang, Yaohui Wang, Quan Kong, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, Francois Bremond. Self-supervised Spatio-temporal Representation Learning via Latent Time Navigation. *In Proc. AAAI Conference on Artificial Intelligence (AAAI) 2023.* [205] (Chapter 5)

- Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, Francois Bremond. Self-supervised Video Pose Representation Learning for Occlusion-robust Action Recognition. *In Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FG) 2021. (Oral presentation)* [201]

- Di Yang, Mahmoud Ali, Gianpiero Francesca, Francois Bremond. From Skeletons to Transferable Action Model with Multi-modal Representations. *PrePrint 2024.* (Chapter 6)

- Yaohui Wang, Di Yang, Francois Bremond, Antitza Dantcheva. Latent Image Animator: Learning to Animate Image via Latent Space Navigation. *In Proc. International Conference on Learning Representations (ICLR) 2022.* [189]

- Srijan Das, Rui Dai, Di Yang, Francois Bremond. VPN++: Rethinking Video-Pose embeddings for understanding Activities of Daily Living. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) 2021.* [37]

- Valeriya Strizhkova, Yaohui Wang, David Anghelone, Di Yang, Antitza Dantcheva, Francois Bremond. Emotion Editing in Head Reenactment Videos using Latent Space Manipulation. *In Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FG) 2021.* [161]

1.6.2 Patents and Software Contributions

Patents:

- Computer-implemented Method for Pre-training A Model to Recognize A Graph-represented Pattern in An Input. EP Patent *(Applied in 2021). Application number: EP21305961.1.* (Chapter 3)

- Motion Representation Calculation Method and System, Training Method, Computer Program, Readable Medium and System. EP Patent *(Applied in 2022). Application number: EP22305979.1*. (Chapter 4)
- Method and System for Training An Encoder Model. EP Patent *(Applied in 2023). Application number: EP23305147.1*. (Chapter 5)

Software:

- Selective Spatio-temporal Aggregation based Pose Refinement System. [200] (Chapter 3). Code is available in https://github.com/walker-a11y/SSTA-PRS.
- Unified framework for skeleton-based action representation learning, classification, segmentation and generation. [202] (Chapter 3). Code is available in https://github.com/walker1126/UNIK.
- Latent Action Composition for skeleton-based action segmentation. [204] (Chapter 4). Code is available in https://github.com/walker1126/Latent_Action_Composition.

Chapter 2

Literature Review

In this chapter, we revisit the literature related to the thesis topics, *i.e.*, action classification, action segmentation and action representation learning.

2.1 Action Recognition Tasks and Evaluation Datasets

Action recognition includes three main tasks: action classification, action segmentation, and action representation learning. Action classification task corresponds to learn a mapping from an input trimmed video clip to an action category (*e.g.*, walking, drinking). Action segmentation is a more challenging task that focuses on predicting per-frame action categories for an untrimmed videos. The given videos could have several different actions in different timestamps and there could be multiple actions occurring at the same time (composable actions). Hence, action segmentation is a frame-wise multi-label action classification task. Different from both two tasks, action representation learning is a pre-training stage prior to action classification and segmentation tasks (see Fig. 2.1). Both action classification and segmentation models rely on a video encoder to embed videos into a low-dimensional vector, namely video representation, to represent the compact information (*i.e.*, features) related to the human actions. The video representation is then fed to a classifier to predict the action category. Therefore, learning a good video encoder that has a strong representation ability on a large-scale pre-training dataset can benefit the target downstream action classification and segmentation tasks on smaller benchmarks by transfer-learning. The action representation could be obtained by fully supervised learning (*i.e.*, pre-training a video encoder on a large-scale dataset using action labels) or self-supervised learning(*i.e.*, pre-training a video encoder

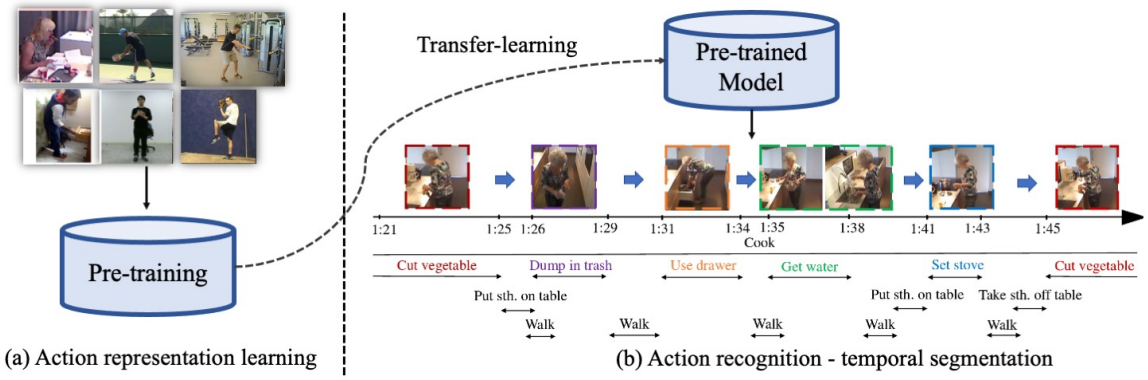

Fig. 2.1 **Action representation learning and downstream task.** (a) Action representation learning methods aim at pre-training a generic video encoder with a large number of video clips. Such video encoder can effectively represent human actions in videos and can be transferred to improve downstream action recognition tasks. For instance, in (b) the pre-trained video encoder is used to extract features of videos from an action segmentation dataset. The features are then learned to predict activity categories for each frame of the given video.

using pretext tasks without the need for action labels). In this chapter, we mainly focus on self-supervised approaches.

Related Datasets: To effectively benchmark action recognition models, numerous datasets spanning various action categories, video modalities, and complexities have been developed (see Tab. 2.1). Notable datasets include UCF101 [158], HMDB51 [87], Kinetics [18, 17], and ActivityNet [46]. Evaluation metrics like top-k accuracy, mean average precision (mAP), and frame-wise accuracy are standard for assessing model performance.

This thesis primarily focuses on human-centric action recognition, leveraging motion features. It emphasizes skeleton-based datasets, including **NTU-RGB+D 60** [143], **Toyota Smarthome** [36], **Penn Action** [213], **Kinetics-Skeleton** [197], **UAV-Human** [101], **PKU-MMD** [27], and **Toyota Smarthome Untrimmed (TSU)** [34]. These datasets offer a range of challenges, from controlled environments to real-world scenarios.

Additionally, we explore **Charades** [153], a dataset without skeleton data, and we transform it using estimated 2D skeleton data. We also use **Mixamo** [75] and **Kinetics-400** [18] to evaluate the learned representations. Furthermore, we assess our models on **UCF101** [158] and **HMDB51** [87] datasets for downstream action recognition tasks.

Dataset	Real-world	2D	3D	#Videos	#Actions	Compositional	Fine-grained	View	Type
Human3.6M [76]	×	✓	✓	209	15	No	No	Shooting	Daily living
N-UCLA [180]	×	×	✓	1,475	10	No	No	Shooting	Daily living
NTU-RGB+D 60 [143]	×	✓	✓	56,880	60	No	No	shooting	Daily living
NTU-RGB+D 120 [103]	×	✓	✓	114,480	120	No	No	shooting	Daily living
HMDB-51 [87]	✓	×	×	7,000	51	No	No	Shooting	YouTube
UCF-101 [158]	✓	×	×	13,320	101	No	No	Shooting	YouTube
SomethingSomething [56]	✓	×	×	220,847	174	No	Yes	Shooting	Object interaction
Epic-Kitchen [35]	✓	×	×	432	149	Yes	Yes	Egocentric	Kitchen
Penn Action [213]	✓	✓	×	2,326	15	No	No	shooting	Sport
UAV-Human [101]	✓	✓	×	21,224	155	No	No	Fish-eye	UAV
Toyota Smarthome [36]	✓	✓	✓	16,115	31	No	Yes	Monitoring	Daily living
PKU-MMD [27]	×	✓	✓	1,076	51	No	No	Shooting	Daily living
50-Salade [160]	✓	×	×	50	17	No	Yes	Shooting	Food
Breakfast [86]	✓	×	×	1,712	48	No	Yes	Shooting	Food
Assemble101 [142]	✓	×	×	4,321	101	No	Yes	Egocentric	Assembling
Charades [153]	✓	×	×	2,300	151	Yes	Yes	Shooting	Daily living
TSU [34]	✓	✓	✓	536	51	Yes	Yes	Monitoring	Daily living
Mixamo [75]	×	✓	✓	2,400	15	No	No	Free	Synthetics
Kinetics [18]	✓	×	×	400,000	400	No	No	Shooting	YouTube
HT100M [115]	✓	×	×	136M	23K	No	No	Shooting	Narrated video
Posetics [202] (This Thesis)	✓	✓	✓	142,000	320	No	No	Shooting	YouTube
PoseCap-1M (This Thesis)	✓	✓	✓	1,000,000	811	No	Yes	Shooting	Human-centric action

Table 2.1 A survey of recent datasets for human-centric and skeleton-based action classification (top), action segmentation (middle) and action representation learning (bottom).

In our exploration of transferability using human skeleton data, we employ pre-training and fine-tuning on real-world videos, a novel approach not previously applied to such datasets. We here provide the details of our main focused datasets:

Posetics [202] (presented in Chapter 3) is created on top of Kinetics-400 [18] videos. It contains 142,000 real-world video clips of 320 action classes with the corresponding 2D and 3D skeletons. We use the Posetics dataset to pre-train our action representation learning framework with skeleton data and we study the transfer-learning on skeleton-based action classification. We use Top-1 and Top-5 accuracy as evaluation metrics [202]. Recently we have extended Posetics as a larger version, namely **PoseCap-1M**, presented in Chapter 6 for better pre-training the generalizable skeleton action representation.

Toyota Smarthome [36] (Smarthome) is a real-world dataset for daily living action classification and contains 16,115 videos of 31 action classes. It provides RGB videos, 2D and 3D skeleton data [200]. As the provided 2D data is more robust for action recognition even for cross-view evaluation [200, 202], unless otherwise stated, we use 2D data for the experiments. For the evaluation, we report mean per-class accuracy following the cross-subject (CS) and cross-view (CV1 and CV2) evaluation protocols.

UAV-Human [101] contains 22,476 video sequences collected by a flying UAV including 2D skeleton data estimated by [48]. In this work, we use only 2D skeleton data and we follow Cross-subject (CS1 and CS2) evaluation protocols.

Penn Action [213] contains 2,326 video sequences of 15 different actions. In our work, we use 2D skeletons obtained by LCRNet++ [139] for experiments and we report Top-1 accuracy following the standard train-test split.

NTU-RGB+D 60 [143] consists of 56,880 sequences of high-quality 3D skeletons, captured by the Microsoft Kinect v2 sensor. We only use sequences of 3D skeletons in this work and we follow the cross-subject (CS) and cross-view (CV) evaluation protocol.

NTU-RGB+D 120 [103] extends the number of action classes and videos of NTU-RGB+D 60 to 120 classes 114,480 videos. We use 3D skeleton sequences and we follow the cross-subject (CS) and cross-set (CSet) evaluation protocols.

Toyota Smarthome Untrimmed (TSU) [34] is a large-scale real-world dataset for daily living action segmentation. It contains densely annotated long-term composite activities where up to 5 actions can happen at the same time in a given frame. We only use the provided 2D skeleton data [200] for the experiments. For evaluation, we report *per-frame* mAP (mean Average Precision) as [32, 31] following the cross-subject (CS) and cross-view (CV) evaluation protocols.

Charades [153] is a real-world dataset containing fine-grained activities similar to TSU. It provides only raw video clips without skeleton data. In this work, we use the 2D skeleton data (2D coordinates) estimated by the toolbox [200]. We report *per-frame* mAP on the localization setting of the dataset. For sake of reproducibility, we will release the estimated skeleton data on Charades.

PKU-MMD [27] is a basic untrimmed video dataset recorded in the laboratory setting. We use only the official 3D skeleton data. As this dataset is not densely labeled, we report the *event-based* mAP for fair comparisons by applying a post-processing [107] on the frame-level predictions to get the action boundaries.

Mixamo [75] is a 3D animation collection, which contains elementary actions and various dancing moves. Each of these motions may be applied to 71 distinct Statics, which share a human skeleton topology, but may differ in their body size and proportions. We use such a synthetic dataset for training and evaluating the generation module in Chapter 4.

Kinetics-400 [18] is a large-scale real-world dataset that contains about 240,000 video clips for 400 action classes collected from YouTube. To evaluate the representation learned on Kinetics-400, we transform the visual encoder to downstream tasks and we report results on Smarthome and the following two datasets:

UCF101 [158] and **HMDB51** [87]: UCF101 (UCF) contains 13,000 videos downloaded from YouTube spanning over 101 human action classes. HMDB51 (HMDB) contains 6,766

video clips from 51 action classes. Evaluation on both datasets is performed using average classification accuracy over three officially provided train/test splits.

2.2 Action Classification in Trimmed Videos

In this section, we present the state-of-the-art of current action classification approaches.

2.2.1 Objectives

Action classification involves identifying and categorizing human actions within video sequences. This review aims to provide a comprehensive analysis of the objectives pursued and of the evaluation metrics employed in the context of action classification, specifically within trimmed videos.

Action classification aims to automatically recognize and categorize human actions within video data. The objective can be broadly categorized into the following aspects:

1. Temporal Dynamics: Capturing the temporal evolution of actions is essential. Researchers strive to discern not only the actions themselves but also the duration of actions in trimmed videos.

2. Contextual Information: The interpretation of actions often benefits from considering the spatial and semantic context in which actions occur. This might involve identifying objects, scenes, or other contextual cues.

3. Viewpoint and Scale Invariance: Effective action classification should be invariant to variations in viewpoint and scale, ensuring robustness across different camera angles and distances.

In the learning stage, to train a network, we need to map the input data into prediction labels, where each training data has its corresponding ground truth label. In the task of action classification, full supervision employs the labels of the training set that contains the action category labels and the corresponding annotation for a video clip. The objective learning function is the general classification loss function *i.e.*, Cross Entropy Loss which is defined as:

$$\mathscr{L}_{CE} = -y\log(P). \tag{2.1}$$

Where P is the predicted score. This loss term is the main loss for video-level action classification in this thesis.

2.2.2 Evaluation Metrics

The effectiveness of action classification algorithms is gauged using various evaluation metrics. These metrics measure the performance of models against ground truth annotations. Commonly used metrics include:

Accuracy: The ratio of correctly classified actions to the total number of actions. Accuracy (see Eq. 2.2) provides an overall measure of the model correctness but may be imbalanced if classes are unevenly distributed.

$$Acc_c = \frac{TP^c + TN^c}{TP^c + TN^c + FP^c + FN^c} \tag{2.2}$$

Precision, Recall, and F-score: These metrics provide a more nuanced understanding by considering true positives, false positives, and false negatives. Precision (see Eq. 2.3) is the ratio of true positives to the total predicted positives, while recall (see Eq. 2.4) is the ratio of true positives to the total actual positives. F-score (see Eq. 2.5) combines precision and recall, providing a balance between the two.

$$P_c = \frac{TP^c}{TP^c + FP^c}, \tag{2.3}$$

$$R_c = \frac{TP^c}{TP^c + FN^c}, \tag{2.4}$$

$$F - score = \frac{2}{|C|} \sum_c \frac{P^c \times R_c}{P_c + R_c}. \tag{2.5}$$

Confusion Matrix: A confusion matrix tabulates the predicted labels against the actual labels, enabling an in-depth analysis of the model performance on individual classes.

Top-k Accuracy: In scenarios where multiple action classes might be plausible, top-k accuracy measures whether the correct label is among the top-k predictions.

Mean Average Precision (mAP): Particularly useful in action detection tasks, mAP evaluates the precision-recall curve and considers the average precision across different levels of recall:

$$mAP = \frac{1}{C} \sum_c P^c. \tag{2.6}$$

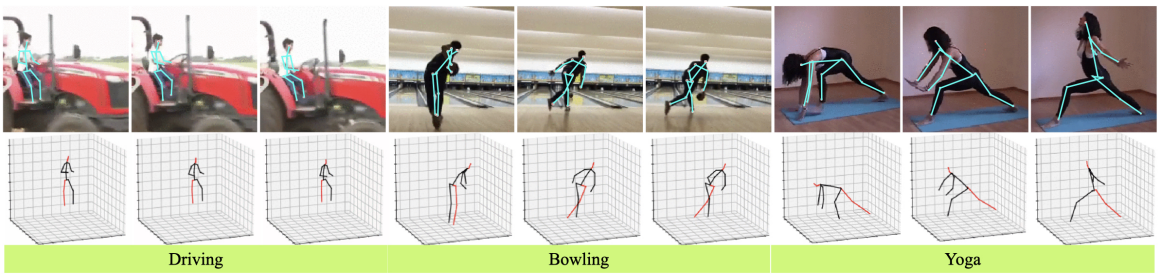

Fig. 2.2 Skeleton-based action classification in real-world.

The domain of action classification in trimmed videos has witnessed significant advancements in both objectives and evaluation metrics. As researchers continue to tackle challenges related to temporal dynamics, context, and fine-grained actions, the field is poised to enhance the accuracy and applicability of action classification algorithms in real-world scenarios.

2.2.3 Methodology

Human action recognition approaches can be mainly categorized into three types. (i) 3D-CNNs [77, 18, 63, 170, 51, 50, 140] and their variants [96, 182] have become the mainstream approach as their models can effectively extract spatio-temporal features for RGB videos and can be pre-trained on a large-scale dataset Kinetics [18] to facilitate transfer learning. (ii) Two-stream CNNs [79, 53] use two inputs of RGB and optical flow to separately model appearance and motion information in videos with a late fusion. Unlike RGB-based methods, (iii) skeleton-based approaches [197, 145, 156, 105] (see Fig. 2.2) can learn good video representation with fewer parameters and are more robust to changes in appearances, environments, and view-points. In this thesis, we mainly focus on skeleton-based approaches for better learning human motions.

2.3 Action Segmentation in Untrimmed Videos

Frame-wise action segmentation in untrimmed videos is a challenging task that involves detecting and localizing actions within a long video sequence at a fine-grained temporal level. Existing approaches often combine temporal modeling with classification or segmentation frameworks to address this task. However, accurately segmenting actions in the presence of complex background, occlusions, and temporal variations remains a difficult problem.

2.3.1 Objectives

Temporal Action Segmentation focuses on per-frame activity classification in untrimmed videos. The main challenge is how to model long-term relationships among various activities at different time steps. Specifically, action segmentation entails the automatic partitioning of untrimmed video sequences into distinct segments, each corresponding to a coherent action. The objectives of action segmentation include:

1. Temporal Localization: Precisely localizing the start and end times of each action segment within an untrimmed video is essential for generating accurate temporal boundaries.

2. Boundary Detection: Detecting action boundaries requires identifying significant changes in motion patterns, appearance, or contextual cues.

3. Multiple Action Handling: Effective segmentation methods should handle cases where multiple actions occur within a single video, ensuring that each action is correctly segmented.

As videos with dense action occurrence generally contain co-occurring actions, *i.e.*, multiple instances occurring at the same time, the video has been embedded into a sequence of frame-level or snippet-level features by the visual encoder [32, 31]. Detecting actions from such temporal features can be seen as multi-label classification task on top of these features. Hence, sequence-to-sequence action detection frameworks utilize the Binary Cross Entropy (BCE) loss described as:

$$\mathscr{L}_{BCE} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C} y_{tc}\log(P_{tc}), \tag{2.7}$$

where T is the number of frames or snippets, C is the number of action classes and P is the predicted score. In other words, after temporal modeling, we perform a binary classification for each frame or snippet feature and for every action class. This loss term is the main loss for frame-level action detection in this thesis.

2.3.2 Evaluation Metrics

The evaluation of action segmentation algorithms involves measuring their ability to accurately detect action boundaries and to segment actions in untrimmed videos. Commonly used metrics include:

Event-level Intersection over Union (IoU): IoU computes the overlap between the predicted segment and the ground truth segment. It is calculated as the ratio of the intersection to the union of the two segments.

Frame-level Average Precision (mAP): (FA) is commonly used in scenarios where multiple action instances are present within a video. It calculates the precision at different levels of recall and then averages them:

$$FA = \frac{\sum_c TP^c}{\sum_c N^c}. \tag{2.8}$$

2.3.3 Methodology

Temporal Action Segmentation focuses on per-frame activity classification in untrimmed videos. The main challenge is how to model long-term relationships among various activities at different time steps. Video-based action recognition methods aim to capture the spatio-temporal dynamics of human actions directly from video sequences. These methods typically employ 3D convolutional neural networks (CNNs) or two-stream networks that combine appearance and motion information. They have shown promising results in recognizing actions in trimmed videos, but their performance in untrimmed videos is still limited by challenges such as temporal localization and handling long-term dependencies.

Current methods mostly focus on directly using untrimmed RGB videos. Since untrimmed videos usually contain thousands of frames, training a single deep neural network directly on such videos is quite expensive. Hence, to solve this problem efficiently, previous works proposed to use a two-step method. In the first step, a pre-trained feature extractor (*e.g.*, I3D [18]) is applied on short sequences to extract corresponding visual features. In the second step, action segmentation is modeled as a sequence-to-sequence (seq2seq) task to translate extracted visual features into per-frame action labels. Temporal Convolution Networks (TCNs) [89, 32, 209] and Transformers [31] are generally applied in the second step due to their ability to capture long-term dependencies.

Recently, a few methods [30, 34] started to explore the use of skeletons in this task, in order to benefit from multi-modal information. In such methods, a pre-trained Graph Convolutional Network (GCN) such as AGCN [145] is used as a visual encoder to obtain skeleton features in the first step. However, unlike in pre-trained I3D which has strong generalizability across domains, pre-trained AGCN is not able to provide high-quality features due to its laboratory-based pre-trained dataset NTU-RGB+D [143]. We found that the performance significantly decreases when the pre-trained model is applied to more challenging real-world

untrimmed skeleton video datasets such as TSU [34] and Charades [153]. The main issue is that the pre-trained visual encoder does not have sufficient expressive power to extract complex action features, especially for composable actions that often occur in real-world videos. In this thesis (chapter 4), we propose novel and effective end-to-end skeleton-based approaches targeting such tasks without the need for the previous feature extraction stage.

2.4 Action Representation Learning in Videos

Video understanding and action recognition have been the focus of extensive research in the field of computer vision. Various approaches have been proposed to tackle the challenges associated with recognizing and understanding human actions in videos. These approaches range from handcrafted feature-based methods to deep learning-based models that can automatically learn discriminative representations from raw video data. Effective representation learning plays a crucial role in improving the performance of action recognition models. Deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been widely adopted for learning discriminative representations from videos. These models can capture spatial and temporal dependencies in videos, enabling more robust and informative representations for action recognition.

2.4.1 Objectives

Self-supervised video action representation learning aims to develop methods that extract meaningful and semantically rich representations from unlabeled video data. The key objectives include:

Temporal Context Encoding: Capturing temporal context is essential for understanding the sequential nature of actions within videos. Self-supervised methods seek to encode temporal dependencies in learned representations.

Semantics and Dynamics: Effective representations should capture both the semantic content of actions and the dynamic evolution of these actions over time.

Generalizability: Learned representations should be generalizable across different tasks, domains, and datasets, enabling transfer learning to downstream tasks.

2.4.2 Evaluation Metrics

Evaluating the quality of self-supervised video action representations involves measuring their effectiveness in subsequent action recognition or related tasks. In this thesis, the focused downstream tasks are action classification or frame-level action segmentation, so the evaluation metrics are the same as the target tasks (*e.g.*, Classification Accuracy, Frame-level mAP) presented in previous sections. However, unlike downstream tasks using the sole dataset for evaluation, the commonly used evaluation protocols for video representation learning are different.

Linear Evaluation: This involves training a linear classifier on top of the learned representations and evaluating its accuracy on action recognition or related tasks.

Fine-Tuning: Representations can be fine-tuned on labeled data for specific tasks, such as action recognition or action localization. Performance improvement after fine-tuning indicates the quality of the learned representations.

Transfer Learning: Assessing the representation transferability to different datasets or domains helps evaluate their generalizability.

2.4.3 Methodology

Contrastive learning and its variants [6, 15, 21, 59, 66, 71, 78, 167, 192] have established themselves as a pertinent direction for self-supervised representation learning for a number of tasks due to promising performances. Recent video representation learning methods [52, 73, 84] are inspired by image techniques. The objective of such techniques is to encourage representational invariances of different views (*i.e.*, positive pairs) of the same instance obtained by data augmentation, *e.g.*, random cropping [21, 192], rotation [116], while spreading representations of views from different instances (*i.e.*, negative pairs) apart. To further improve the representation capability, CMC [116] scaled contrastive learning to any number of views. MoCo [66] incorporated a dynamic dictionary with a queue and a moving-averaged encoder. To omit a large number of negative pairs, BYOL [59] and SwAV [15] were targeted to solely rely on positive pairs. DINO [16] completes the interpretation initiated in BYOL of self-supervised learning as a form of Mean Teacher self-distillation with no labels. Besides contrastive model, Masked visual modeling [65] has been proposed to learn effective visual representations based on the simple pipeline of masking and reconstruction. Based on

this, VideoMAE [169] are shown data-efficient learners for self-supervised video pre-training. However, these methods miss a crucial time element when they are straightforward applied to the *video* domain with views generated by *image* data augmentation technique. In our work (in chapter 5), we adopt recent contrastive learning frameworks [59, 66] and we focus on learning time-aware representations for videos by latent spatio-temporal decomposition and navigation in the representation space.

In this chapter, we have provided an overview of the state-of-the-art in video understanding and action recognition. We discuss the importance of benchmark datasets and evaluation metrics for evaluating action recognition models. We also highlighted the advancements and challenges in skeleton-based action recognition, video-based action recognition, frame-wise action segmentation, video and action representation learning, as well as the emerging field of multi-modal action representation learning. These advancements lay the foundation for the approaches proposed in this thesis, which aim to address the limitations and improve the performance of human action recognition in real-world videos.

Chapter 3

Unified Framework for Learning Skeleton Action Representation

We present in this chapter a unified framework for real-world skeleton action recognition[1]. This framework includes a novel human pose (skeleton) refinement method, named SSTA-PRS, a novel generalizable skeleton action recognition model, named UNIK, and a skeleton pre-training dataset, named Posetics. SSTA-PRS incorporates a Selective Spatio-Temporal Aggregation mechanism, named SST-A, that refines and smooths the keypoint locations extracted by several expert pose estimators, and an effective weakly-supervised self-training framework which leverages the aggregated poses as pseudo ground-truth instead of handcrafted annotations for real-world pose estimation. UNIK is a novel topology-free skeleton-based action recognition method that is not only effective to learn spatio-temporal features on human skeleton sequences but also able to generalize across datasets. This is achieved by learning an optimal dependency matrix from the uniform distribution based on a multi-head attention mechanism. To study the cross-domain generalizability of skeleton-based action recognition in real-world videos, we re-evaluate state-of-the-art approaches as well as the proposed UNIK in light of a novel Posetics dataset. This dataset is created from Kinetics-400 videos by estimating, refining and filtering poses. We provide an analysis on how much performance improves on the smaller benchmark datasets after pre-training on Posetics for the action classification task. Extensive experiments are conducted for evaluating not only the upstream pose refinement but also the downstream action recognition performance. We show that the proposed UNIK, with pre-training on Posetics, generalizes well and outperforms state-of-the-art when transferred onto four target action classification

[1]Project website: https://github.com/walker1126/UNIK/

datasets: Toyota Smarthome, Penn Action, NTU-RGB+D 60 and NTU-RGB+D 120. The works in this chapter have been published in IEEE/CVF Winter Conference on Applications of Computer Vision(WACV) 2021 [200] and in British Machine Vision Conference (BMVC) 2021 [202].

3.1 Introduction

Action recognition based on skeleton data has recently witnessed increasing attention and progress, as skeleton-based human action recognition methods rely on 2D or 3D positions of human key joints only, they are able to filter out noise caused, for instance, by background clutter, changing light conditions, and to focus on the action being performed [172, 212, 165, 216, 194, 94, 13, 197, 98, 145, 54, 146, 126, 144, 156, 105, 22, 99]. However, most of recent explorations are based on accurate 3D human skeletons obtained from RGBD sensors [143] and there are still challenges in skeleton estimation and skeleton model generalization, when dealing with real-world videos. Specifically, state-of-the-art pose estimators [139, 137, 14, 48, 85, 179, 60] struggle in obtaining high-quality 2D or 3D pose data due to occlusion, truncation and low-resolution in real-world un-annotated videos. Moreover, state-of-the-art skeleton-based action recognition models adopting Graph Convolutional networks (GCNs) have difficulties to generalize across domains, especially with different human topological structures. Motivated by the above problems, we focus on learning skeleton-based video representations for action recognition. In this chapter, we firstly propose a skeleton estimation and refinement framework that uses a Selective Spatio-Temporal Aggregation based Pose Refinement System, named **SSTA-PRS**, to extract high-quality 2D skeletons from un-annotated real-world videos. Secondly, we propose a unified skeleton-based action recognition model, named **UNIK**, with a large-scale pre-training dataset, named **Posetics**, for the generalization to real-world videos.

Skeleton Estimation in Real-world: To deal with the absence of keypoints (*i.e.*, joints) due to occlusion, truncation and low-resolution in real-world pose estimation (as shown in Fig. 3.1), we construct a multi-expert pose estimation system to predict improved 2D poses. It is fine-tuned with the pseudo ground-truth 2D pose generated by a novel Selective Spatio-Temporal Aggregation (SST-A) which integrates the pose proposals computed from several existing expert pose estimators. In this work, we select LCRNet++ [139], OpenPose [14] and AlphaPose [48] as the experts.

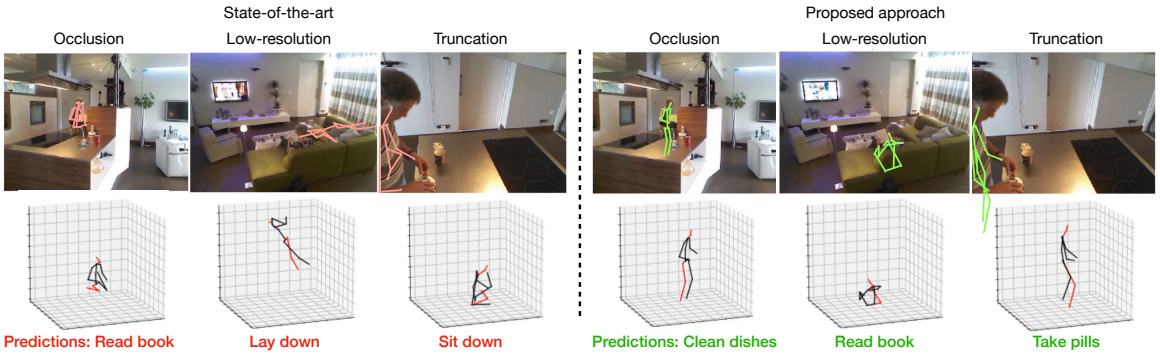

Fig. 3.1 **Skeleton-based action recognition** on Toyota Smarthome with poses (Left) extracted by AlphaPose [48] (left), LCRNet++ [139] (middle), OpenPose [14] (right) and high-quality poses (Right) obtained by the proposed pose refinement system. The action predictions using refined poses with the same action recognition system become more accurate. (3D reconstructions are from VideoPose [125] over 2D)

Skeleton Modeling for Action Recognition: Subsequently, we focus on designing an effective and generic skeleton model to extract action features on top of such refined skeleton sequence of the video for action recognition. To process skeleton sequences, recent approaches, namely Graph Convolutional Networks (GCNs) [197], models human joints, as well as their natural connections (*i.e.*, bones) in skeleton spatio-temporal graphs to carry both spatial and temporal inferences. Consequently, several successors, namely Adaptive GCNs (AGCNs), with optimized graph construction strategies to extract multi-scale structural features and long-range dependencies have been proposed and have shown encouraging results. Promising examples are graph convolutions with learnable *adjacency matrix* [145], higher-order polynomials of *adjacency matrix* [98] and separate multi-scale subsets of *adjacency matrix* [105]. All these *adjacency matrices* are manually pre-defined to represent the relationships between joints according to human topology. Nevertheless, compared to RGB-based methods such as spatio-temporal Convolutional Neural Networks (CNNs) [18, 64] that are pre-trained on Kinetics [18] to boost accuracy in downstream datasets and tasks, GCN-based models are limited because they are always trained individually on the target dataset (often small) from scratch. Our insight is that the generalization abilities of these approaches are hindered by the need for different adaptive adjacency matrices when different topological human structures are used (*e.g.*, joints number, joints order, bones), as in the case of the three datasets of Fig. 3.2. However, we note that such adaptive sparse *adjacency matrices* are transformed into fully dense matrices in deeper layers in order to capture long-

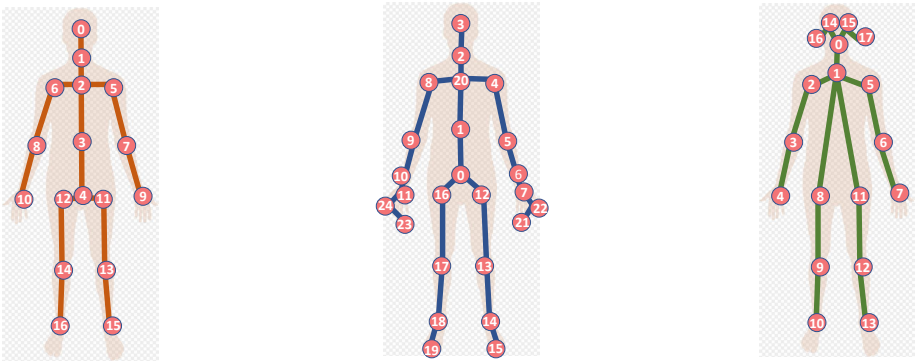

Fig. 3.2 **Human joint labels** of three datasets: Toyota Smarthome (left), NTU-RGB+D (middle), and Kinetics-Skeleton (right). We note the different numbers, orders and locations of joints.

range dependencies between joints. This new structure contradicts the initial and original topological skeleton structure.

Based on these considerations and as the human-intrinsic graph representation is deeply modified during training, we hypothesize that there should be a more optimized and generic initialization strategy that can replace the *adjacency matrix*. To validate this hypothesis, we introduce UNIK, a novel unified framework for skeleton-based action recognition. In UNIK, the *adjacency matrix* is initialized into a uniformly distributed *dependency matrix* where each element represents the dependency weight between the corresponding pair of joints. Subsequently, a multi-head aggregation is performed to learn and aggregate multiple dependency matrices by different attention maps. This mechanism jointly leverages information from several representation sub-spaces at different positions of the *dependency matrix* to effectively learn the spatio-temporal features on skeletons. The proposed UNIK does not rely on any topology related to the human skeleton, which makes it much easier to transfer onto other skeleton datasets. This opens up a great design space to further improve the recognition performance by transferring a model pre-trained on a sufficiently large dataset.

Skeleton Pre-training Dataset: Another reason for poor generalization abilities is that many skeleton datasets have been captured in lab environments with RGBD sensors (*e.g.*, NTU-RGB+D [143, 103]). Then, the action recognition accuracy significantly decreases when the pre-trained models on the sensor data are transferred to the real-world videos, where skeleton data encounter a number of occlusions and truncations of the body. To address

this, we create the Posetics dataset by estimating and refining poses, as well as filtering, purifying and categorizing videos and annotations based on the real-world Kinetics-400 [18] dataset. To this aim, we apply multi-expert pose estimators [14, 48, 139] and a refinement algorithm [200]. Our experimental analysis confirms that pre-training on Posetics improves state-of-the-art skeleton-based action recognition methods, when transferred and fine-tuned on all evaluated datasets [36, 213, 143, 103].

Contributions: In summary, the contributions of this chapter are: (i) We propose a novel Selective Spatio-Temporal Aggregation mechanism (SST-A), that integrates the advantage of several expert pose estimation systems in both spatial and temporal domains, and introduce a confidence metric C to evaluate the quality of the aggregated poses. (ii) We present a weakly-supervised self-training Pose Refinement System (SSAT-PRS) based on LCRNet++ [139] using pseudo-ground truth poses, generated by our SST-A mechanism instead of using hand-crafted pose annotations. (iii) we go beyond GCN-based architectures by proposing UNIK with a novel design strategy by adopting dependency matrices and a multi-head attention mechanism for skeleton-based action recognition. (iv) We revisit real-world skeleton-based action recognition focusing on cross-domain transfer learning. The study is conducted on four target datasets with pre-training on Posetics, a novel and large-scale action classification dataset that features higher quality skeleton detections based on Kinetics-400. (v) We demonstrate that pre-training UNIK on Posetics and fine-tuning it on the target real-world datasets (*e.g.*, Toyota Smarthome [36] and Penn Action [213]) can be a generic and effective methodology for skeleton-based action classification.

3.2 Related Work

Human Pose Estimation in Real-World: Most state-of-the-art approaches for 2D human pose estimation employ 2D CNNs architectures for a single image in a strongly-supervised setting [121, 14, 179, 48, 67, 25, 85]. For 3D pose estimation, [139, 118] focus on end-to-end reconstruction by directly estimating 3D poses from RGB images without intermediate supervision. [214] applies GCNs for regression tasks, especially 2D to 3D human pose regression. [125] demonstrates that 3D poses in video can be effectively estimated with a fully convolutional model based on dilated TCNs over 2D keypoint sequences. Among these methods, [139, 121, 118, 48, 67, 179] have first to incorporate a person detector, followed by the estimation of the joints and then the computation of the pose for each person. These approaches give full-body prediction once the people is detected, but the detection speed

slows down with the increase of the number of people present in the image. [14, 25, 85, 108] are bottom-up approaches which detect all joints in the image using heatmaps that estimate the probability of each pixel to correspond to a particular joint, followed by associating body parts belonging to distinct individuals. These approaches cannot always provide with the none-visible body parts for each individual due to occlusions and truncations.

By annotating poses in the real-world, approaches [139, 14, 48, 138] are becoming more robust to occlusion and they can provide us with pre-trained pose estimators, so that we can extract skeleton data from real-world videos without expensive handcraft annotations. In particular, LCRNet++ [139] is an attractive pose estimator, which leverages a Faster R-CNN [137] like architecture with a CNNs backbone. A Region Proposal Network extracts candidate boxes around humans. To deal with occlusions and truncation, LCRNet++ proposes 'anchor-poses' for pose classes instead of object classes: these key poses typically correspond to a person standing, sitting, etc. Bottom-up method OpenPose [14] proposes an alternative approach by regressing affinities between joints (*i.e.*the direction of the bones), together with the heatmaps. AlphaPose [48] improves the performance of top-down pose estimation algorithms by detecting accurate human poses even with inaccurate bounding boxes. Closer to our work, Rockwell et al. [138] propose an effective self-training framework that adapts human 3D mesh recovery systems to consumer videos. They focus on recovering from occlusions and truncations, but they do not have solutions to tackle low-resolution images and the instability of the extracted 3D meshes along time. In our work, we combine the advantages of the three expert pose estimators [139, 14, 48] by spatio-temporal aggregating their results and getting a more accurate pose than using only one expert.

Human Action Recognition: Human action recognition approaches can be mainly categorized into three types. (i) 3D-CNNs [77, 18, 63, 170, 51, 50, 140] and their variants [96, 182] have become the mainstream approach as the models can effectively extract spatio-temporal features for RGB videos and can be pre-trained on a large-scale dataset Kinetics [18] to facilitate transfer learning. (ii) Two-stream CNNs [79, 53] use two inputs of RGB and optical flow to separately model appearance and motion information in videos with a late fusion. Unlike RGB-based methods, (iii) skeleton-based approaches [197, 145, 156, 105] can learn a good video representation with less amounts of parameters and are more robust to changes in appearances, environments, and view-points. In this work, we specifically focus on improving the skeleton-based action recognition performance and the model generalization ability.

Skeleton-Based Action Recognition: Early skeleton-based approaches using Recurrent Neural Networks (RNNs) [212, 165, 216, 155, 194] or Temporal Convolutional Networks (TCNs) [82] were proposed due to their high representation capacity. However, these approaches ignore the spatial semantic connectivity of the human body. Subsequently, [94, 13, 212] proposed to map the skeleton as a pseudo-image (*i.e.*, in a 2D grid structure to represent the spatial-temporal features) based on manually designed transformation rules and to leverage 2D CNNs to process the spatio-temporal local dependencies within the skeleton sequence by considering a partial human-intrinsic connectivity. ST-GCN [197] used spatial graph convolutions along with interleaving temporal convolutions for skeleton-based action recognition. This work considered the topology of the human skeleton, however it ignored the important long-range dependencies between the joints. In contrast, recent AGCN-based approaches [98, 145, 54, 144, 126, 146, 156, 105] shown a significant improvement in performance, thanks to the benefit of improved representation of human skeleton topology to process long-range dependencies for action recognition. Specifically, 2s-AGCN [145] introduced an adaptive graph convolutional network to adaptively learn the topology of the graph with self-attention, which was found to be beneficial in action recognition and hierarchical structure of GCNs. Associated extension, MS-AAGCN [146] incorporated multi-stream adaptive graph convolutional networks that used attention modules and 4-stream ensemble based on 2s-AGCN [145]. These approaches primarily focused on spatial modeling. Consequently, MS-G3D Net [105] presented a unified approach for capturing complex joint correlations directly across space and time. However, the accuracy depends on the scale of the temporal segments, which should be carefully tuned for different datasets, preventing transfer learning. Thus, these previous approaches [145, 146, 105] learn adaptive adjacency matrices from the sub-optimal initialized human topology. In contrast, our work proposes an optimized and unified dependency matrix that can be learned from a *uniform distribution* by a multi-head attention process without the constraint of human topology and a limited number of attention maps in order to improve performance, as well as generalization capacity for skeleton-based action recognition.

Model Generalization for Skeletons: Previous methods [197, 145, 146, 105] were only evaluated on the target datasets, trained from scratch without taking advantages of fine-tuning on a pre-trained model. To explore the transfer ability for action recognition using human skeleton, recent research [164, 97] proposed view-invariant 2D or 3D pose embedding algorithms with pre-training performed on lab datasets [76, 103] that do not correspond to real-world and thus these techniques struggle to improve the action recognition performance on

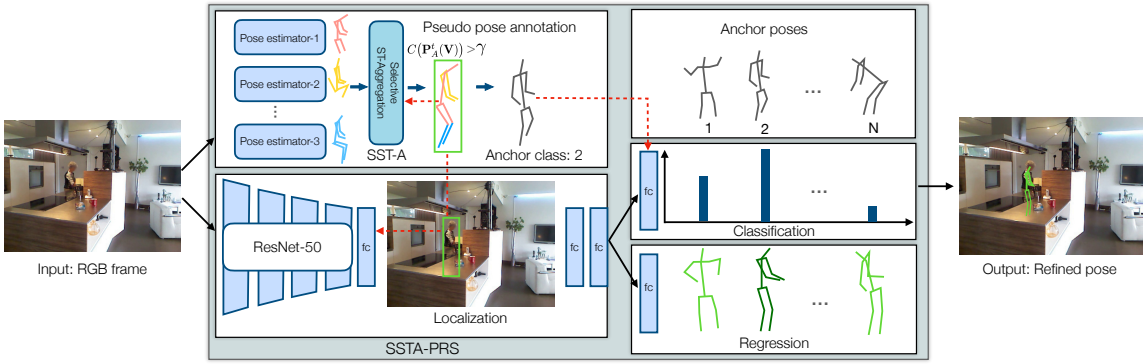

Fig. 3.3 **Overview of Pose-Refinement System (SSTA-PRS).** Given an RGB frame, a less noisy 2D pose $\mathbf{P}_A^t(\mathbf{V})$ and its confidence value C is computed by the Selective Spatio-Temporal Aggregation (SST-A) with the pose proposals obtained by several pose estimation systems and the previous aggregated pose. If the confidence is higher than the threshold γ, we are able to calculate the pseudo ground-truth bounding box and anchor class according to this improved pose to fine-tune the localization and classification branches of an LCRNet [139] architecture in a weakly-supervised setting. Finally, this refined pose estimation system is used to extract high-quality 2D poses in the real-world videos for the downstream action recognition task.

downstream tasks with large-scale real-world videos [36, 101]. To the best of our knowledge, we are the first to explore the skeleton-based pre-training and fine-tuning strategies for real-world videos.

3.3 SSTA-PRS: Refined Skeleton Acquisition Approach

The proposed framework includes SSTA-PRS, a human pose refinement system, UNIK, a generic skeleton model and Posetics, a large-scale skeleton pre-training dataset. In this section, we present SSTA-PRS, the weakly-supervised pose refinement approach.

3.3.1 Model Architecture

The overall architecture of the proposed method for pose refinement is shown in Fig. 3.3. Given an RGB frame, several pose proposals are obtained by multiple expert pose estimation systems [139, 137, 14, 48, 85, 179, 60] and the Selective Spatio-Temporal Aggregation mechanism (SST-A) computes an improved pose, which is more accurate, smoother, and more stable along time. With this aggregated pose, we compute a confidence metric to estimate its quality. Then, we select the aggregated poses with higher confidences than a

threshold and calculate the pseudo ground-truth bounding box and anchor class to fine-tune the localization and classification branches of an LCRNet [139] architecture. Finally, the refined pose estimation system is used to extract higher-quality poses in real-world videos.

3.3.2 Selective Spatio-Temporal Aggregation

Selective Spatio-Temporal Aggregation (SST-A) is the key component to deal with the **absence of keypoints** caused by occlusion, truncation and low-resolution, and with the **instability in time domain** due to pose estimation from a single frame. Our insight is that 1) bottom-up methods directly predict the keypoints through the heatmaps, however, they may miss joints that are none-visible due to occlusions or truncations because the number of each body part prediction may not correspond to the number of people in the image. 2) Top-down methods regress the coordinates of keypoints over the bounding box of the people. As long as people are detected, the keypoints of full-body can be predicted. But these methods may miss people in low-resolution images, resulting in missing all the joints of these people. According to the above analysis, by combining the results of both families of methods, we can reduce the number of missing joints and obtain more stable and higher-quality full-body keypoints. Therefore, we leverage multiple expert pose estimation systems, including methods from both families to extract poses for the same frame as several pose proposals, and then aggregate them to recover the missing keypoints. In this work, we select two top-down estimation systems LCRNet++ [139] and AlphaPose [48] and a bottom-up estimation system OpenPose [14] to provide the pose proposals, which are then combined into an improved pose sequence through our SST-A mechanism. Moreover, our pose sequence is extracted frame by frame with the estimators, so there is a certain lack of temporal continuity, resulting in a static joint shaking in the video. This problem is also an obstacle for the performance of action recognition. Hence, our SST-A also uses a temporal filtering mechanism to smooth the entire sequence by eliminating unstable values.

As shown in Fig. 3.4, we note all the N keypoints in one body as a set $\mathbf{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$, the frames as a set $F = \{0, 1, 2, ..., T\}$ and the position of one joint $\mathbf{v}$ $(\mathbf{v} \in \mathbf{V})$ in the frame t $(t \in F)$ estimated by the pose estimation system k_m $(k_m \in K = \{k_1, k_2, ..., k_M\})$ as $\mathbf{P}^t_{k_m}(\mathbf{v})$, noted that K is the ensemble of pose estimation systems. The final aggregated pose sequence of the body $\mathbf{V}$ is noted as $\mathbf{P}^F_A(\mathbf{V}) = \{\mathbf{P}^t_A(\mathbf{v}) | \mathbf{v} \in \mathbf{V}, t \in F\}$ and our aggregation system has two steps.

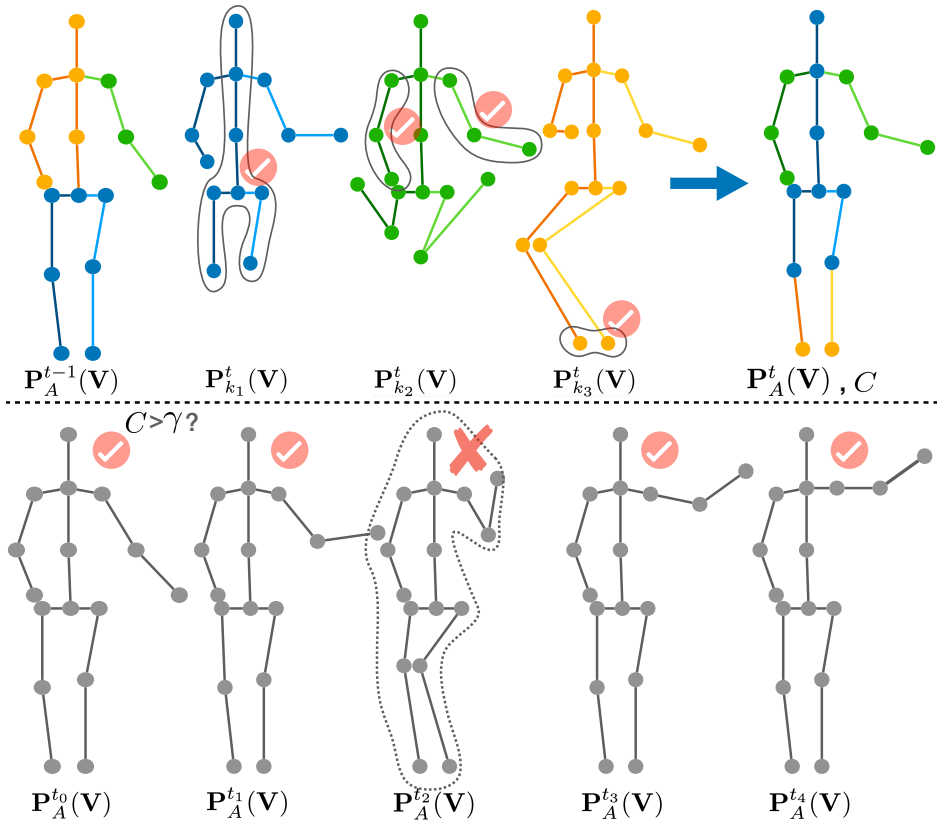

Fig. 3.4 **Two steps of SST-Aggregation**. 1) Aggregation in both spatial and temporal level (top). The pose $\mathbf{P}_A^t(\mathbf{V})$ of current frame is aggregated from the three pose proposals $\mathbf{P}_{k_1}^t(\mathbf{V})$ (blue), $\mathbf{P}_{k_2}^t(\mathbf{V})$ (green) and $\mathbf{P}_{k_3}^t(\mathbf{V})$ (yellow). 2) Selective temporal filter (bottom). The pose with a low confidence in the aggregated sequence will be discarded.

(1) Joint-level aggregation: Each keypoint of the pose is calculated from the prediction results of the three estimators [14, 139, 48] in the current frame $\mathbf{P}_{k_m}^t(\mathbf{v})$ and the aggregated result of the previous frame $\mathbf{P}_A^{t-1}(\mathbf{v})$. So this step is to select the closest keypoint to the same part aggregated in the previous frame. For the first frame, we select any one of the two closest keypoints obtained by pose estimators. Joint-level aggregation can be written as:

$$\mathbf{P}_A^t(\mathbf{v}) = \begin{cases} \mathbf{P}_{k_a}^t(\mathbf{v}), \text{ if } t = 0 \\ (k_a, k_b) = \underset{(k_i,k_j)\in K^2, i\neq j}{\operatorname{argmin}} \Big(D\big(\mathbf{P}_{k_i}^t(\mathbf{v}), \mathbf{P}_{k_j}^t(\mathbf{v})\big)\Big) \\ \mathbf{P}_{k_a}^t(\mathbf{v}), \text{ if } t > 0 \\ k_a = \underset{k_i\in K}{\operatorname{argmin}} \Big(D\big(\mathbf{P}_{k_i}^t(\mathbf{v}), \mathbf{P}_A^{t-1}(\mathbf{v})\big)\Big) \end{cases} \tag{3.1}$$

where D is the Euclidean distance between two key-points in the image, noted as:

$$D\big(\mathbf{P}_1(\mathbf{v}),\mathbf{P}_2(\mathbf{v})\big) = \sqrt{\big(\mathbf{P}_1(\mathbf{v})-\mathbf{P}_2(\mathbf{v})\big)^2} \tag{3.2}$$

(2) Body-level aggregation: Followed by the first step which can effectively solve the problem of missing keypoints, we define a **confidence metric** $C \in (0,1]$ that describes the likelihood that the aggregated pose is the real pose in order to further smooth the pose sequence. We believe that when the average similarity between the aggregated pose and the pose proposals is very high, the pose proposals are also very similar, indicating that the pose proposal itself is likely to be accurate, and the aggregation result will have a high confidence. This selective likelihood filter is written as (3.3), which is to discard the abnormal poses with a very low confidence in the whole sequence,

$$\mathbf{P}_A^t(\mathbf{V}) = \begin{cases} \mathbf{P}_A^t(\mathbf{V}), \text{ if } C\big(\mathbf{P}_A^t(\mathbf{V})\big) >= \gamma \\ \text{discard, if } C\big(\mathbf{P}_A^t(\mathbf{V})\big) < \gamma \end{cases} \tag{3.3}$$

where C (3.4) is defined to describe the confidence of this aggregated pose. (D_{normal} is the distance between the aggregated head and neck while offset $\varepsilon = 10^{-12}$ is to prevent errors in case of $D_{normal} = 0$)

$$C\big(\mathbf{P}_A^t(\mathbf{V})\big) = \exp\Big(-\frac{1}{NM}\sum_{\mathbf{V}}\sum_{K}\frac{D\big(\mathbf{P}_A^t(\mathbf{v}),\mathbf{P}_{k_m}^t(\mathbf{v})\big)}{D_{normal}+\varepsilon}\Big) \tag{3.4}$$

γ is a filtering parameter that represents a threshold. If the confidence of the pose in the current frame is lower than this threshold, it will be discarded from the sequence. After this two-step SST-A is completed, we obtain a higher-quality full-body skeleton sequence from a video, which can be effectively used as pseudo ground-truth pose for our self-training Pose Refinement system in Sec. 3.3.3.

3.3.3 Self-Training Pose Refinement System

SST-A can effectively integrate the advantages of [139, 14, 48]. However, this aggregation method may increase the workload in practice because we have to estimate the poses several times with different systems. Hence, we propose a self-training framework using the higher-quality 2D poses obtained from SST-A as supervised pseudo ground-truth, to refine one of the pose estimation models. Once the model is refined, the other models are not needed for inference. In fact, we only need to run SST-A on a small part of the dataset, and then it can be used for fine-tuning the network. As shown in Fig. 3.3, we build our pose

refinement model (SSTA-PRS) based on LCRNet++ [139] owing to its particularity of its three branches (localization, classification and regression), we do not have to provide truly accurate pose labels but only fine-tune the localization and classification branches with the pseudo ground-truth 2D poses.

Overview of SSTA-PRS architecture

As LCRNet++ [139], our SSTA-PRS framework also contains 4 main components. 1) **Localization:** it leverages a Faster R-CNN [137] like architecture with a ResNet-50 backbone [69]. Given an input image, a Region Proposal Network (RPN) [137] extracts candidate boxes around humans. 2) **Classification:** these regions are then classified into different 'anchor-poses' pre-defined by K-means clustering that typically correspond to a person standing, a person sitting, etc. In this paper, 'anchor-poses' are defined in 2D only, and the refinement occurs in this joint 2D pose space. 3) **Regression:** a class-specific regression is applied to estimate body joints in 2D. First, for each class of pose, we define offline the 'anchor-poses', computed as the center over all elements in the corresponding cluster. After fitting all the 2D anchor-poses into each of the candidate boxes, we perform class-specific regressions to deform these anchor-poses and match the actual 2D pose in each box. 4) **Post-processing:** for each individual, multiple pose candidates can overlap and produce valid predictions. These pose candidates are combined by pose proposal integration [139], taking into account their 2D overlap and classification scores. As the approach is holistic, it outputs full-body poses, even in case of occlusions or truncation by image boundaries.

Weakly-supervised training

We train this model with a weakly-supervised setting, which only refines the 2D localization and classification. The reason is that firstly, our pseudo pose annotations are not sufficiently accurate for regression while they are accurate enough for localization and classification. Secondly, the in-the-wild pre-trained model has good prediction performance when the localization and classification are correct. However, in low-resolution images, the bounding boxes of people are usually very difficult to search, which may result in no estimated keypoint on the body, or an error in the classification stage leading to inaccurate pose prediction. Therefore, if the classification and localization branches are correctly fine-tuned, the model should find the correct anchor class so that the final prediction can be more accurate.

Pseudo 2D pose ground-truth: it contains two parts, the bounding box of people and the anchor class. Both are calculated using the SST-A pose results. We take the maximum and minimum values of the pose in x and y directions as the boundary of the initial bounding box. We then expand the box by 10% as ground-truth for the localization branch, because the key-points do not correspond exactly to the boundary of the person. The class label of pose P, noted as $Class_P \in \{0,1,...,B\}$, is set by finding the closest 2D anchor-pose $Anchor_P$ according to the similarity S [139] between the oriented 2D poses centered at the left-top corner of bounding box: $Class_P = \text{argmin}_b S(Anchor_b, P)$. This label is used by the classification branch as pseudo ground-truth.

Loss function: our loss is the sum of the following two losses, described as:

$$L = L_{loc} + L_{classif} \tag{3.5}$$

The loss of the localization component is the loss of the region proposal network [137] (RPN):

$$L_{loc} = L_{RPN} \tag{3.6}$$

Same as [139], let u be the probability distribution estimated by SSTA-PRS, obtained by the fully connected layers of the classification branch after RoI pooling, followed by a Softmax function. The classification loss is defined using the standard cross entropy loss:

$$L_{classif}(u, Class_P) = -\log u(Class_P) \tag{3.7}$$

With the proposed SSTA-PRS, we can obtain high-quality skeletons from real-world videos and leverage such data for understanding human activities. In the following section, we present the skeleton modeling approach.

3.4 UNIK: Unified Skeleton Modeling

In this section, we present UNIK, the unified spatio-temporal dependencies learning network for skeleton-based action recognition.

3.4.1 Model Architecture

Skeleton Sequence Modeling: As shown in Fig. 3.6 (a), the sequence of the input skeletons is modeled by a 3D spatio-temporal matrix, noted as $\mathbf{f_{in}}$. For each frame, the 2D or 3D body joint coordinates are arranged in a vector within the spatial dimension in any order as long as the order is consistent with other frames in the same video. For the temporal dimension, the same body joints in two consecutive frames are connected. T, V, and C_{in} represent the length of the video, the number of joints of the skeleton in one frame, as well as the input channels (2D or 3D at the beginning and expanded within the building blocks), respectively. The input $\mathbf{f_{in}}$ and the output $\mathbf{f_{out}}$ for each building block (see 3.4.1) are represented by a matrix in $\mathbb{R}^{C_{in} \times T \times V}$ and a matrix in $\mathbb{R}^{C_{out} \times T \times V}$, respectively.

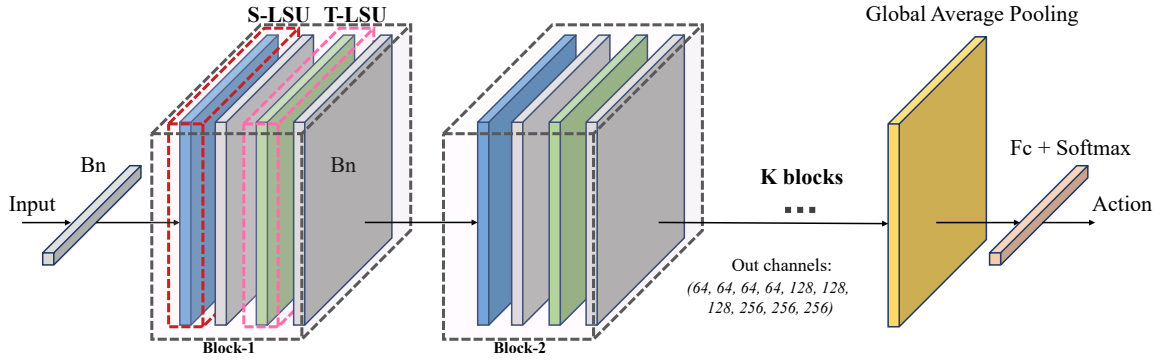

Fig. 3.5 **Overall architecture.** There are K blocks with a 1D Batch normalization layer at the beginning, a global average pooling layer and a fully connected classifier at the end. Each block contains a Spatial Long-short dependency Unit **(S-LSU)**, a Temporal Long-short dependency Unit **(T-LSU)** and two Batch normalization layers.

Overall Architecture: The overall architecture is composed of K building blocks (see Fig. 3.5). The key components of each block are the Spatial Long-short Dependency learning Unit (S-LSU), as well as the Temporal Long-short Dependency learning Unit (T-LSU) that extract both spatial and temporal multi-scale features on skeletons over a large receptive field. The building block ST-LS$_{\text{block}}$ is formulated as follows:

$$\mathbf{f_{out}} = \text{ST-LS}_{\text{block}}(\mathbf{f_{in}}) = \text{T-LSU}\left(\text{S-LSU}(\mathbf{f_{in}})\right). \tag{3.8}$$

S-LSU and T-LSU are followed by a 2D Batch normalization layer respectively. A 1D Batch normalization layer is added in the beginning for normalizing the flattened input data. Given a skeleton sequence, the modeled data is fed into the building blocks. After the last

block, global average pooling is performed to pool feature maps of different samples to the same size. Finally, the fully connected classifier outputs the prediction of the human action. The number of blocks K and the number of output channels should be adapted to the size of the training set, as a large network cannot be trained with a small dataset. However, in this work, we do not need to adjust K, as we propose to pre-train the model on a large, generic dataset (see 3.5). We set $K = 10$ with the number of output channels $64, 64, 64, 64, 128, 128, 128, 256, 256, 256$ (see Fig. 3.5). In order to stabilize the training and to ease the gradient propagation, a residual connection is added for each block.

Spatial Long-short Dependency Unit (S-LSU): To aggregate the information from a larger spatial-temporal receptive field, a sliding temporal window of size τ is set over the input matrix. At each step, the input $\mathbf{f_{in}}$ across τ frames in the window becomes a matrix in $\mathbb{R}^{C_{in} \times T \times \tau V}$. For the purpose of spatial modeling, we use a multi-head and residual based S-LSU (see Fig. 3.6 (b)) and formulated as follows:

$$\mathbf{f_{out}} = \text{S-LSU}(\mathbf{f_{in}}) = \sum_{i=1}^{N} \mathbf{E}_i \cdot \big(\mathbf{f_{in}} \times (\mathbf{W}_i + \mathbf{A}_i)\big), \tag{3.9}$$

where N represents the number of heads. $\mathbf{E}_i \in \mathbb{R}^{C_{out} \times C_{in} \times 1 \times 1}$ denotes the 2D convolutional weight matrix with 1×1 kernel size, which embeds the features from C_{in} to C_{out} by the dot product. $\mathbf{W}_i \in \mathbb{R}^{\tau V \times \tau V}$ is the "dependency matrix" mentioned in Sec. 3.1 to process the dependencies for every pair of spatial features. Inspired by [68], $\mathbf{W}_i$ is learnable and uniformly initialized as random values within bounds (Eq. 3.10).

$$\mathbf{W}_i = \text{Uniform}(-bound, bound), \quad \text{where} \quad bound = \sqrt{\frac{6}{(1+a^2)V}}, \tag{3.10}$$

where a denotes a constant indicating the negative slope of the rectifier [68]. In this work, we take $a = \sqrt{5}$ as the standard initialization strategy of the fully connected layers for $\mathbf{W}_i$, in order to efficiently reach the optimal dependencies.

Self-attention Mechanism: The matrix $\mathbf{A}_i$ in Eq. 3.9 represents the non-local self attention map that adapts the dependency matrix $\mathbf{W}_i$ dynamically to the target action. This adaptive attention map is learned end-to-end with the action label. In more details, given the input feature map $\mathbf{f_{in}} \in \mathbb{R}^{C_{in} \times T \times \tau V}$, we first embed it into the space $\mathbb{R}^{C_e \times T \times \tau V}$ by two convolutional layers with 1×1 kernel size. The convolutional weights are denoted as

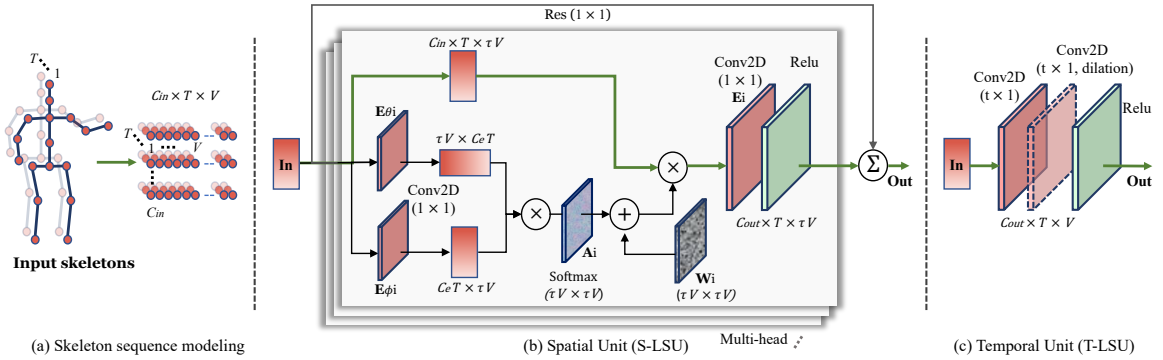

Fig. 3.6 **Unified Spatial-temporal Network.** (a) The input skeleton sequence is modeled into a matrix with C_{in} channels $\times$ T frames $\times$ V joints. (b) In each head of the S-LSU, the input data over a temporal sliding window (τ) is multiplied by a dependency matrix obtained from the unified, uniformly initialized $\mathbf{W_i}$ and the self-attention based $\mathbf{A_i}$. $\mathbf{E}_i$, $\mathbf{E}_{\theta i}$ and $\mathbf{E}_{\phi i}$ are for the channel embedding from C_{in} to C_{out}/C_e respectively by (1×1) convolutions. The final output is the sum of the outputs from all the heads. (c) The T-LSU is composed of convolutional layers with $(t \times 1)$ kernels . d denotes the dilation coefficient which can be different in each block.

$\mathbf{E}_{\theta i} \in \mathbb{R}^{C_e \times C_{in} \times 1 \times 1}$ and $\mathbf{E}_{\phi i} \in \mathbb{R}^{C_e \times C_{in} \times 1 \times 1}$, respectively. The two embedded feature maps are reshaped to $\tau V \times C_e T$ and $C_e T \times \tau V$ dimensions. They are then multiplied to obtain the attention map $\mathbf{A}_i \in \mathbb{R}^{\tau V \times \tau V}$, whose elements represent the attention weights between each two joints adapted to different actions. The value of the matrix is normalized to $0 \sim 1$ using a Softmax function. We can formulate $\mathbf{A}_i$ as:

$$\mathbf{A}_i = \text{Softmax}\left((\mathbf{E}_{\theta i}^{\mathbf{T}} \cdot \mathbf{f}_{\mathbf{in}}^{\mathbf{T}}) \times (\mathbf{E}_{\phi i} \cdot \mathbf{f}_{\mathbf{in}})\right). \tag{3.11}$$

Temporal Long-short Dependency Unit (T-LSU): For the temporal dimension, the video length is generally large. If we use the same method for spatial dimension, *i.e.*, setting the dependency weight to $T \times T$ weights for every pair of frames, it will consume too much calculation. Therefore, we leverage multiple 2D convolutional layers with kernels of different dilation coefficient d and temporal size t on the $C_{out} \times T \times N$ feature maps to learn the multi-scale long-short term dependencies (see Fig. 3.6 (c)). The T-LSU can be formulated as:

$$\mathbf{f_{out}} = \text{T-LSU}(\mathbf{f_{in}}) = \text{Conv}_{2D(t \times 1, d)}(\mathbf{f_{in}}). \tag{3.12}$$

Joint-bone Two-stream Fusion: Inspired by the two-stream methods [145, 144, 105], we use a two-stream framework where a separate model with identical architecture is trained

using the bone features initialized as vector differences of adjacent joints directed away from the body center. The Softmax scores from the joint and bone models are summed to obtain final prediction scores.

3.4.2 Design Strategy

In this section, we present our design strategy that goes beyond GCNs by using a generic dependency matrix $\mathbf{W}_i$ (see Eq. 3.9) and the attention mechanism $\mathbf{A}_i$ to model the relations between joints in our unified formulation.

Dependency Matrix: For many human actions, the natural connectivity between joints are not the most appropriate to be used to extract features on skeletons (*e.g.*, for "drinking", the connectivity between the head and the hand should be considered, but the original human topology does not include this connectivity). Hence, it is still an open question as to what kind of adjacency matrix can represent the optimal dependencies between joints for effective feature extraction. Recent works [98, 145, 105] aim at optimizing the adjacency matrices to increase the receptive field of graph convolutions, by higher-order polynomials to make distant neighbors reachable [98] or leveraging an attention mechanism to guide the learning process of the adjacency matrix [145, 105]. Specifically, they decompose the adjacency matrix into a certain number of subsets according to the distances between joints [105] or according to the orientation of joints with respect to the gravity (*i.e.*, body center) [145], so that each subset is learned individually by the self-attention. The learned feature maps are then aggregated together for action classification. However, the number of subsets is constrained by the body structure. Moreover, we note that the manually pre-defined subsets of the adjacency matrix with prior knowledge (*i.e.*, pre-defined body topology) are all sparse. At the initial learning stage, this spatial convolution relies on a graph-representation, while at the deeper stage, the relations coded within the adjacency matrix are no longer sparse and the joint connections are represented by a complete-graph, which corresponds to a fully connected layer in the narrow sense. Finally, the dependencies converge to a sparse representation again, which is locally optimal but completely different from the original topological connectivity of the human body (see Fig. 3.9). This motivates us, in this work, to revise the *adjacency matrix* by a generic *dependency matrix* that is prospectively initialized with a fully dense and uniform distribution (Eq. 3.10) to better reach the globally optimal representation.

Multi-head Aggregation: With our proposed initialization strategy, we can repeat the self-attention mechanism by leveraging multiple dependency matrices and sum the outputs to automatically aggregate the features focusing on different body joints (Eq. 3.9). As the number of attention maps (*i.e.*, heads) N is no longer limited by the human topology, we can use it as a flexible hyper-parameter to improve the model. In the ablation study (see Fig. 3.9 and Tab. 3.2), our insight has been verified. Overall, our design strategy makes the architecture more flexible, effective and generic, which facilitates the study of cross-domain transfer learning in this field for datasets using different joint distributions (see Fig. 3.2).

3.5 Posetics: Skeleton Dataset

In this section, we introduce Posetics, a novel large-scale pre-training dataset. The Posetics dataset is created to study the transfer learning on skeleton-based action recognition. It contains 142,000 real-world video clips with the corresponding 2D and 3D poses including 17 body joints. All video clips in Posetics dataset are filtered from Kinetics-400 [18], to contain at least one human pose over 50% of frames.

Motivation and Data Collection: Recent skeleton-based action recognition methods on NTU-RGB+D [143, 103] can perform similarly or better compared to RGB-based methods. However, as laboratory indoor datasets may not contain occlusions, it is difficult to use such datasets to pre-train a generic model that can be transferred onto real-world videos, where skeleton data encounter a number of occlusions and truncations of the body. On the other hand, the accuracy based on skeleton data on the most popular real-world pre-training dataset, Kinetics [18], is still far below the accuracy on other datasets. The main problems are: (i) the skeleton data is hard to obtain by pose estimators as Kinetics is not human-centric. Human body may be missing or truncated by the image boundary in many frames. (ii) Many action categories are highly related to objects rather than human motion (*e.g.*, "making cakes", "making sushi" and "making pizza"). These make it difficult to effectively learn the human skeleton representation for recognizing actions. Hence, recent datasets [103, 197] are unable to significantly boost the action recognition performance when applied to different datasets. In order to better study the generalizability of skeleton-based models in the real-world, we extract the pose (*i.e.*, skeleton) data on Kinetics-400 [18] videos. Specifically, we compare the recent pose estimators and we extract pose data from RGB videos through multiple pose estimation systems. Then we apply the SSTA-PRS [200] presented in Sec. 3.3, a pose refinement system, for obtaining higher quality pose data in real-world videos. Moreover, for

problem (i), we filter out the videos where no body was detected, and for problem (ii), we slightly and manually modify the video category labels of Kinetics-400, and place emphasis on relating verbs to poses. (*e.g.*, For "making cakes", "making sushi" and "making pizza", we collectively chose the label "making food", whereas "washing clothes", "washing feet", and "washing hair" remain with the original labels). All in one, we organize 320 action categories for Posetics and this dataset can be more appropriately used for studying the real-world generalizability of skeleton-based action recognition models across datasets by transfer learning.

3.6 Experiments on Skeleton Refinements

Our objective is to obtain high-quality poses from real-world videos in order to understand human activities. We conduct a wealth of experiments to evaluate our system with two protocols: (1) Evaluation by the upstream pose refinement task using pose ground-truth, which is to directly compare the accuracy of the poses obtained from the pose estimators [139, 14, 48] with our proposed SSTA-PRS. (2) Evaluation by the downstream action recognition task using ground-truth action labels (see Sec. 3.7).

3.6.1 Datasets and Evaluation Protocols

Smarthome-Pose: in order to evaluate directly the poses extracted by our SSTA-PRS, we chose the middle frames for randomly selected 1,400 videos of Toyota Smarthome [36] (Smarthome) and we annotated the 2D poses to create a test set containing 1,400 images with 640×480 resolution and many occlusions, truncations. We follow the PCKh @0.5 (percent of keypoints within a threshold of 0.5 times head length) as the pose evaluation protocol. We regard the distance in pixels between head and neck as the head length.

NTU-Pose: NTU-RGB+D [143] is a large-scale multi-modal dataset which consists of 56,880 sequences of high-quality 2D/3D skeletons with 25 joints, associated with depth maps, RGB and IR frames captured by the Microsoft Kinect v2 sensor. We selected 60 videos (6,098 frames) with the same subject performing different actions and we took the 2D skeleton as the ground-truth for pose evaluation. The dataset was recorded in a laboratory, so in this work, we changed the original quality of the videos by reducing the resolution to 320×180 and adding partial occlusions to make it similar to our real-world settings. We use

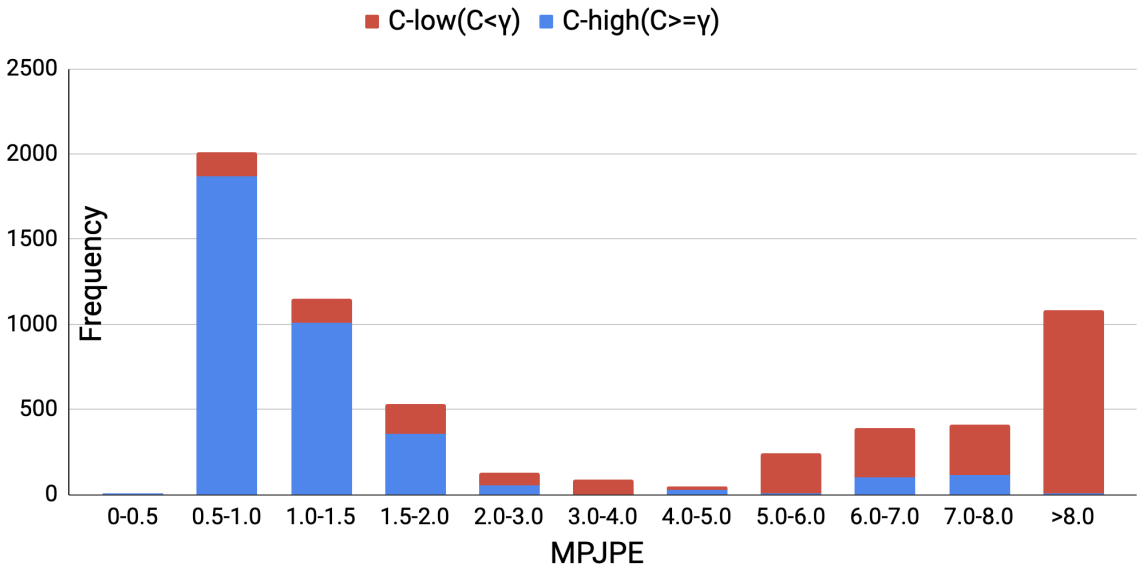

Fig. 3.7 Histogram of pose frequency in function of MPJPE with threshold $\gamma = 0.18$ (*i.e.*high confidence when C $> \gamma$).

the 2D MPJPE (mean per joint position error normalized by head length) and PCKh @2.0 protocols.

3.6.2 Implementation Details

Pose estimators: we select 1) OpenPose 18-joints [14], 2) AlphaPose Sample-Baseline [48] using YOLOv3 [136] as detector and 3) LCRNet++ In-The-Wild [139] as three expert pose estimation models and LCRNet++ In-The-Wild [139] as the student model for refining the pose. The poses contain the 13 common joints that all three estimators can detect.

Pose refinement: we select all the videos from NTU-Pose, 10% of the videos from Smarthome and 9.0K videos from Charades [153], and 10% of the frames for each video with uniform sampling to get a large dataset containing 3.0K images from NTU-Pose, 40.2K images from Smarthome, and 65.9K from Charades. We then split 20% of the images as the validation set. We apply SST-A mechanism using [139, 67, 14] as k_1, k_2 and k_3 and take 13 main keypoints for aggregation with $\gamma = 0.18$ as the confidence threshold for temporal filter to generate pseudo ground-truth 2D poses (Sec. 3.3.2). Then, we use In-The-Wild pre-trained model of LCRNet++ [139], which sets 20 'anchor poses' and leverages ResNet50 [70] as a backbone and we follow the standard setting values from [139]. We fine-tune this model

using 4 images per batch, and 512 boxes per image. The refined model is used to estimate 2D poses of the whole set of Smarthome and Charades.

To evaluate the performance of our upstream pose refinement system (SSTA-PRS), we experiment on NTU-Pose and Smarthome-Pose to directly compare the performance of expert pose estimators with our SSTA-PRS.

3.6.3 Results And Discussion

SST-A: We estimate 2D poses using three expert estimators and then perform SST-A without discarding any frames. The results in Tab. 3.1 show that SST-A is effective to integrate the advantages of the expert estimators and achieves a better performance (+7.7% on NTU-Pose, +1.3% on Smarthome-Pose).

Confidence metric: To analyze the reliability of the confidence metric (Sec. 3.3.2) that filters the poses for pseudo annotations, we analyse on the NTU-Pose the variation of the MPJPE with the confidence C. Fig. 3.8 shows the distribution of the aggregated poses from proposed SST-A. We find that the error decreases globally with the increase of confidence. Based on this figure, we select $\gamma = 0.18$ as the confidence threshold that can keep most of the aggregated poses within the error of 2.0. According to this threshold, we analyze the frequency of the retained (*i.e.*with C-high confidence) and discarded (*i.e.*with C-low confidence) poses within different error intervals (Fig. 3.7). Within the intervals of smaller errors, we keep the most of the poses and remove the ones in the larger error intervals. In order to have sufficient training samples, we still keep some poses with a few errors (but high confidence), corresponding to cases of complex scenes. Our fine-tuning system is weakly-supervised training, these poses can still play a positive role in localization and classification. Therefore, the confidence metric is instrumental in our work.

SSTA-PRS: After this filtering stage, we can get higher-quality pseudo 2D pose annotations for fine-tuning SSTA-PRS. Compared with the three expert estimators (Tab. 3.1), our SSTA-PRS is the most effective (+13.9% on NTU-Pose and +9.3% on Smarthome-Pose).

3.7 Experiments on Action Recognition

For evaluation of our framework on action recognition tasks, extensive experiments were conducted on five action classification datasets: **Toyota Smarthome (Smarthome) [36],**

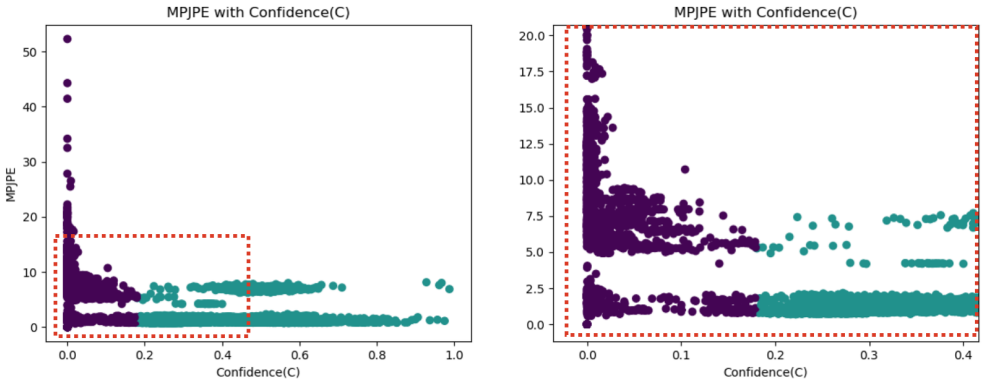

Fig. 3.8 Distribution of aggregated poses with MPJPE and Confidence. (purple: high confidence with $\gamma >= 0.18$, green: low confidence with $\gamma < 0.18$) Zoom of the red bounding box is on the right.

Methods	**NTU-Pose** PCKh @2.0 (%)	**Smarthome-Pose** PCKh @0.5 (%)
LCRNet++ [145]	54.1	64.4
AlphaPose [48]	53.2	55.5
OpenPose [14]	45.4	58.9
SST-A only(ours)	61.8	65.7
SSTA-PRS(ours)	**68.0**	**73.7**

Table 3.1 PCKh of poses from different pose estimators and proposed SSTA-PRS using SST-A only (Sec. 3.3.2) and using both SST-A and self-training (Sec. 3.3.3) on NTU-Pose and Smarthome-Pose.

Penn Action [213], NTU-RGB+D 60 (NTU-60) [143], NTU RGB+D 120 (NTU-120) [103] and the proposed **Posetics**. Firstly, we perform (i) exhaustive ablation study on Smarthome and NTU-60 without pre-training to verify the effectiveness of our proposed *dependency matrix* and *multi-head attention*. Then we (ii) re-evaluate state-of-the-art models [197, 145, 105], as well as our model on the proposed Posetics dataset (baselines are shown in Tab. 3.4), to provide an analysis on performance improvements on target datasets: Smarthome, Penn Action, NTU-60 and NTU-120, after pre-training on Posetics in order to demonstrate that our model generalizes well and benefits the most from pre-training. (iii) Final fine-tuned model is evaluated on all datasets to compare with the other state-of-the-art approaches for action recognition.

Evaluation Protocols: For Posetics, we split the dataset into 131,268 training clips and 10,669 test clips. We use Top-1 and Top-5 accuracy as evaluation metrics [197]. With respect to real-word settings, 2D poses extracted from images and videos tend to be more accurate than 3D poses, which are more prone to noise. Therefore, we only use 2D data for evaluation and comparison of the models on Posetics. We note that for pre-training, both 2D and 3D data can be used in order to obtain different models that can be transferred to datasets with different skeleton data. For the other datasets, we evaluate cross-subject (CS on Smarthome, NTU-60 and 120), cross-view (CV1 and CV2 on Smarthome and CV on NTU-60), cross-setup (CSet on NTU-120) protocols and the standard protocol (on Penn Action). Unless otherwise stated, we use 17 (2D) joints on Smarthome and Penn Action, 25 (3D) joints on NTU-60 and 120.

3.7.1 Implementation Details

Unless otherwise stated in the ablation study, all UNIK models have $N = 3, \tau = 1$ for S-LSU, and $t = 9$, $d = 1,3,3,3,3,1,1,1,1,1$, in each block respectively for T-LSU. We use SGD for training with momentum 0.9, an initial learning rate of 0.1 for 50, 30, 50, 60, and 65 epochs with step LR decay with a factor of 0.1 at epochs {30, 40}, {10, 20}, {30, 40}, {30, 50}, and {45, 55} for Smarthome, Penn Action, NTU-60, NTU-120, and Posetics, respectively. Weight decay is set to 0.0001 for final models. For NTU-60 and 120, all skeleton sequences are padded to 300 frames by replaying the actions. For Smarthome, Penn Action, Posetics, we randomly choose 400, 150, 150 frames respectively for each training epoch and all frames for test. 2D and 3D inputs are pre-processed with normalization and centering following [125], [145] respectively. As we have both 2D and 3D skeleton data on Posetics, we pre-train two models for transferring to benchmarks with different types of skeleton data. Note that for ablation study of UNIK (see 3.7.2), we train all models from scratch, without pre-training.

Number of Joints: SSTA-PRS [200] and LCRNet++ [139] provide 13 joints of the main body. We add "hip", "chest", "neck" and "nose" by interpolation and we obtain 17 joints for all experiments of real-world datasets (*i.e.*, Posetics, Smarthome, Penn Action). On NTU-60 and 120, we use 3D Kinect skeleton data with 25 joints for ablation study of UNIK (Sec. 5.2) while 17 main body joints for generalizability study (Sec. 5.3) to adapt to the pre-trained model on Posetics.

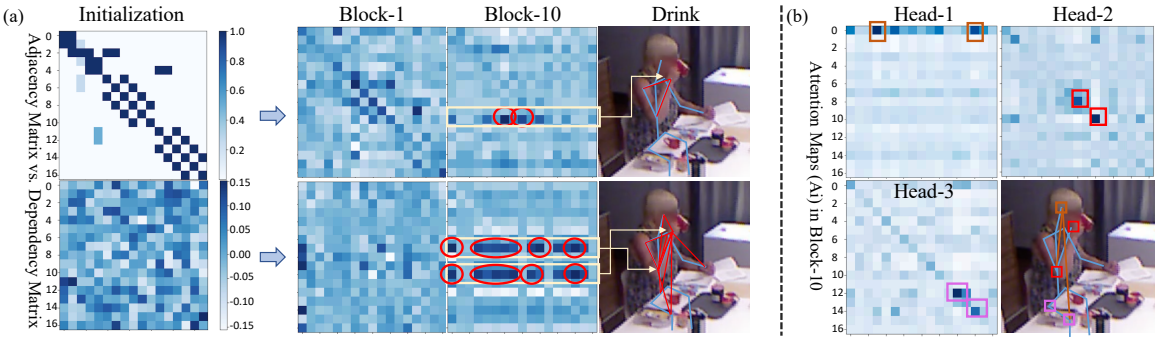

Fig. 3.9 (a) **Adaptive Adjacency Matrix [145] (top) vs. Dependency Matrix (bottom)** in different blocks for action "Drink" of Smarthome (right). They have different initial distributions. During training, the dependencies will become optimized representations, that are salient and more sparse in the deeper blocks, while our proposed matrix represents longer range dependencies (indicated by the red circles and red lines). (b) **Multi-head attention maps** in Block-10. Similar to dependency matrices, attention maps are salient and sparse in the deep block. The different heads automatically learn the relationships between the different body joints (as shown in the boxes and lines with different colors) to process long-range dependencies between joints instead of using pre-defined adjacency matrices.

3.7.2 Ablation Study of UNIK

Impact of Dependency Matrix: Here we compare the dependency matrices with the adaptive adjacency matrices. In order to verify our analysis in Sec. 3.4.2, we visualize the adjacency matrices [145] before and after learning. As shown in Fig. 3.9 (a) (top), we find that the previous learned graph [145] becomes a complete-graph, whose relationships are represented by weights that are well distributed over the feature maps. In contrast, our method is able to explore longer range dependencies, while being based on a dependency matrix with self-attention, which freely searches for dependencies of the skeleton from the beginning without graph-representation (see Fig. 3.9 (a)-bottom). Quantitatively, results in Tab. 3.2 show the effectiveness of the Dependency Matrix. Overall, we conclude that both our method and AGCN-based methods are fully connected layers with different initialization strategies and attention mechanisms in the spatial dimension, both are better than using a fixed graph [197]. It becomes evident that for skeleton-based tasks, where the number of nodes (*i.e.*, spatial body joints) is not large, multi-head attention based dependency matrix learning along with temporal convolutions can be a more generic and effective way to learn spatio-temporal dependencies compared with graph convolution.

Datasets (J)	**Matrix** ($N=3,\tau=1$)			**#Heads-N** ($\tau=1$)						**TW-τ** ($N=3$)				**TD** ($N=3,\tau=1$)	
	FM	AM	DM	0	1	3	6	9	12	1	3	6	9	×	✓
SH(%)	50.4	55.7	**58.5**	56.8	58.1	**58.5**	57.9	56.3	58.1	**58.5**	56.6	56.2	55.5	58.5	**58.9**
NTU-60(%)	84.3	86.1	**87.3**	86.8	87.0	87.3	87.1	85.8	**88.0**	87.3	86.8	**87.8**	85.0	87.3	**87.8**

Table 3.2 Ablation study on Smarthome (SH) CS and NTU-60 CS using joint (J) data only. FM: Fixed Adjacency Matrix (ST-GCN), AM: Adaptive Adjacency Matrix (AGCNs), DM: Dependency Matrix (Ours). TW: Temporal window size. TD: Temporal dilation.

Impact of Multi-head Attention: In this section, we visualize the multi-head attention maps and we analyze the impact of the number of heads N for UNIK with $N = 1,3,6,9,12,16$. As shown in Fig. 3.9, our multi-head aggregation mechanism can automatically learn the relationships between different positions of body joints by conducting the spatial processing (see Eq. 3.10) using the unified dependency matrices with a uniform initialization. Quantitative results in Tab. 3.2 show that obtaining a correct number of heads N is instrumental in improving the accuracy in a given dataset, but weakens the generalization ability across datasets with different types of actions (*e.g.*, the model benefits predominantly from $N = 12$ for NTU-60, and $N = 3$ for Smarthome). Consequently, we set $N = 3$ as a unified setting for all experiments and all datasets in order to balance the efficiency and performance of the model, as well as the generalization ability.

Other Ablations: For further analysis, results in Tab. 3.2 also show that (i) similar to [105], the size of the sliding window (see 3.4.1) τ can help to improve the performance, however weakening the generalizability of the model as it is sensitive to the number of frames in the video clip. (ii) Temporal dilated convolution contributes to minor boosts. See SM for more ablation study about initialization of Dependency Matrix and multi-stream fusion.

3.7.3 Impact of Pre-training:

In this section, we pre-train [145, 105] our proposed UNIK in a unified setting, ($N = 3, K = 10, \tau = 1$). Note that for pre-training GCN-based models [145, 105], we need to manually calibrate the different human topological structures in different datasets to keep the pre-defined graphs unified. For evaluation, we report the classification results on all the four datasets to demonstrate the impact of pre-training and to compare the generalization capacities *i.e.*, benefits compared to training from scratch. Note that unless otherwise stated, we use the consistent skeleton data (2D on Smarthome, Penn Action and 3D on NTU-60, 120), number of joints (17 main joints) for fair comparison of all models. On NTU-60 and

Methods	Pre-training	**Smarthome** (J)			**Penn Action** (J)	***NTU-60** (J+B)		***NTU-120** (J+B)	
		CS (%)	CV1 (%)	CV2 (%)	Top-1 Acc. (%)	CS (%)	CV (%)	CS (%)	CSet (%)
2s-AGCN [145]	Scratch	55.7	21.6	53.3	89.5	84.2	93.0	78.2	82.9
MS-G3D [105]	Scratch	55.9	17.4	56.7	88.5	**86.0**	**94.1**	**80.2**	**86.1**
UNIK (Ours)	Scratch	**58.9**	**21.9**	**58.7**	**90.1**	85.1	93.6	79.1	83.5
2s-AGCN [145]	Posetics	58.8	32.2	57.9	96.4	85.8	93.4	79.7	85.0
MS-G3D [105]	Posetics	59.1	26.6	60.1	92.2	86.2	94.1	80.6	86.4
UNIK (Ours)	Posetics	**62.1**	**33.4**	**63.6**	**97.2**	**86.8**	**94.4**	**80.8**	**86.5**

Table 3.3 Generalizability study of state-of-the-art by comparing the impact of transfer learning on Smarthome, Penn Action, NTU-60 and 120 datasets. The blue values indicate the best generalizabilities that can take the most advantage of pre-training on Posetics. "*" indicates that we only use 17 main joints adapted to the pre-trained model on Posetics.

120, we use both joint (J) and bone (B) data to compare the full models with two-stream fusion.

Generalizability Study: The results suggest that a consistent pre-training boosts all models, see Tab. 3.3, in particular, small benchmarks (*e.g.*, Smarthome CV and Penn Action with $5\% \sim 12\%$ improvement), as we do not have sufficiently large training data. Previous work [105] has a weak transfer capacity, due to the dataset-specific model settings (*e.g.*, the number of GCN scales and G3D scales) not always being able to adapt to the transferred datasets. On NTU-60, we take the main 17 joints for fine-tuning as we estimate and refine the main 17 joints on Posetics, and our pre-trained model outperforms state-of-the-art model [105]. Therefore, we conclude that our pre-trained model is the most generic-applicable especially for real-world scenarios. We provide further analysis in SM on (i) the pre-training on Posetics using 25 joints including the additional 8 joints on fingers and feet derived from linear interpolation for transferring on NTU-60 with full 25 joints and (ii) the evaluation of pre-trained features by linear classification on smaller datasets with the fixed backbone.

3.7.4 Comparison with State-of-the-art

We compare our full model (*i.e.*, Joint+Bone fusion) with and without pre-training to state-of-the-art methods, reporting results in Tab. 3.4 (Posetics, Smarthome and Penn Action). Note that for a fair comparison, we use the same skeleton data (2D and 17 joints) for all models. For real-world benchmarks using estimated skeleton data (*e.g.*, Posetics, Smarthome and Penn Action), our model without pre-training outperforms all state-of-the-art methods [109, 197, 145, 156, 105] in skeleton (*i.e.*, pose) stream and with pre-training, it outperforms the embedding-based method [164] that pre-trained on Human3.6M [76]. On NTU-60 and 120

Methods	RGB	Pose	Pre-training	Posetics		Smarthome			Penn Action
				Top-1(%)	Top-5(%)	CS(%)	CV1(%)	CV2(%)	Accuracy(%)
I3D [18]	✓		Kinetics-400	46.4	60.1	53.4	34.9	45.1	96.3
AssembleNet++ [140]	✓		Kinetics-400	-	-	63.6	-	-	-
NPL [131]	✓		Kinetics-400	-	-	-	39.6	54.6	-
Separable STA [36]	✓	✓	Kinetics-400	-	-	54.2	35.2	50.3	-
VPN [38]	✓	✓	Kinetics-400	-	-	60.8	43.8	53.5	-
Multi-task [108]	✓	✓	Scratch	-	-	-	-	-	97.4
LSTM [109]		✓	Scratch	-	-	42.5	13.4	17.2	-
ST-GCN [197]		✓	Scratch	43.3	67.3	53.8	15.5	51.1	89.6
2s-AGCN [145]		✓	Scratch	47.0	70.8	60.9	22.5	53.5	93.1
Res-GCN [156]		✓	Scratch	46.7	70.6	61.5	-	-	93.4
MS-G3D Net [105]		✓	Scratch	47.1	70.0	61.1	17.5	59.4	92.7
UNIK (Ours)		✓	Scratch	**47.6**	**71.3**	**63.1**	**22.9**	**61.2**	**94.0**
Pr-ViPE [164]		✓	Human3.6M	-	-	-	-	-	97.5
UNIK (Ours)		✓	Posetics(Ours)	-	-	**64.3**	**36.1**	**65.0**	**97.9**

Table 3.4 Comparison with state-of-the-art methods on the Posetics, Toyota Smarthome and Penn Action dataset. The best results using RGB data are marked in blue for reference.

(see Tab. 3.3), we compare to the most impressive two-stream graph-based methods [145, 105] and our model performs competitively without pre-training. We argue that we simplify our model as generically as possible without data-specific settings, which can improve the performance but weaken the transfer behavior (*e.g.*, the setting of N and τ). Subsequently, we further compare RGB-based methods [18, 36, 140, 131, 38, 108] for reference, that can be pre-trained on Kinetics-400 [18]. It suggests that previous skeleton-based methods [109, 197, 145, 105] without leveraging the pre-training are limited by the poor generalizability and the paucity of pre-training data. In contrast, our proposed framework, UNIK with pre-training on the Posetics dataset, outperforms state-of-the-art using RGB and even both RGB and pose data on the downstream tasks (*e.g.*, Smarthome and Penn Action).

3.8 Conclusion

In this chapter, we have proposed a novel method to extract pose sequences from challenging real-world videos. Owing to the proposed novel aggregation mechanism (SST-A) and weakly-supervised self-training framework, our method can be applied on videos in low-resolution, videos containing human body occlusions and truncations. We have also proposed UNIK, a unified framework for real-world skeleton-based action recognition. Our experimental analysis shows that UNIK is effective and has a strong generalization ability to transfer across datasets. In addition, we have introduced Posetics, a large-scale real-world skeleton-based action recognition dataset featuring high quality skeleton annotations. Our

experimental results demonstrate that pre-training on Posetics improves performance of the action recognition approaches. Future work involves an analysis of our framework for additional tasks involving skeleton sequences (*e.g.*, skeleton-based action segmentation, skeleton-based motion generation, etc.).

Chapter 4

Joint Skeleton Action Generation and Representation Learning

In this chapter, we focus on improving the generalization ability of UNIK onto more challenging tasks by action representation learning from generated skeleton data prior to action recognition.

Skeleton-based action classification and segmentation in real-world videos necessitates the recognition of composable actions in variant viewpoints. Existing methods often struggle to adequately express such actions due to limitations in visual feature extraction from skeleton sequences. In response, in this chapter, we present two novel self-supervised frameworks, namely Latent Action Composition (LAC)[1] and View-Invariant Action representation (ViA)[2].

LAC introduces a self-supervised approach that harnesses synthesized composable motions to enhance skeleton-based action segmentation. The framework incorporates a unique generation module, allowing the synthesis of diverse motion sequences through a linear latent space. By leveraging these synthesized sequences, LAC employs contrastive learning to develop robust visual encoders, which can be seamlessly applied to action segmentation tasks without the requirement of additional temporal models. Our extensive study demonstrates the superiority of LAC-based representations over state-of-the-art methods on various datasets, including TSU, Charades, and PKU-MMD.

On the other hand, ViA addresses the limitations of current self-supervised approaches, which predominantly focus on constrained scenarios with recorded data in laboratory settings. This framework tackles the challenges posed by variations in subjects and camera viewpoints in real-world videos. By utilizing motion retargeting as a pretext task, ViA disentangles

[1] Project website: https://walker1126.github.io/LAC/
[2] Project website: https://walker1126.github.io/ViA-project/

latent action-specific 'Motion' features from the visual representation of 2D or 3D skeleton sequences. These features remain invariant to skeleton geometry and camera view, thereby facilitating cross-subject and cross-view action classification tasks. Our evaluation on diverse datasets, including NTU-RGB+D 60, NTU-RGB+D 120, Toyota Smarthome, UAV-Human, and Penn Action, demonstrates versatility and effectiveness of ViA-generated representations in enhancing action classification accuracy.

Overall, the proposed LAC [204] and ViA [203] frameworks significantly contribute to the advancement of self-supervised skeleton-based action segmentation and representation learning, offering promising avenues for improved performance in real-world scenarios. The two works in this chapter have been published in IEEE/CVF International Conference on Computer Vision (ICCV) [204] in 2023, and in International Journal of Computer Vision (IJCV) [203] in 2024.

4.1 Introduction

Human-centric activity recognition is a crucial task in real-world video understanding. In this context, *skeleton data* that can be represented by 2D or 3D human keypoints plays an important role, as they are complementary to other modalities such as RGB [77, 18, 63, 51, 50, 140, 96, 182, 4] and optical flow [79, 53]. As the human skeleton modality has witnessed a tremendous boost in robustness *w.r.t.* content changes related to camera viewpoints and subject appearances, the study of recognizing activities directly from 2D/3D skeletons has gained increasing attention [43, 41, 13, 197, 145, 24, 157, 202, 22, 99, 44, 203]. While the aforementioned approaches have achieved remarkable success, they still have challenges in (1) segmenting composable actions and (2) recognizing cross-view and -subject actions in real-world videos. This chapter presents two novel joint generative and action representation learning framework based on skeletons focusing on these two challenges.

Action Segmentation in Untrimmed Videos: Current approaches often focus on *trimmed videos* containing *single actions*, which constitutes a highly simplified scenario. In this work, we tackle the challenging setting of *action segmentation in untrimmed videos based on skeleton sequences*. In untrimmed videos, activities are composable *i.e.*, a motion performed by a person generally comprises multiple actions (co-occurrence), each with a duration of a few seconds. To model *long-term dependency* among different actions, expressive skeleton features are required. Current approaches [89, 130, 129, 34] obtain such features through visual encoders such as AGCNs [145] pre-trained on trimmed datasets. However, due to

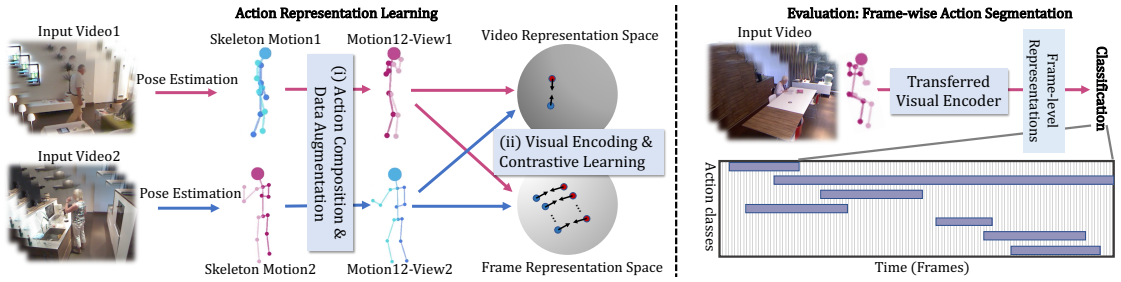

Fig. 4.1 **General pipeline of LAC.** Firstly, in the representation learning stage (left), we propose (i) a novel action generation module to combine skeletons of multiple videos (*e.g.*, 'Walking' and 'Drinking' shown in the top and bottom respectively). We then adopt a (ii) contrastive module to pre-train a visual encoder by learning data augmentation invariant representations of the generated skeletons in both video space and frame space. Secondly (right), the pre-trained encoder is evaluated by transferring to action segmentation tasks.

the limited motion information in the trimmed samples, the performance of such features in classifying complex actions is far from satisfactory. To address this issue, we propose to construct *synthesized composable skeleton data* for training a more effective visual encoder, endowed with strong representability of subtle action details for action segmentation.

In this chapter, we first propose Latent Action Composition (LAC), a novel framework to leverage synthesized composable motion data for self-supervised action representation learning. As illustrated in Fig. 4.1 (left), as opposed to current self-supervised approaches [89, 130, 129, 34], LAC learns action representations in two steps: a first *action composition* step is followed by a *contrastive learning* step.

Action composition is a novel initialization step to train a generative module that can generate new skeleton sequences by combining multiple videos. As high-level motions are difficult to combine directly by the joint coordinates (*e.g.*, 'drink' and 'sitdown'), LAC incorporates a novel Linear Action Decomposition (LAD) mechanism within an autoencoder. LAD seeks to learn an action dictionary to express subtle motion distribution in a discrete manner. Such action dictionary incorporates an orthogonal basis in the latent encoding space, containing two sets of directions. The first set, named 'Static', includes directions representing static information of the skeleton sequence, *e.g.*, viewpoints and body size. The other set, named 'Motion', includes directions representing temporal information of the skeleton sequence, *e.g.*, the primitive dynamics of the action performed by the subject. The new skeleton sequence is generated via a linear combination of the learned 'Static' and 'Motion' directions. We adopt motion retargeting to train the autoencoder and the dictionary using skeleton sequences with 'Static' and 'Motion' information built from 3D synthetic

data [75]. Once the action dictionary is constructed, in the following *contrastive learning* step, 'Static'/'Motion' information and action labels are not required and composable motions can be generated from any multiple input skeleton sequences by combining their latent 'Motion' sets.

The *contrastive learning* step aims at training a skeleton visual encoder such as UNIK [202] in a self-supervised manner, without the need for action labels (see Fig. 4.1 (middle)). It is designed for the resulting visual encoder to be able to maximize the similarity of different skeleton sequences, obtained via data augmentation from the same original sequence, across large-scale datasets. Unlike current methods [40, 74, 134, 74, 163, 97, 112, 203] that perform contrastive learning for the video-level representations, we perform contrastive learning additionally on the frame space to finely maximize the per-frame similarities between the positive samples. Subsequently, the so-trained frame-level skeleton visual encoder is transferred and retrained on action segmentation datasets [34, 153].

To assess the performance of LAC, we train the skeleton visual encoder on the large-scale dataset Posetics [202] and we evaluate the quality of the learned skeleton representations (see Fig. 4.1 (right)) by fine-tuning onto unseen action segmentation datasets (*e.g.*, TSU [34], Charades [153], PKU-MMD [27]). Experimental analyses confirm that action composition and contrastive learning can significantly increase the expressive power of the visual encoder. The fine-tuning results outperform state-of-the-art accuracy (see Sec. 4.5).

View-invariant Action Recognition: Many studies [191, 202, 44] have shown that 2D estimated skeletons are more accurate and more effective for action recognition compared to their estimated 3D counterparts in many real-world scenarios [213, 36, 101, 18], but 2D skeletons are sensitive to view and subject variations. Based on this observation, we hypothesize that action recognition, particularly based on 2D skeletons, could be further improved by embedding a *view-invariant representation of skeleton sequences*. In this context, in this chapter, we secondly propose ViA, a View-Invariant Action representation learning framework. Based on the disentanglement of 'Motion' and 'Static' features on the skeleton sequence, ViA also leverages motion retargeting as the training pre-text task. As the learned 'Motion' representation is subject and view agnostic, it can be effectively applied for cross-subject and cross-view action recognition by transfer-learning.

To assess the performance of ViA, we first pre-train ViA on the large-scale real-world Posetics dataset with a rich variety of subjects and viewpoints and we evaluate the quality of the learned action representation by fine-tuning and linear evaluation protocols on unseen 2D real-world action recognition datasets (*e.g.*, Toyota Smarthome, UVA-Human and Penn Ac-

tion). As ViA is not limited to 2D skeletons, we additionally validate the effectiveness of ViA on laboratory 3D datasets (*e.g.*, NTU-RGB+D 60 and 120). Experimental analyses confirm that through motion retargeting, ViA outperforms state-of-the-art methods [164, 97, 201, 162] on self-supervised action representation learning and the learned video representations can notably transfer to videos with cross-view and cross-subject challenges (see Sec. 4.6).

Contributions: In summary, the contributions of this chapter include the following. (i) We introduce LAC, a novel generative and contrastive framework, streamlined to synthesize complex motions and improve the skeleton action representation capability. (ii) In the generative step, we introduce a novel Linear Action Decomposition (LAD) mechanism to represent high-level motion features thanks to an orthogonal basis. The motions for multiple skeleton sequences can thus be linearly combined by latent space manipulation. (iii) In the contrastive learning step, we propose to learn the skeleton representations in both, video and frame space to improve generalization onto frame-wise action segmentation tasks. (iv) We conduct experimental analysis and we show that pre-training LAC on Posetics and transferring it onto an unseen target untrimmed video dataset represents a generic and effective methodology for action segmentation. (v) Based on the LAD mechanism, we introduce a novel skeleton-based action recognition framework ViA. Similar to LAC, ViA also leverages motion retargeting as a pretext task, but ViA aims at learning view- and subject-invariant skeleton-based action representations. (vi) We conduct a study that shows that pre-training ViA on Posetics and transferring it onto an unseen target dataset represents a generic and effective methodology for view- and subject-invariant action classification.

4.2 Related Work

Temporal Action Segmentation focuses on per-frame activity classification in untrimmed videos. The main challenge has to do with how to model long-term relationships among various activities at different time steps. Current methods mostly focus on directly using untrimmed RGB videos. Since untrimmed videos usually contain thousands of frames, training a single deep neural network directly on such videos is quite expensive. Hence, to solve this problem efficiently, previous works proposed to use a two-step method. In the first step, a pre-trained feature extractor (*e.g.*, I3D [18]) is applied on short sequences to extract corresponding visual features. In the second step, action segmentation is modeled as a sequence-to-sequence (seq2seq) task to translate extracted visual features into per-frame

action labels. Temporal Convolution Networks (TCNs) [89, 32, 209] and Transformers [31] are generally applied in the second step due to their ability to capture long-term dependencies.

Recently, few methods [30, 34] started to explore using skeletons in this task, in order to benefit from multi-modality information. In such methods, a pre-trained Graph Convolutional Network (GCN) such as AGCN [145] is used as a visual encoder to obtain skeleton features in the first step. However, unlike pre-trained I3D which has strong generalizability across domains, pre-trained AGCN is not able to provide high-quality features due to its laboratory-based pre-trained dataset NTU-RGB+D [143]. We found that the performance significantly decreases when the pre-trained model is applied to more challenging real-world untrimmed skeleton videos datasets such as TSU [34] and Charades [153]. The main issue is that the pre-trained visual encoder does not have a sufficient expressive power to extract the complex action features especially for composable actions that often occur in real-world videos.

LAC differs from previous two-step methods. We propose a motion generative module to synthesize complex composable actions and to leverage such synthetic data to train a more general skeleton visual encoder [202] which is sensitive to composable action. Unlike previous approaches, the pre-trained visual encoder in LAC has stronger representation capabilities for skeleton sequences compared to the previous two-step methods [30, 34] using pre-trained AGCN. In such strategy, the model can be end-to-end refined on the action segmentation tasks without need for the second stage.

View-invariant Skeleton Representation. To explore the view-invariant representation ability of human skeletons, previous methods [95, 88, 120, 119] aim at disentangling the view-dependent and pose-dependent features by two independent encoders on top of a single 3D skeleton using probabilistic embedding for view-invariant action recognition. To further address inherent ambiguities in 2D skeleton due to 3D-to-2D projection for action recognition, recent methods [164, 215, 141] perform the disentanglement learning on specific sensors (*e.g.*, motion capture system) capturing multi-view 2D skeletons. However, the aforementioned methods all process the skeleton sequence frame by frame, they are challenged in capturing the temporal features of the sequence and they are often not available when applied to common 2D datasets [36, 101, 213] where collecting data in multi-view is expensive and challenging.

In our work, ViA applies a generative task for the disentanglement and does not need multi-view data and 3D reconstruction. Moreover, unlike previous works that only disentangled static aspects ‘view’ and ‘pose’ for a single frame, the disentanglement of ViA is designed for ‘Character’ (including ‘view’ and ‘pose’) and ‘Motion’ coded in a sequence.

The important temporal dimension is considered to better generalize to action understanding tasks. By disentangling 'Motion' features using orthogonal decomposition in the latent space, ViA eliminates the requirement of explicit regularization terms [164, 215] that encourage disentanglement and smoothness of the learned representation.

Self-supervised Skeleton Action Representation learning involves extracting spatio-temporal features from numerous unlabeled data. Current methods [201, 97, 168, 112, 203] adopt contrastive learning [167, 192, 66] as the pretext task to learn skeleton representations invariant to data augmentation. However, recent techniques [162, 206, 201, 97, 168, 112, 203] merge the temporal features by average pooling and conduct contrastive learning on top of the global temporal features for the skeleton sequences. Thus they may lose important information of complex actions particularly in the case of co-occurring actions [34, 153]. In our work, we extend the visual encoder and the contrastive module to finely extract per-frame features. We use contrastive loss for both sequence and frame, to make sure that the skeleton sequences are discriminative in both spaces. The skeleton visual encoder can have a strong representation ability for the sequence and also for each frame to better generalize to frame-wise action segmentation tasks.

Motion Retargeting aims to transfer motion from sequence of target subject onto source subject, where the main challenge lies in developing effective mechanisms to disentangle motion and appearance. As one of the most important applications of video generation [171, 184, 185, 208, 154, 186], previous image-based motion retargeting approaches explore to leverage structure representations such as 2D human keypoints [174, 3, 19, 203] and 3D human meshes [104, 183] as motion guidance. Recently, self-supervised methods [151, 152, 189] showed remarkable results on human bodies and faces by only relying on data without extracting information.

Skeleton-based methods [1, 173, 3, 2] focus on transferring motion across skeletons of different shapes. Previous method [3] showed that transferring motion across characters enforces the disentanglement of static and dynamic information in a skeleton sequence. While they have achieved good performance, such a method is unable to compose different actions for creating novel actions. Our method is different, we seek to learn an orthogonal basis in the feature space to represent the action distribution in a linear and discrete manner. In such a novel strategy, both static and dynamic features can be learned from a single encoder and skeleton sequences with complex motions are able to be synthesized by simply modifying the magnitudes along the basis.

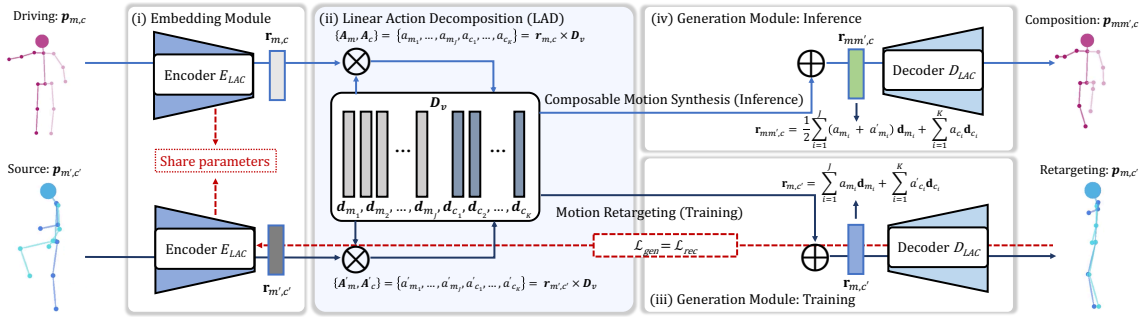

Fig. 4.2 **Overview of the Composable Action Generation model in LAC.** The model consists of a visual encoder $\mathrm{E_{LAC}}$ and a decoder $\mathrm{D_{LAC}}$. In the latent space, we apply Linear Action Decomposition (LAD) by learning a visual action dictionary $\mathbf{D}_v$, which is an orthogonal basis where each vector represents a basic 'Motion'/'Static' transformation. Given a pair of skeleton sequences $\mathbf{p}_{m,c}$ and $\mathbf{p}_{m',c'}$, (i) their latent codes $\mathbf{r}_{m,c}$ and $\mathbf{r}_{m',c'}$ are embedded by $\mathrm{E_{LAC}}$. (ii) Their projections A_m, A_c and $A_{m'}$, $A_{c'}$ along $\mathbf{D}_v$ can be computed. The linear combination of $A_m/A_{m'}$ with corresponding directions in $\mathbf{D}_v$ constitutes the 'Motion' features and similarly the 'Static' features can also be obtained. (iii) In the **training** stage, we leverage motion retargeting for learning the whole framework by swapping their 'Motion' features and generating transferred motions. (iv) In the **inference** stage, we adopt linear combination of $\mathbf{r}_m$ and $\mathbf{r}_{m'}$ to obtain the composable motion features $\mathbf{r}_{mm'}$ and the composable skeleton sequences can be generated.

4.3 LAC: Latent Action Composition

LAC is composed of two modules (see Fig. 4.1), a skeleton sequence generation module to synthesize the co-occurring actions and a self-supervised contrastive module to learn skeleton visual representations using the synthetic data. Subsequently, the skeleton visual encoder trained by the contrastive module can be transferred to downstream fine-grained action segmentation tasks. In this section, we introduce the full architecture and training strategy of LAC.

4.3.1 Action Generation Module

In this work, we denote the static information of a skeleton sequence (*i.e.*, 'viewpoint', 'subject body size', etc.) as 'Static', while the temporal information (*i.e.*, the dynamics of the 'action' performed by the subject) as 'Motion'. As shown in Fig. 4.2, the generative module is an autoencoder, consisting of an encoder and a decoder for skeleton sequences. To disentangle 'Motion' features from 'Static' in a linear latent space, we introduce a Linear Action Decomposition mechanism to learn an action dictionary where each direction

represents a basic high-level action for the skeleton encoding. We apply motion retargeting for training the autoencoder (*i.e.*, transferring the motion of a driving skeleton sequence to the source skeleton sequence maintaining the source skeletons invariant in viewpoint and body size). In the inference stage, the extracted 'Motion' features from multiple skeleton sequences can be combined linearly and composable skeletons can be generated by the decoder. The input skeletons can be in 3D or 2D.

Skeleton Sequence Autoencoder: The input skeleton sequence with 'Static' c and 'Motion' m is modeled by a spatio-temporal matrix, noted as $\mathbf{p}_{m,c} \in \mathbb{R}^{T\times V\times C_{in}}$. T, V, and C_{in} respectively represent the length of the video, the number of body joints in each frame, and the input channels ($C_{in} = 2$ for 2D data, or $C_{in} = 3$ if we use 3D skeletons). As shown in Fig. 4.2 (i), LAC adopts an encoder $\mathrm{E_{LAC}}$ to embed a pair of input skeleton sequences $\mathbf{p}_{m,c}/\mathbf{p}_{m',c'}$ into $\mathbf{r}_{m,c}/\mathbf{r}_{m',c'} \in \mathbb{R}^{T'\times C_{out}}$. T' is the size of temporal dimension after convolutions and C_{out} is the output channel size. To generate skeleton sequences, a skeleton sequence decoder $\mathrm{D_{LAC}}$ (see Fig. 4.2 a.(iii)) is used to generate new skeleton sequences from the representation space. The autoencoder is designed by multiple 1D temporal convolutions and upsampling to respectively encode and decode the skeleton sequence. We provide in Tab. 4.1 the building details of $\mathrm{E_{LAC}}$ and $\mathrm{D_{LAC}}$.

Linear Action Decomposition: The goal of Linear Action Decomposition (LAD) is to obtain the 'Motion' features on top of the encoded latent code of a skeleton sequence (see Fig. 4.2 a.(ii)). Our insight is that the high-level action of a skeleton sequence can be considered as a combination of multiple basic and independent 'Motion' and 'Static' transformations (*e.g.*, raising hand, bending over) with their amplitude from a fixed reference pose (*i.e.*, standing in the front view, see Fig. 4.4). Hence, we explicitly model the basic 'Static' and 'Motion' transformations using a unified action dictionary for the encoded latent skeleton features. Specifically, we first predefine a learnable orthogonal basis, noted as $\mathbf{D}_v = \{\mathbf{d_{m}}1, \mathbf{d_{m}}2, ..., \mathbf{d_{m}}J, \mathbf{d_{c}}1, \mathbf{d_{c}}2, ..., \mathbf{d_{c}}K\}$ with $J \in [1, C_{out})$ and $K = C_{out} - J$, where each vector indicates a basic 'Motion'/'Static' transformation from the reference pose. Due to $\mathbf{D}_v$ entailing an orthogonal basis, both directions $\mathbf{d_i}, \mathbf{d_j}$ follow the constraint:

$$< \mathbf{d_i}, \mathbf{d_j} >= \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases} \tag{4.1}$$

We implement $\mathbf{D}_v \in \mathbb{R}^{C_{out} \times C_{out}}$ as a learnable matrix and we apply the Gram-Schmidt algorithm during each forward pass in order to satisfy the orthogonality. Then, we consider the 'Motion' features of $\mathbf{p}_{m,c}$, denoted as $\mathbf{r}_m$, as a linear combination between motion orthogonal directions in $\mathbf{D}_v$, and associated magnitudes (amplitude) $A_m = \{a_{m1}, a_{m2}, ..., a_{mJ}\}$. Similarly, the 'Static' features $\mathbf{r}_c$ are the linear combination between 'Static' orthogonal directions in $\mathbf{D}_v$, and associated magnitudes $A_c = \{a_{c1}, a_{c2}, ..., a_{cK}\}$. For $\mathbf{p}_{m',c'}$, we can obtain its decomposed components $\mathbf{r}_{m'}$, $\mathbf{r}_{c'}$ in the same way:

$$\begin{aligned} \mathbf{r}_m &= \sum_{i=1}^{J} a_{mi}\mathbf{d}_{\mathbf{m}i}, \quad \mathbf{r}_c = \sum_{i=1}^{K} a_{ci}\mathbf{d}_{\mathbf{c}i}, \\ \mathbf{r}_{m'} &= \sum_{i=1}^{J} a'_{mi}\mathbf{d}_{\mathbf{m}i}, \quad \mathbf{r}_{c'} = \sum_{i=1}^{K} a'_{ci}\mathbf{d}_{\mathbf{c}i}. \end{aligned} \tag{4.2}$$

For the skeleton encoding $\mathbf{r}_{m,c}/\mathbf{r}_{m',c'}$, the set of magnitudes A_m/A'_m and A_c/A'_c can be computed as the projections of $\mathbf{r}_{m,c}/\mathbf{r}_{m',c'}$ onto $\mathbf{D}_v$, as Eq. 4.3:

$$\begin{aligned} a_{mi} &= \frac{< \mathbf{r}_{m,c} \cdot \mathbf{d}_{\mathbf{m}i} >}{\|\mathbf{d}_{\mathbf{m}i}\|^2}, \quad a_{ci} = \frac{< \mathbf{r}_{m,c} \cdot \mathbf{d}_{\mathbf{c}i} >}{\|\mathbf{d}_{\mathbf{c}i}\|^2}, \\ a'_{mi} &= \frac{< \mathbf{r}_{m',c'} \cdot \mathbf{d}_{\mathbf{m}i} >}{\|\mathbf{d}_{\mathbf{m}i}\|^2}, \quad a'_{ci} = \frac{< \mathbf{r}_{m',c'} \cdot \mathbf{d}_{\mathbf{c}i} >}{\|\mathbf{d}_{\mathbf{c}i}\|^2}. \end{aligned} \tag{4.3}$$

As $\mathbf{r}_{m,c}$ has the temporal dimension of size T', for each 'Motion' feature in the temporal dimension, we can obtain $T'\times$ sets of motion magnitudes A_m to represent the temporal dynamics of $\mathbf{r}_m$. For $\mathbf{r}_c$, as static information, we firstly merge the temporal dimension of $\mathbf{r}_{m,c}$ by average pooling and we then conduct the projection process to obtain a unified A_c. With such trained LAD, the decoder $\mathrm{D}_{\mathrm{LAC}}$ can generate different skeleton sequences by taking an arbitrary combination of magnitudes A_m and A_c along their corresponding directions as input. The high-level action can thus be controlled by the manipulations in the latent space.

Training (Motion Retargeting): We apply a general motion retargeting [3] to train the generative autoencoder and ensure that 'Motion' directions in LAD orthogonal basis $\mathbf{D}_v$ are 'Static'-disentangled (see Fig. 4.2 (iii)). The main training loss function is the *reconstruction loss*: $\mathscr{L}_{gen}$=$\mathscr{L}_{rec}$. Reconstruction loss aims at guiding the network towards a high generation quality. The new retargeted (motion swapped) skeleton sequence with 'Motion' m, and 'Static' c', noted as $\mathbf{p}_{m,c'}$ is generated from the recombined features, $\mathbf{r}_m + \mathbf{r}_{c'}$. Similarly, $\mathbf{p}_{m',c}$ can also be generated by swapping the pair of sequences. The skeleton sequence generation can be formulated as $\mathbf{p}_{m,c'} = \mathrm{D}_{\mathrm{LAC}}(\mathbf{r}_m + \mathbf{r}_{c'})$ and $\mathbf{p}_{m',c} = \mathrm{D}_{\mathrm{LAC}}(\mathbf{r}_{m'} + \mathbf{r}_c)$. The reconstruction

Stages	$\mathrm{E_{LAC}}$	$\mathrm{D_{LAC}}$	$\mathrm{E_V}$
Input	2D sequence $[T, 2V]$	Rep. $[T',\ 160]$	2D sequence $[T \times V,\ 2]$
1	$\mathrm{Conv}(\ 8,\ 64\ )$	Upsample(2) $\mathrm{Conv}(\ 7,\ 128\ )$	$\mathrm{Conv}\begin{pmatrix} 1\times1,\ 64 \\ 9\times1,\ 64 \end{pmatrix}\times4$
2	$\mathrm{Conv}(\ 8,\ 96\ )$	Upsample(2) $\mathrm{Conv}(\ 7,\ 64\ )$	$\mathrm{Conv}\begin{pmatrix} 1\times1,\ 128 \\ 9\times1,\ 128 \end{pmatrix}\times3$
3	$\mathrm{Conv}(\ 8,\ 160\ )$	Upsample(2) $\mathrm{Conv}(\ 7,\ 2V\ )$	$\mathrm{Conv}\begin{pmatrix} 1\times1,\ 256 \\ 9\times1,\ 256 \end{pmatrix}\times3$
4	-	-	S-GAP $(2\times V,\ 256)$
Rep.	-	-	$\mathrm{E_{Vf}}$: $[T,\ 256]$ $\mathrm{E_{Vs}}$: T-GAP to $[1,\ 256]$
5	-	-	FC, Softmax
Output	$[T',\ 160]$	2D sequence $[T,\ 2V]$	Per-frame Action Class

Table 4.1 **Main building blocks** of the autoencoder $\mathrm{E_{LAC}}$, $\mathrm{D_{LAC}}$ and the skeleton visual encoder $\mathrm{E_V}$ in LAC. We take the 2D sequence as example. The dimensions of kernels are denoted by $t \times s, c$ (2D kernels) and t, c (1D kernels) for temporal, spatial, channel sizes. S/T-GAP, FC denotes temporal/spatial global average pooling, and fully-connected layer respectively. Rep. indicates the learned representation.

loss consists of two components: $\mathscr{L}_{rec} = \mathscr{L}_{self} + \mathscr{L}_{target}$. Specifically, at every training iteration, the decoder network $\mathrm{D_{LAC}}$ is firstly used to reconstruct each of the original input samples $\mathbf{p}_{m,c}$ using its representation $\mathbf{r}_m + \mathbf{r}_c$. This component of the loss is denoted as $\mathscr{L}_{self}$ and formulated as a standard autoencoder reconstruction loss (see Eq. 4.4).

$$\begin{aligned} \mathscr{L}_{self} &= \mathbb{E}[\|\mathrm{D_{LAC}}(\mathbf{r}_m + \mathbf{r}_c) - \mathbf{p}_{m,c}\|^2], \\ \mathscr{L}_{target} &= \mathbb{E}[\left\|\mathrm{D_{LAC}}(\mathbf{r}_m + \mathbf{r}_{c'}) - \mathbf{p}_{m,c'}\right\|^2]. \end{aligned} \tag{4.4}$$

Moreover, at each iteration, the decoder is also encouraged to re-compose new combinations. As the generative module is trained on a synthetic dataset [75] including the cross-character motion retargeting ground-truth skeleton sequences, we can explicitly apply the cross reconstruction loss $\mathscr{L}_{target}$ (see Eq. 4.4) through the generation. The same reconstruction losses are also computed for $\mathbf{p}_{m',c'}$.

Inference (Composable Action Generation): As the trained LAD represents high-level motions in a linear space by the action dictionary, we can generate at the inference stage (see Fig. 4.2 (iv)) composable motions by the linear addition of 'Motion' features encoded from multiple skeleton sequences. We use the average latent 'Motion' features for the decoder to

generate composable motions. We note that, even if in some cases the combined motions may not be realistic, it can still help to increase the expressive power of the representation, which is important to express subtle details. Taking the motion combination of the two sequences $\mathbf{p}_{m,c}$ and $\mathbf{p}_{m',c'}$ as an example, the skeleton sequences $\mathbf{p}_{mm',c}$ and $\mathbf{p}_{mm',c'}$ with the combined motions m and m' are generated as follows:

$$\begin{aligned} \mathbf{p}_{mm',c} &= \mathrm{D}_{\mathrm{LAC}}\left(\frac{1}{2}(\mathbf{r}_m + \mathbf{r}_{m'}) + \mathbf{r}_c\right), \\ \mathbf{p}_{mm',c'} &= \mathrm{D}_{\mathrm{LAC}}\left(\frac{1}{2}(\mathbf{r}_m + \mathbf{r}_{m'}) + \mathbf{r}_{c'}\right). \end{aligned} \tag{4.5}$$

As skeleton sequences $\mathbf{p}_{mm',c}$ and $\mathbf{p}_{mm',c'}$ have the same composed motion but different 'Static' (*e.g.*, viewpoints), they can form a positive pair for self-supervised contrastive learning to train a transferable skeleton visual encoder for fine-grained action segmentation tasks in Sec. 4.3.2.

4.3.2 Self-supervised Action Representation Learning

In this section, we provide details of the self-supervised contrastive module of LAC.

We re-denote the generated composable skeleton sequence $\mathbf{p}_{mm',c}$ (in Sec. 4.3.1) as a query clip q and multiple positive keys (*e.g.*, the sequence $\mathbf{p}_{mm',c'}$), denoted as $k_1^+, ..., k_P^+$, can be generated by only modifying its 'Static' magnitudes A_c in the latent space. We follow the general contrastive learning method [66] based on the momentum encoder, to maximize the mutual information of positive pairs (*i.e.*, the generated composable skeleton sequences with the same motion but different Statics), while pushing negative pairs (*i.e.*, other skeleton sequences with different Motions) apart. Deviating from [66], the queue (memory) [66] stores the features of each frame for skeleton sequences and we propose to additionally enhance the per-frame representation similarity of positive pairs. The visual encoder can extract skeleton features that are globally invariant and also finely invariant to data augmentation and it generalizes better to frame-wise action segmentation tasks.

Skeleton Visual Encoder: To have a strong capability to extract skeleton spatio-temporal features, we adopt the recent topology-free skeleton backbone network UNIK [202] as the skeleton visual encoder $\mathrm{E_V}$ (see Tab. 4.1 for details). To obtain the global sequence space, we adopt an temporal average pooling layer to merge the temporal dimension of the visual representations, denoted as $\mathrm{E_{Vs}}(q), \mathrm{E_{Vs}}(k_1^+), ..., \mathrm{E_{Vs}}(k_P^+) \in \mathbb{R}^{C_{out} \times 1}$ (see Tab. 4.1). Per-frame

features can be obtained by $\mathrm{E_V}$ before the temporal average pooling layer (see Tab. 4.1) and denoted as $\mathrm{E_{Vf}}(q,\tau),\mathrm{E_{Vf}}(k_1^+,\tau),...,\mathrm{E_{Vf}}(k_P^+,\tau) \in \mathbb{R}^{C_{out}\times T}$.

Contrastive Loss: We apply general contrastive InfoNCE loss [122] to train our visual encoder $\mathrm{E_V}$ to encourage similarities between both sequence-level and frame-level representations of positive pairs, and discourage similarities between negative representations, denoted as $\mathrm{E_{Vs}}(k_1^-),...,\mathrm{E_{Vs}}(k_N^-)$ in sequence space and $\mathrm{E_{Vf}}(k_1^-,\tau),...,\mathrm{E_{Vf}}(k_N^-,\tau)$ in frame space. The InfoNCE [122] objective is defined as: $\mathscr{L}_q = \mathscr{L}_{q-s} + \mathscr{L}_{q-f}$, where

$$\mathscr{L}_{q-s} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\mathrm{Sim}\left(\mathrm{E_{Vs}}(q),\mathrm{E_{Vs}}(k_p^+)\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathrm{E_{Vs}}(q),\mathrm{E_{Vs}}(k_n^-)\right)}}\Big), \tag{4.6}$$

$$\mathscr{L}_{q-f} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\sum_{\tau=1}^{T}\mathrm{Sim}\left(\mathrm{E_{Vf}}(q,\tau),\mathrm{E_{Vf}}(k_p^+,\tau)\right)}}{\sum_{n=1}^{N} e^{\sum_{\tau=1}^{T}\mathrm{Sim}\left(\mathrm{E_{Vf}}(q,\tau),\mathrm{E_{Vf}}(k_n^-,\tau)\right)}}\Big), \tag{4.7}$$

where τ represents the frame index in the temporal dimension of frame-level representations, P represents the number of positive keys, N denotes the number of negative keys (we use $P = 4$ and $N = 65,536$ for experiments), and the similarity is computed as:

$$\mathrm{Sim}(x,y) = \frac{\phi(x)\cdot\phi(y)}{\|\phi(x)\|\cdot\|\phi(y)\|}\cdot\frac{1}{Temp}, \tag{4.8}$$

where $Temp$ refers to the temperature hyper-parameter [192], and ϕ is a learnable mapping function (*e.g.*, a MLP projection head [52]) that can substantially improve the learned representations.

Transfer-Learning for Action Segmentation: For transferring the visual encoder on downstream tasks, we attach $\mathrm{E_{Vf}}$ to a fully-connected layer followed by a Softmax Layer to predict per-frame actions. The output size of each fully-connected layer depends on the number of action classes (see Tab. 4.1). Then, we re-train the visual encoder $\mathrm{E_V}$ with action labels. For processing long sequences, we adopt a sliding window to extract features for a temporal segment and we use Binary Cross Entropy loss to optimize the visual encoder step by step. In this way, $\mathrm{E_V}$ can be re-trained end-to-end instead of pre-extracting features for all frames. In the inference stage, we combine the predictions of all the temporal sliding windows in an online manner [102].

4.4 ViA: View-invariant Action Representation

To address the limitations of current skeleton-based action recognition models due to the large variations across subjects and camera viewpoints, we introduce ViA. ViA aims at pre-training a generic and view-invariant visual encoder. To do so, based on the LAD mechanism of LAC, we can generate multi-view skeleton sequences with the same motion by modifying only the disentangled 'Static' features of the given sequence. Specifically, the composable skeletons from the generated module $\mathbf{p}_{mm',c}$ and the $\mathbf{p}_{mm',c'}$ can be positive samples for the contrastive learning module presented in Sec. 4.3.2. As cross-view action recognition is generally evaluated on trimmed dataset [36, 143, 103], in this work, the model properties are verified by transfer-learning of $\mathrm{E_V}$ for action classification tasks. In practice, we attach the skeleton encoder, where the pre-trained weights are used as initialization, to a temporal global average pooling layer and a fully-connected layer followed by a Softmax Layer. The output size of each fully-connected layer depends on the number of action classes. Then, we re-train the network with action labels on the target datasets. Following common evaluation protocols used in previous unsupervised action representation frameworks [97, 206, 162, 201], we conduct both a *linear evaluation* by training only the fully-connected layer with the backbone frozen, and a *fine-tuning evaluation* by further refining the whole network on downstream tasks.

4.5 Experiments and Analysis on LAC

In this section, we conduct extensive experiments to evaluate LAC on both generation and action segmentation tasks. Firstly, we study the generalization ability of LAC by quantifying the performance improvement obtained by transfer-learning on target action segmentation datasets (*i.e.*, **Toyota Smarthome Untrimmed**, **Charades** and **PKU-MMD**) after pre-training on the large-scale dataset **Posetics**. Secondly, we evaluate the quality of the skeleton sequences generated by LAC using the synthetic dataset **Mixamo**. Finally, we provide an exhaustive ablation study.

4.5.1 Implementation Details

Building Details of Networks: In the generation module, the autoencoder has two networks, *i.e.*, a *skeleton sequence encoder* $\mathrm{E_{LAC}}$ and a *skeleton sequence decoder* $\mathrm{D_{LAC}}$, built as in [3]. Both networks are composed of multiple 1D temporal convolutions to process the

Methods	**Mod.**	**TSU**		**Charades**
		CS(%)	CV(%)	mAP(%)
TGM [130]	RGB	26.7	-	13.4
PDAN [32]	RGB	32.7	-	23.7
SD-TCN [34]	RGB	29.2	18.3	21.6
MS-TCT [31]	RGB	33.7	-	25.4
Bi-LSTM [58]	Skeleton	17.0	14.8	8.2
TGM [130]	Skeleton	26.7	13.4	9.0
SD-TCN [34]	Skeleton	26.2	22.4	9.8
LAC-unsup (Ours)	Skeleton	**34.1**	**22.8**	**22.3**
LAC-sup (Ours)	Skeleton	**36.8**	**23.1**	**25.6**

Table 4.2 Frame-level mAP on TSU and Charades for comparison with SoTA action segmentation methods. RGB-based results (top) are shown for reference. Mod.: Modality.

skeleton sequences. To decode the skeleton sequence, D_{LAC} includes upsampling processes along the temporal dimension to reconstruct the skeleton sequences.

The skeleton visual encoder in the contrastive modules E_V is composed of 10 convolutional building blocks. Each building block contains a spatial network and a temporal convolutional network to extract both spatial and temporal multi-scale features from the skeleton sequence. For the spatial processing, we utilize 1×1 convolutions to expand the data channels and then multiply the features by uniformly initialized [68] and learnable dependency matrices (which replace the adjacency matrices used in GCN-based methods [197, 145, 105, 22]). For the temporal processing, we utilize 9×1 convolutions. The size of the temporal dimension of embedded latent 'Motion' T' depends on the duration of the input sequence. For transfer-learning on action segmentation tasks, we attach the visual encoder to a fully-connected layer followed by a Softmax Layer to predict per-frame classifications. The output size of each fully-connected layer depends on the number of action classes. Then, we re-train the network with action labels.

Training Details of Generation Module: The autoencoder can be previously and effectively trained on a synthetic dataset using cross-reconstruction ground truth, *i.e.*, the same motion pattern performed by different characters and in different viewpoints obtained by rotated 3D and projected 2D skeletons. As Mixamo [75] is a 3D animation collection, including elementary actions, and various dancing moves, we first train LAC on Mixamo to disentangle the 'Motion' features and learn the action dictionary. Then we conduct contrastive learning using the pre-trained and fixed autoencoder, in order to train the skeleton visual encoder E_V

Methods	Mod.	PKU-MMD mAP@IoU 0.1(%)	0.3(%)	0.5(%)
GRU-GD [107]	RGB	82.4	81.3	74.3
SSTCN-GD [30]	RGB	83.7	82.1	76.5
Augmented-RGB [30]	RGB	86.3	84.5	81.1
JCRRNN [102]	Skeleton	45.2	-	32.5
Convolution Skeleton [27]	Skeleton	49.3	31.8	12.1
Skeleton boxes [92]	Skeleton	61.3	-	54.8
Hi-TRS [23]	Skeleton	-	-	67.3
Window proposal [93]	Skeleton	92.2	-	90.4
LAC-unsup (Ours)	Skeleton	**91.8**	**90.2**	**88.5**
LAC-sup (Ours)	Skeleton	**92.6**	**91.4**	**90.6**

Table 4.3 Event-level mAP on PKU-MMD CS at IoU thresholds of 0.1, 0.3 and 0.5 for comparison with SoTA methods. RGB-based results (top) are shown for reference. Mod.: Modality.

Methods	Pre-training	Training data	Toyota Smarthome Untrimmed CS(%)	CV(%)	PKU-MMD (IoU=0.1) CS(%)	CV(%)	Charades mAP(%)
Random init. [202]	Scratch	5%	8.5	6.8	57.4	59.5	8.8
Self-supervised	Posetics w/o labels	5%	**25.2**	**15.6**	**73.9**	**75.4**	**12.6**
Random init. [202]	Scratch	10%	12.9	9.5	66.4	68.1	9.3
Self-supervised	Posetics w/o labels	10%	**29.0**	**17.9**	**79.8**	**81.1**	**17.4**

Table 4.4 Transfer learning results by **fine-tuning** on all benchmarks of Toyota Smarthome Untrimmed, PKU-MMD and Charades with randomly selected **5% (top)** and **10% (bottom)** of labeled training data.

in a self-supervised manner on the large-scale trimmed pre-training dataset, Posetics. Finally, the trained visual encoder is transferred onto target action segmentation tasks.

Training details of Contrastive Module: We adopt UNIK as the visual encoder with the same hyper-parameter settings as [202]. For self-supervised pre-training on Posetics, we adopt all related hyper-parameter settings of [52] to train the contrastive model MoCo [66]. For the momentum encoder, we use a queue storing N =8192 negatives with m_{base} =0.994 and we use a 2-layer projection MLP. The temperature *Temp* is set as 0.1. We adopt a half-period cosine schedule [52] of learning rate decaying, with a base learning rate of 0.1 and maximum training iterations of 200. For the downstream action segmentation tasks, we use an initial learning rate of 0.1 for 50 epochs with step LR decay with a factor of 0.1 at epochs {30, 40} for all the three evaluated datasets. Weight decay is set to 1×10^{-4} for final models. For action segmentation on TSU, Charades and PKU-MMD, we adopt a temporal sliding window with sizes 300, 64, 300 frames respectively along the untrimmed sequences

Methods	**Pre-training**	**Toyota Smarthome Untrimmed**			**PKU-MMD** (IoU=0.1)			**Charades**	
		#Params	CS(%)	CV(%)	#Params	CS(%)	CV(%)	#Params	mAP(%)
Random init.	Scratch	13.1K	8.1	6.9	13.3K	11.8	12.4	40.2K	6.1
Supervised	Posetics w/ labels	13.1K	20.8	18.3	13.3K	61.8	62.4	40.2K	14.3
Self-supervised	Posetics w/o labels	13.1K	**18.5**	**16.6**	13.3K	**55.2**	**58.8**	40.2K	**12.7**
Random init.	Scratch	3.45M	28.2	11.0	3.45M	86.5	92.9	3.45M	18.6
Supervised	Posetics w/ labels	3.45M	36.8	23.1	3.45M	92.6	94.6	3.45M	25.6
Self-supervised	Posetics w/o labels	3.45M	**34.1**	**22.8**	3.45M	**91.8**	**93.9**	3.45M	**22.3**

Table 4.5 Transfer-learning results by **linear evaluation (top)** and **fine-tuning (bottom)** on Toyota Smarthome Untrimmed, PKU-MMD and Charades with self-supervised pre-training on Posetics. Results with supervised pre-training are also reported for reference.

for training the visual encoder. 2D skeleton inputs (on TSU and Charades) are pre-processed with normalization and centering following [125].

4.5.2 Evaluation on Temporal Action Segmentation

In this section, we evaluate the transfer ability of LAC by both *linear evaluation* (*i.e.*, by training only the fully-connected layer while keeping frozen the backbone) and *fine-tuning evaluation* (*i.e.*, by refining the whole network) on three action segmentation datasets TSU, PKU-MMD and Charades with self-supervised pre-training on Posetics. We also report the results with supervised pre-training for reference (*i.e.*, we use the generated composable skeletons and the combined action labels for pre-training).

Linear Evaluation: Tab. 4.5 (top) shows the linear results on the three datasets. This evaluates the effectiveness of transfer-learning with fewer parameters (only the classifier is trained) compared to training directly on the target datasets from scratch (random initialization). The results suggest that the weights of the model can be well pre-trained without action labels, providing a strong transfer ability (*e.g.*, +10.4% on TSU CS and +6.6% on Charades) and the pre-trained visual encoder is generic enough to extract meaningful action features from skeleton sequences.

Fine-tuning: Tab. 4.5 (bottom) shows the fine-tuning results, where the whole network is re-trained. The self-supervised pre-trained model also performs competitively compared to supervised pre-trained models. From these results we conclude that collecting a large-scale trimmed skeleton dataset, without the need of action annotation, can be beneficial to downstream fine-grained tasks for untrimmed videos (*e.g.*, +5.9% on TSU CS and +11.8% on CV).

Training with fewer labels: In many real-world applications, labeled data may be lacking, which makes it challenging to train models with good performance. To evaluate LAC in such cases, we transfer the visual encoder pre-trained on Posetics onto all the tested datasets by fine-tuning with only 5% and 10% of the labeled data. As shown in Tab. 4.4, without pre-training, the accuracy of the visual encoder [202] significantly decreases. In contrast, LAC with prior action representation learning achieves good performance on all three datasets in such setting.

Comparison with SoTA: We compare our fine-tuning results to other SoTA skeleton-based approaches [58, 130, 34, 102, 92, 27, 23, 93] on the real-world datasets TSU and Charades (see Tab. 4.2) and also laboratory dataset PKU-MMD (see Tab. 4.3). As previous approaches are based on supervised pre-training on large-scale datasets [103, 18], we also report our supervised results. The results in Tab. 4.2 show that LAC, even with self-supervised pre-training, outperforms all previous skeleton-based approaches [58, 130, 34] with supervised pre-training on our main target real-world datasets in a large margin (*e.g.*, +7.4% on TSU CS and +12.5% on Charades). It suggests that composable motions are important to increase the expressive power of the visual representation and the end-to-end fine-tuning can benefit downstream tasks. Even if PKU-MMD does not contain composable actions, the performance is still slightly improved by learning a fine-grained skeleton representation. The results using RGB data are also reported for reference. The TSU and Charades datasets contain many object-oriented actions that are difficult to identify using skeleton data only. However, even in the absence of the object information, LAC surprisingly achieves better accuracy compared to all SoTA RGB-based methods [32, 34, 31, 130, 30]. We deduce that training the visual encoder end-to-end is more effective compared to using two-step processing. Moreover, skeletons can always be combined with RGB data by multi-modal fusion networks [30, 37] to further improve the performance.

4.5.3 Evaluation on Action Generation.

As the generative model with LAD represents our main novelty for addressing the action segmentation challenges, we evaluate here the generation quality of LAC.

Quantitative Comparison: The generation model of LAC is trained on the Mixamo dataset to have an action composition ability before the contrastive learning. We compare the motion retargeting accuracy on this dataset. Specifically, we randomly split the training and

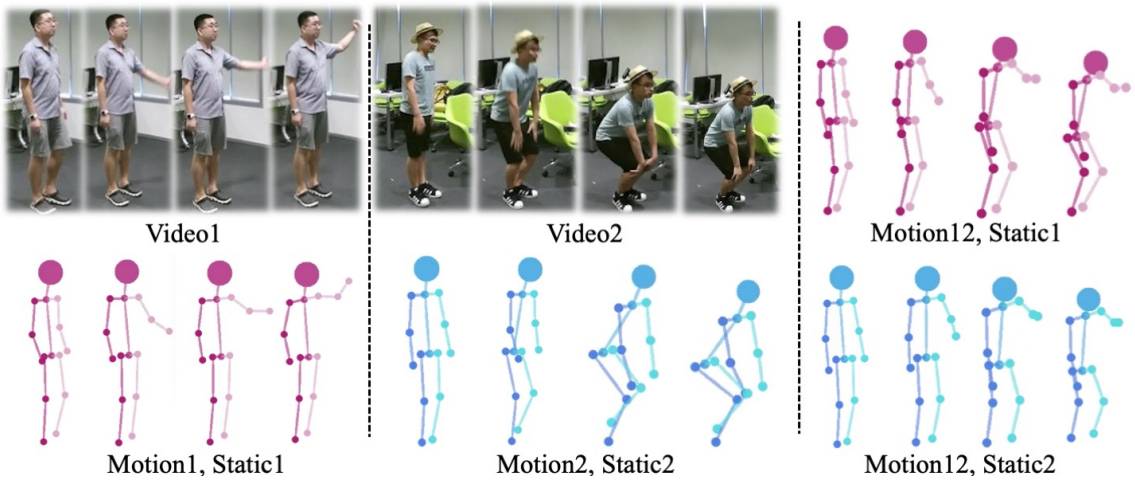

Fig. 4.3 **Motion composition visualization.** The input pair of videos and corresponding skeleton sequences (left) have simple motions. The generated skeleton sequences (right) are composed by both motions while keeping their respective viewpoint and body size ('Static') invariant.

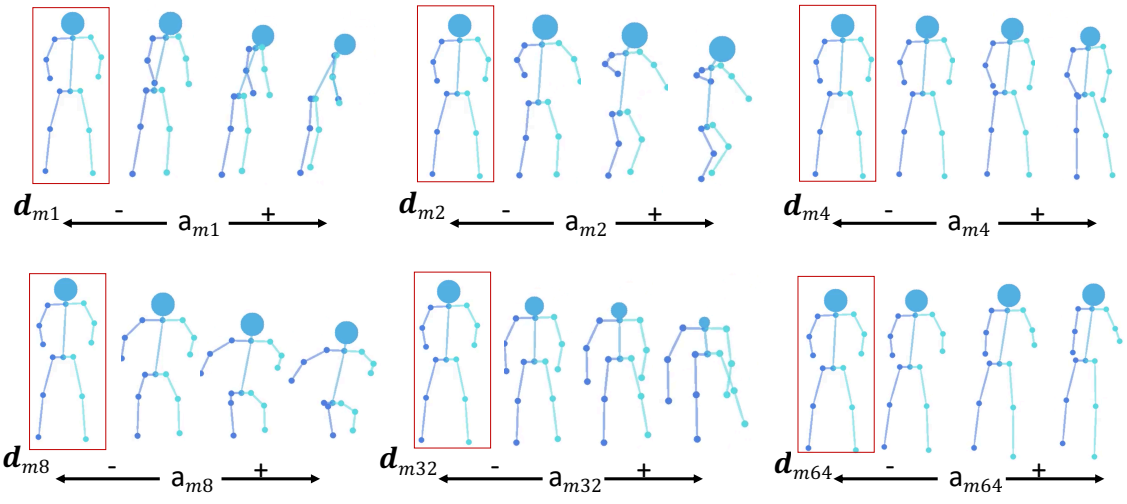

Fig. 4.4 **Linear manipulation of six 'Motion' directions** in $\mathbf{D}_v$ on a skeleton sequence. Results indicate that each direction represents a meaningful motion transformation from a 'reference pose' marked in red (*e.g.*, $\mathbf{d_{m8}}$ for squat, $\mathbf{d_{m32}}$ for bending over).

Methods	Mean Square Error
NKN [174]	1.51
MotionRetargeting2D [3]	0.96
ViA [203]	0.86
LAC w/ $\mathbf{D}_v$ (Ours)	
size $J = 16$, $K = 144$	1.23
size $J = 32$, $K = 128$	1.02
size $J = 64$, $K = 96$	0.88
size $J = 128$, $K = 32$	**0.82**
size $J = 144$, $K = 16$	0.85

Table 4.6 Quantitative comparisons of LAC to other SoTA motion retargeting methods on the Mixamo dataset.

test sets on this dataset and we follow the same setting and protocol described in [3, 203]. We firstly explore how many directions (*i.e.*, the values of J and K) are required in the proposed action dictionary $\mathbf{D}_v$. We empirically test four different values for J from 16 to 144. From results reported in Tab. 4.6, we observe that when using 128 directions (out of all dim=160 directions) for 'Motion', the model achieves the best reconstruction accuracy and outperforms SoTA methods [174, 3, 203]. Hence, we set J=128 and K=32 for all other experiments.

Motion Direction Interpretation and Visualization: We visualize an example of motion composition inference of two videos. Fig. 4.3 demonstrates that 'Static' and 'Motion' are well disentangled and the high-level motions can be effectively composed by decoding the linear combination of both latent 'Motion' components learned by the proposed LAD. To further understand what each direction in $\mathbf{D}_v$ represents, we proceed to visualize $\mathbf{d}_{\mathbf{m}i}$. We generate skeletons for a single input skeleton sequence using its disentangled 'Static' features $\mathbf{r}_c$ combined by different $\mathbf{r}_m$ respectively obtained by a linearly grown a_{mi} on its corresponding 'Motion' directions $\mathbf{d}_{\mathbf{m}i}$ (see Fig. 4.4 for visualization of six directions), where other magnitudes on directions except $\mathbf{d}_{\mathbf{m}i}$ are set to 0. We find that each direction represents a basic high-level motion transformation (*e.g.*, $\mathbf{d}_{\mathbf{m}32}$ represents bending over) and the corresponding magnitude represents the range of the motion. All motion transformations start from a fixed 'reference pose', regardless of original motions of the input skeleton sequences. Such a 'reference pose' can be considered as a normalized form of the given skeleton sequence. In such a learning strategy, complex motions can be combined and the motion diversity can be controlled in an interpretive way by latent space manipulation.

Toyota Smarthome Untrimmed	CS (%)	CV (%)
L0: Base: w/o LAC	29.8	13.8
L1: +Motion Composition		
Number of motions=2	33.8	21.9
Number of motions=3	32.1	21.1
L2: +Frame-level Contrast		
Temporal sample rate=2	34.0	22.5
Temporal sample rate=4	**34.1**	**22.8**
Temporal sample rate=8	33.7	22.0

Table 4.7 mAP on Toyota Smarthome Untrimmed CS and CV for showing impacts of two types of hyper-parameter for modulating the generated skeleton sequences.

4.5.4 Ablation Study

To understand the contribution of the two individual components of LAC, we conduct ablation experiments on our main target fine-grained dataset TSU, with self-supervised pre-training and fine-tuning protocol.

Impact of Action Composition: We start from a baseline model [202] that is pre-trained on the trimmed dataset (*i.e.*, Posetics) in a general contrastive learning strategy [66] without using composable motions and frame-level contrast for action segmentation. The results in Tab. 4.7 (see L0) suggest that the visual encoder has a weak capability to learn features on top of an untrimmed skeleton sequence without learning a composable action representation. We then perform the self-supervised training on Posetics (in only the video space) with composable motions from different number of motions. As daily living videos contain in average two co-occurring actions [34], combining motions from two skeleton sequences in the pre-training stage can significantly improve the representation ability of the visual encoder and better generalize to real-world untrimmed action segmentation tasks (see Tab. 4.7 L1). Such number can simply be changed to adapt to different target datasets.

Impact of Frame-wise Contrast: To validate that frame-wise contrastive learning can further improve the fine-grained action segmentation tasks, we additionally maximize the per-frame similarity between the positive samples. We also select different uniform temporal sampling rates to reduce the redundant computational cost instead of using all the frames. The results in Tab. 4.7 L2 suggest that frame-wise contrast with uniformly sampling every 4 frames is the most effective to improve the action segmentation accuracy.

Dataset	TGM [130]	SD-TCN [34]	LAC (Ours)
TSU-CS(%)	25.6	24.4	**33.2**
TSU-CV(%)	13.9	20.8	**21.7**
Charades(%)	9.1	8.7	**21.4**
PKU-MMD(%)	87.3	87.5	**91.0**

Table 4.8 Fine-tuning results (*i.e.*, Frame-level mAP on TSU and Charades and Event-level mAP on PKU-MMD) with individual pre-training only on the target action segmentation datasets for further comparison with SoTA methods.

4.5.5 Further Discussion

Transfer Learning vs. Self Pre-training: Our target is to train a generic skeleton encoder that can fit different downstream tasks. Hence, like current RGB-based methods using large-scale dataset such as Kinetics [18, 17] for pre-training, our model is pre-trained on the large-scale Posetics dataset to learn a generic skeleton representation. Such a representation can be transferred onto different downstream tasks without the need for individual pre-training. This is a very effective practice for action segmentation models. To demonstrate the advantage of transfer-learning and to further compare LAC with SoTA methods, we here compare LAC with [130, 34] in Tab. 4.8 with self pre-training, *i.e.*, *solely self-supervised pre-training the encoder on the tested dataset* (on TSU, PKU-MMD CS-IoU@0.1 and Charades) using the proposed contrastive module without additional data and without action labels. The results show that, without extra training data, LAC can still outperform previous models [130, 34], as in the second stage, LAC adopts end-to-end fine-tuning to refine the visual encoder, which is more effective than using temporal modeling on the pre-extracted features [130, 34]. Moreover, current untrimmed datasets are not large enough, thus the generated actions have less diversity, so the representation ability of the skeleton encoder is less impressive than pre-training on Posetics.

4.6 Experiments and Analysis on ViA

We conduct extensive experiments to evaluate ViA. Firstly, we compare ViA against the state-of-the-art self-supervised models on the large-scale pre-training dataset **Posetics**, by linear evaluation Secondly, we study the generalizability of ViA to quantify the performance improvement obtained by transfer-learning on the target 2D datasets (*i.e.*, **Toyota Smarthome, UAV-Human** and **Penn Action**) as well as 3D datasets (*i.e.*, **NTU-RGB+D 60** and **NTU-RGB+D 120**) after pre-training on Posetics. Thirdly, we evaluate the quality of the motion

(*i.e.*, action) generated by the retargeting module on the synthetic dataset **Mixamo**. Finally, we provide an exhaustive ablation study of ViA.

4.6.1 Training Details

We adopt UNIK [202] as the visual encoder with the same hyper-parameter settings as [202]. For self-supervised pre-training on Posetics, we adopt a learning rate of 5×10^{-4} for 300 epochs. For downstream action classification, we use an initial learning rate of 0.1 for 50, 50 and 30 epochs with step LR decay with a factor of 0.1 at epochs {30, 40}, {30, 40} and {10, 20} for Smarthome, UAV-Human and Penn Action respectively. Weight decay is set to 1×10^{-4} for final models. For Posetics, Smarthome, UAV-Human and Penn Action, we randomly choose 150, 400, 150 and 100 frames respectively for each training epoch and all frames for test. 2D skeleton inputs are pre-processed with normalization and centering following [125]. As we have both 2D and 3D skeleton data on Posetics, we can re-implement another state-of-the-art approach [164] which also requires 3D data for comparison. As recent skeleton-based action recognition methods [145, 105, 202] adopt two-stream fusion to improve the classification accuracy, we also use a two-stream fusion for fair comparison, where a separate model with identical architecture is trained using the Joint and Bone features. The Softmax scores from the two models are summed to obtain final prediction scores.

4.6.2 Evaluation on Self-supervised Action Classification

Our objective is to improve action recognition performance on 2D skeleton datasets by learning an action representation on a sufficiently large dataset. Hence, in this section, we evaluate ViA on self-supervised action classification (*i.e.*, linear evaluation) on the large-scale Posetics dataset and then compare ViA with state-of-the-art approaches.

Comparison with State-of-the-art (SoTA). For fair comparison, we re-implement recent state-of-the-art skeleton-based action representation learning approaches [164, 97, 201] on the Posetics dataset using 2D skeleton data. Results are depicted in Tab. 4.9 (top): ViA is more effective when compared to 3D-based methods [97] applied onto 2D real-world datasets. Intuitively, we think that the variation of the subject body sizes and of the viewpoints might weaken the robustness of the SoTA embedding networks. In contrast, ViA encourages similarity of the representation for actions performed by different subjects under different viewpoints. This shows that our model is more effective and robust to real-world videos. Compared to previous view-invariant embedding approaches [164] based on single frame,

Methods	Posetics	
	Top-1(%)	Top-5(%)
Linear (Baseline)	8.2	21.4
Pr-ViPE [164]	17.2	35.3
OR-VPE [201]	14.6	31.2
3s-CrosSCLR [97]	18.8	38.1
ViA (Ours)	**20.7**	**40.1**
TCNs [82]	34.0	57.2
ST-GCN [197]	43.3	67.3
2s-AGCN [145]	47.0	70.8
Res-GCN [157]	46.7	70.6
MS-G3D Net [105]	47.1	70.0
UNIK [202]	47.6	71.3
ViA (Ours ft.)	**48.0**	**72.6**

Table 4.9 Comparison of Top-1 and Top-5 classification accuracy with state-of-the-art **unsupervised methods (top)** on Posetics. **Fully-supervised results (bottom)** with fine-tuning (reported as ft.) are also reported for reference.

our method considering temporal features is more robust for action recognition. In Tab. 4.9 (bottom) we compare fine-tuning results of ViA to other supervised methods [82, 197, 145, 157, 105, 202] that are trained without representation learning (*i.e.*, training from scratch). Compared to the UNIK backbone model used in our work [202], the pre-training provides minor improvement, as the training data (*i.e.*, Posetics) is sufficiently large. However, when transferring ViA onto smaller benchmark datasets, the impact of representation learning is significant (see Sec. Evaluation on Transfer-learning).

4.6.3 Evaluation on Transfer-learning

In this section, we study the transfer ability of ViA by both linear evaluation and fine-tuning evaluation with self-supervised training on Posetics. We transfer the model onto three 2D skeleton action classification benchmarks *i.e.*, Toyota Smarthome, UAV-Human and Penn Action with no additional pre-training. As Smarthome and UAV-Human mainly focus on the cross-subject and cross-view challenges, the results measure the view- and subject-invariance of the 2D action representation of ViA models. We also report the results with supervised pre-training for reference (*i.e.*, we add a classifier at the end of the visual encoder and adopt cross entropy loss using action labels during training).

Methods	NTU-60		NTU-120	
	CS(%)	CV(%)	CS(%)	CSet(%)
SeBiRe [120]	-	79.7	-	-
CrosSLR [97]	77.8	83.4	67.9	66.7
Colorization [206]	75.2	83.1	-	-
ViA (Ours)	**78.1**	**85.8**	**69.2**	**66.9**
W/o pre-training	86.5	94.6	80.1	84.5
Self-supervised pre-training	**89.6**	**96.4**	**85.0**	**86.5**

Table 4.10 Comparison with previous self-supervised state-of-the-art by **linear evaluation (top)** on NTU-RGB+D 60 and NTU-RGB+D 120. Transfer learning results by **fine-tuning (bottom)** are also reported for reference.

Methods	Pre-training	Toyota Smarthome				UAV-Human			Penn Action	
		#Params	CS(%)	CV1(%)	CV2(%)	#Params	CS1(%)	CS2(%)	#Params	Top-1(%)
Random init.	Scratch	7.97K	24.6	17.2	20.7	39.85K	3.8	4.1	3.85K	29.8
Supervised	Posetics w/ labels	7.97K	51.9	35.4	52.2	39.85K	32.9	56.1	3.85K	97.3
Self-supervised	Posetics w/o labels	7.97K	**49.5**	**33.6**	**52.6**	39.85K	**29.5**	**46.7**	3.85K	**90.2**
Random init.	Scratch	3.45M	63.1	22.9	61.2	3.45M	39.2	67.3	3.45M	94.0
Supervised	Posetics w/ labels	3.45M	64.5	36.1	65.2	3.45M	42.6	69.5	3.45M	98.0
Self-supervised	Posetics w/o labels	3.45M	**64.0**	**35.6**	**65.4**	3.45M	**41.3**	**68.5**	3.45M	**97.7**
Previous SoTA	-	[202]				[26]			[164]	
	-	-	63.1	22.9	61.2	-	38.0	67.0	-	97.5
ViA (Ours)	-	-	**64.5**	**36.1**	**65.4**	-	**42.6**	**69.5**	-	**98.0**

Table 4.11 Transfer learning results by **linear evaluation (top)** and **fine-tuning (middle)** on Smarthome, UAV-Human and Penn Action with self-supervised pre-training on Posetics compared to Baseline (random initialization). Results with supervised pre-training and **previous state-of-the-art (bottom)** are also reported.

Linear Evaluation. Tab. 4.11 (top) shows the linear results on the three datasets. This evaluates the effectiveness of transfer-learning with fewer parameters (only the classifier is trained) compared to classification from random initialization. The results suggest that the weights of the model can be well pre-trained without action labels, providing a strong transfer ability especially on smaller benchmarks (*e.g.*, +31.9% Smarthome on CV2 and +70.4% on Penn Action) and the pre-trained visual encoder is generic enough to extract meaningful action features from skeleton sequences.

Fine-tuning. Tab. 4.11 (middle) shows the fine-tuning results, when the whole network is re-trained. These results suggest that pre-training can improve upon previous SoTA [202] with no pre-training. The self-supervised pre-trained model also performs competitively compared to supervised pre-trained models. From these results we conclude that collecting a large-scale video dataset, without the need of action annotation, can be beneficial to downstream tasks, especially when using our proposed view-invariant ViA for the 2D action classification

Methods	Pre-training	Training data	Toyota Smarthome			UAV-Human		Penn Action
			CS(%)	CV1(%)	CV2(%)	CS1(%)	CS2(%)	Top-1 Accuracy(%)
Random init. [202]	Scratch	5%	22.9	5.6	33.7	10.9	10.4	32.4
Self-supervised	Posetics w/o labels	5%	**38.6**	**16.8**	**42.6**	**21.7**	**33.3**	**65.8**
Random init. [202]	Scratch	10%	33.8	8.5	39.5	17.8	25.6	39.8
Self-supervised	Posetics w/o labels	10%	**45.3**	**22.7**	**46.6**	**31.0**	**43.7**	**85.2**

Table 4.12 Transfer learning results by **fine-tuning** on all benchmarks of Smarthome, UAV-Human and Penn Action with randomly selected **5% (top)** and **10% (bottom)** of labeled training data.

task (*e.g.*, +12.7% on Smarthome CV1 and +4.2% on CV2). Furthermore, we compare our fine-tuning results to other SoTA skeleton-based supervised approaches [202, 26, 164]. The results in Tab. 4.11 (bottom) show that ViA outperforms all previous approaches on all the three real-world datasets.

Training with Fewer Labels. In some real-world applications, labeled data may be lacking, which makes it challenging to train models with good performance. To evaluate ViA in such cases, we pre-train with Posetics and then fine-tune the visual encoder with 5% and 10% of the labeled data. As shown in Tab. 4.12, without pre-training, the accuracy of the baseline [202] significantly decreases with the amount of training data. In contrast, ViA still achieves good performance on all three datasets.

3D Skeleton Action Classification. As ViA can be simply extended to take 3D skeleton sequence as input, we further analyze the transfer ability of ViA onto 3D skeleton action recognition tasks. We firstly compare [97], [201] and ViA on Posetics using officially provided 3D data (we get 17.1%, 12.9% and 19.3%, respectively for linear evaluation). These results are lower than related results in Tab. 4.9 using 2D data. We argue that, although 3D skeletons are more robust to the view variation, 2D skeletons extracted from images or videos tend to be more accurate than extracted 3D skeletons. In contrast, laboratory datasets (*e.g.*, NTU-RGB+D 60 & 120) provide 3D skeleton data obtained by RGBD sensors that have a higher quality than the ones provided by 2D data. To study the impact of action representation learning, we also transfer the ViA pre-trained on Posetics (3D skeletons) without action labels onto NTU-RGB+D-60 and NTU-RGB+D-120 by fine-tuning. The action recognition performance can still be improved (*e.g.*, +4.9% on NTU-RGB+D-120 CS, see Tab. 4.10 (bottom)). To compare with other recent self-supervised methods [120, 97, 206], we follow the same pre-training setting and linear evaluation protocol and report state-of-the-art accuracy (see Tab. 4.10 (top)).

Methods	Sup.	Unsup.
NKN [174]	1.51	-
MotionRetargeting2D [3]	0.96	2.56
ViA w/o $\mathbf{D}_c$ (Ours)	2.42	-
ViA w/ $\mathbf{D}_c$ (Ours)		
size $K = 2$	1.16	-
size $K = 64$	0.89	-
size $K = 32$	**0.86**	**2.47**

Table 4.13 Quantitative comparisons of Mean Square Error (MSE) show that our framework outperforms other SoTA motion retargeting methods on Mixamo.

4.6.4 Evaluation on Cross-view Motion Retargeting

As motion retargeting is our pretext task, we evaluate here the proposed LMD mechanism of ViA by motion retargeting performance. We randomly split training and test sets on the Mixamo dataset and we follow the same setting and protocol described in [3]. As previous SoTA [174, 3] are supervised approaches, for fair comparison, we also train ViA using cross reconstruction loss on [75] in a supervised manner. To validate the impact of proposed $\mathbf{D}_c$, we learn $\mathbf{r}_c$ by decomposing $\mathbf{r}_{m,c}$ on a pre-defined and fixed subspace without the learnable $\mathbf{D}_c$. From the evaluation results reported in Tab. 4.13, we observe that in the absence of $\mathbf{D}_c$, the model fails to generate high-quality skeletons, which proves the effectiveness of $\mathbf{D}_c$. Then we conduct an ablation analysis on the size of $\mathbf{D}_c$. The results suggest that motion retargeting by ViA (w/ $\mathbf{D}_c$ in including 32 directions) performs the best and achieves SoTA accuracy. We also report unsupervised results by cycle consistency learning.

Qualitative Evaluation. To demonstrate that the 'View/Subject' and 'Motion' are well disentangled by the proposed framework, we visualize an example of motion retargeting inference of two Penn Action's videos (see Fig. 4.5). Then we visualize the representations of all the skeletons on Mixamo with t-SNE (see Fig. 4.7 with both supervised and unsupervised motion retargeting). Qualitative results validate that the 'View/Subject' and 'Motion' parts of 2D skeleton sequences have been effectively disentangled. To further understand the learned 'Motion' features, we generate skeletons for each single input sequence with only $\mathbf{r}_m$ (see Fig. 4.6 (b)), and then with $\mathbf{r}_m$ combined by different $\mathbf{r}_c$ obtained by a linearly grown A_c (see Fig. 4.6 (c)). We observe that $\mathbf{r}_m$ represents the motion in a 'canonical view', regardless of original views of the input skeleton sequences. As such a 'canonical view' can be considered as a normalized form of the given skeleton sequence, learning transformations between

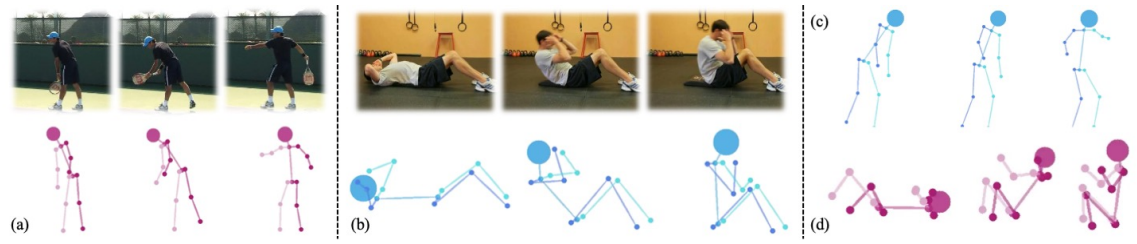

Fig. 4.5 Qualitative results on **Motion Retargeting.** (a) and (b) are the input pair of videos and corresponding 2D skeleton sequences. (c) is the generated 2D skeleton sequence that represents the character of (b) performing the motion in (a) while maintaining the viewpoint and body size invariance. (d) is the generated 2D skeleton sequence that represents the character of (a) performing the motion in (b).

generative sequence and source sequence using $\mathbf{D}_c$ and A_c is considerably more efficient than direct generation once the 'canonical view' is fixed.

4.6.5 Ablation Study

To understand the contribution of each loss function in ViA, we conduct ablation experiments on Smarthome CV2 with fine-tuning protocol. To perform more studies on the characteristics of view-invariant representations, we additionally set a Cross-view (CV) action recognition protocol on the Mixamo dataset (*i.e.*, Top-1 classification accuracy) using two different 2D skeleton projections generated by random 3D rotations of the same action for 2D cross-view evaluation. We start from a baseline model that has been previously pre-trained on the synthetic dataset (*i.e.*, Mixamo) using motion retargeting annotations for cross-character reconstruction. Already this visual encoder has a strong capability to embed the 2D skeleton sequence into a view-invariant representation. The results in Tab. 4.14 (see L0) suggest that the generalizability is hindered by the lack of action diversity in the synthetic training dataset if directly transferring the baseline visual encoder for action classification. Therefore, from the full results in Tab. 4.14, we infer that additional self-supervised training on Posetics can improve the real-world generalizability. Specifically, a **self reconstruction loss (L1)** can help the visual encoder learn the global characteristics of the real-world data and thus facilitate the classification. The **cycle reconstruction loss (L2)** and the **triplet loss (L3)** aim at maximizing the embedding similarity between the same action performed from two viewpoints, while minimizing the embedding similarity between different actions. These losses are instrumental in extracting a more generic representation for the downstream action classification task. Finally, The **velocity loss (L4)** contributes to minor boosts.

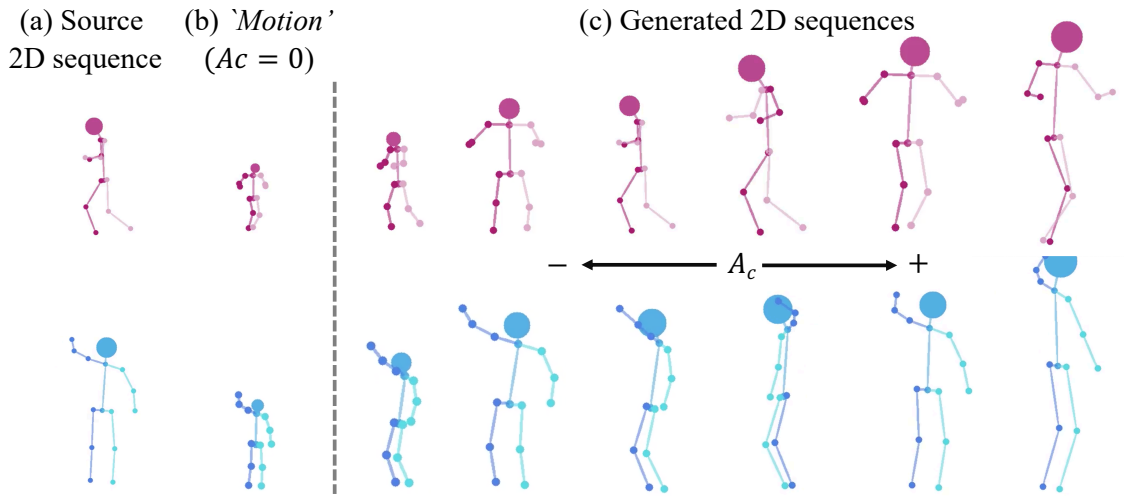

Fig. 4.6 Qualitative results on **2D Motion Generation.** Given a source skeleton sequence, we can generate multiple sequences by latent space manipulation on disentangled 'Motion' and 'Character' magnitudes (A_c).

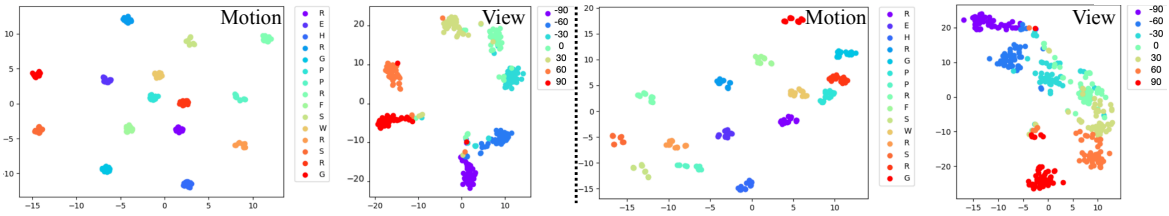

Fig. 4.7 Skeleton representations (marked by different colors with 'Motion' and 'View') on Mixamo with ViA by supervised (left) and unsupervised (right) motion retargeting.

4.7 Conclusion

In this chapter, we present LAC, a novel self-supervised action representation learning framework for the setting of skeleton action segmentation. We show that high-level motions of skeleton sequences can be learned and linearly combined using an orthogonal basis in the latent space. Moreover, we augment a contrastive learning module to better extract frame-level features, in addition to the generated composable skeleton sequences. Our experimental analysis confirms that a skeleton visual encoder that extracts such skeleton representation is able to boost downstream action segmentation tasks.

Furthermore, we presented ViA, a generic framework aimed at learning view-invariant skeleton action representation via Latent Action Decomposition. We showed that self-supervised motion retargeting with contrastive learning can be an effective pretext task to

Methods	$\mathscr{L}_{self}$	$\mathscr{L}_{cycle}$	$\mathscr{L}_{trip}$	$\mathscr{L}_{vel}$	**Smarthome** CV2(%)	**Mixamo** CV(%)
L0: Baseline					61.7	71.7
L1: +Self	✓				62.9	76.5
L2: +Cycle	✓	✓			63.8	82.5
L3: +Trip	✓	✓	✓		65.0	85.8
L4: +Vel (Full)	✓	✓	✓	✓	**65.4**	**87.2**

Table 4.14 Ablation study of ViA on Smarthome CV2 and Mixamo CV with transfer learning (fine-tuning).

learn view-invariant action representation for real-world 2D skeleton sequences. Experimental analysis confirmed that a visual encoder extracting such representation on large-scale datasets such as Posetics significantly boosts the performance when transferred onto downstream target datasets for cross-subject and cross-view action classification tasks.

Future work will extend our generative approach to RGB videos, in order to improve the capturing of the object information, which can be crucial and complementary to the skeleton-based model.

Chapter 5

Time-aware Video Action Representation Learning

Besides learning human skeleton motion representations, visual representations from RGB videos are also important to extract action features (*e.g.*, human-object interaction). In this chapter, we introduce our exploration on RGB-based action representation learning approach.

Self-supervised video representation learning aimed at maximizing similarity between different temporal segments of one video, in order to enforce feature persistence over time. This leads to a loss of pertinent information related to temporal relationships, making actions such as 'enter' and 'leave' indistinguishable. To mitigate this limitation, we propose Latent Time Navigation (LTN), a time-parameterized contrastive learning strategy that is streamlined to capture fine-grained motions. Specifically, we maximize the representation similarity between different video segments from one video, while maintaining their representations *time-aware* along a subspace of the latent representation code including an orthogonal basis to represent temporal changes. Our extensive experimental analysis suggests that learning video representations by LTN consistently improves the performance of action classification in fine-grained and human-oriented tasks (*e.g.*, on Toyota Smarthome dataset). In addition, we demonstrate that our proposed model, when pre-trained on Kinetics-400, generalizes well onto the unseen real world video benchmark datasets UCF101 and HMDB51, achieving state-of-the-art performance in action recognition. This work has been published in AAAI Conference on Artificial Intelligence (AAAI) [205] in 2023.

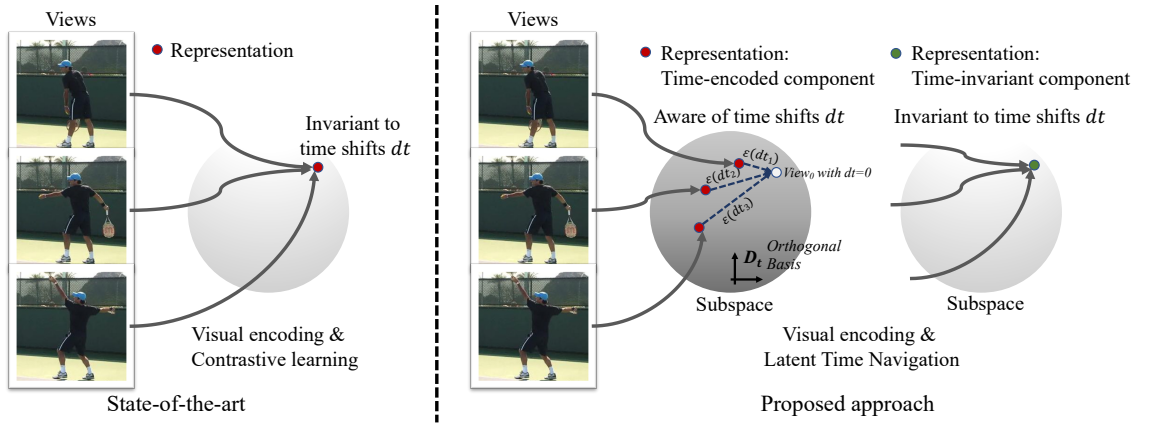

Fig. 5.1 Current methods (left) leverage on contrastive learning to maximize representation similarities of multiple positive views (segments with time spans and data augmentation) of the same video instance to represent them as a consistent representation. To further improve the representation capability for fine-grained tasks without losing important motion variance, our approach (right) incorporates a time-parameterized contrastive learning (LTN) to keep the video representations aware to time shifts (starting time) in a decomposed time-encoded subspace.

5.1 Introduction

Contrastive learning [61] is a prominent variant in learning self-supervised visual representations. The associated objective is to minimize the distance between latent representations of positive pairs, while maximizing the distance between latent representations of negative pairs. For instance, a visual encoder aims at learning the invariance of multiple *views* of a scene, which constitute positive pairs, by extracting generic features of images [6, 15, 21, 59, 66, 71, 78, 167, 192] or videos [52, 62, 73, 84, 97, 100, 123, 201, 203, 163]. Then, the trained visual encoder can be transferred to other downstream tasks.

Remarkable results have been reported by augmentation-invariant contrastive learning. In this context, contrastive learning methods enable the visual encoder to find compact and meaningful image representations, invariant to data augmentation. The latent representation of two augmented views of the same instance are enforced to be similar via contrastive learning. In *image-based tasks*, a common augmentation method relates to random cropping [21, 192]. When extending this idea to *videos*, which are endowed with additional temporal information, cropping in the spatial dimension [84] is not sufficient for training an effective visual encoder. Therefore, recent works [52, 100, 163] sample different views with a *temporal shift*, learning representations that are invariant to time changes. However, for

downstream tasks involving temporal relationships, a representation invariant to temporal shifts might omit valuable information. For instance, in differentiating actions such as 'enter' and 'leave' the temporal order is fundamental. Hence, a trained visual encoder remains a challenge in handling downstream video understanding tasks such as fine-grained human action recognition [36, 56, 101].

Motivated by the above, we propose Latent Time Navigation (LTN), a time parameterization scheme streamlined to learn time-aware representations on top of the contrastive module. As illustrated in Fig. 5.1, deviating from current contrastive methods [52, 66, 167, 192] which directly maximize the similarity between representations obtained from the visual encoder for positive samples, LTN encompasses the following steps. Firstly, we decompose a subspace (*i.e.*, a learnable orthogonal basis and associated magnitudes) from the latent representation code for the video segment, namely 'time-encoded component', to do with temporal changes (*e.g.*, changes in appearances, motion, object locations). The other subspace ('time-invariant component') has to do with invariant information. Subsequently, we embed the *time shift value* used for generating data view into a high-dimensional vector as the magnitudes of the directions in the orthogonal basis and then we encode this time information into the 'time-encoded component' by linear combination of the orthogonal basis and the magnitudes. Finally, we conduct contrastive learning on the entire time-parameterized representations in order to maximize the similarity between positive pairs along the 'time-invariant component', while maintaining their representations *time-aware* along the 'time-encoded component'. We note that LTN incorporates time information for video representations and therefore is able to model subtle motions within an action. Consequently, the time-aware representation obtained from the trained visual encoder generalizes better to unseen action recognition datasets, especially to our target human-oriented fine-grained action classification dataset [36].

In summary, the contributions of this chapter include the following. (i) We propose Latent Time Navigation (LTN) to parameterize the time information (used for generating data views) on top of contrastive learning, in order to learn a *time-aware* video representation. (ii) We demonstrate that LTN can effectively learn the consistent amount of temporal changes with the video segments on the decomposed 'time-encoded components'. (iii) We set a new state-of-the-art with LTN on the real world dataset (*e.g.*, Toyota Smarthome) for fine-grained action recognition with self-supervised action representation learning. (iv) We demonstrate that our proposed model, when pre-trained on Kinetics-400 dataset, generalizes well to unseen real-world video benchmarks (*e.g.*, UCF101 and HMDB51) with both linear evaluation and fine-tuning.

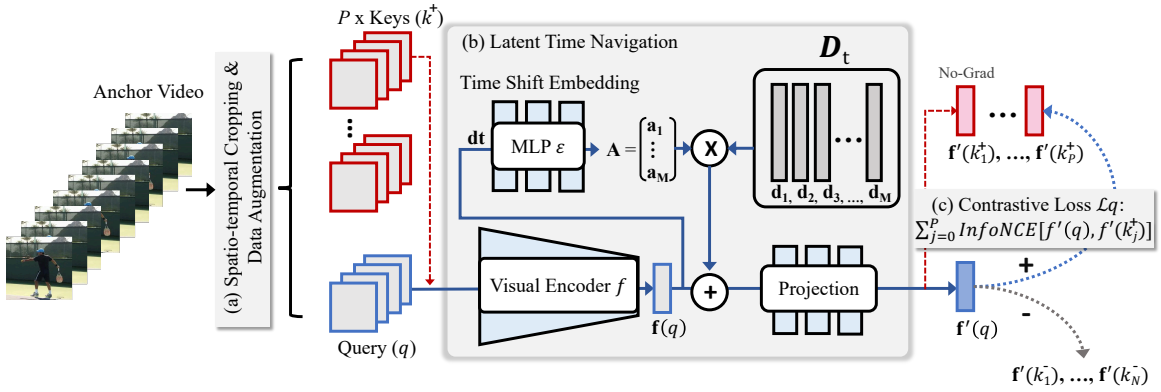

Fig. 5.2 Overview of the proposed LTN framework. At each training iteration, given an input video, (a) a query clip (q) and multiple positive key clips ($k_1^+, k_2^+, ..., k_P^+$) are generated by data augmentation with different temporal shifts **dt**. All clips are then fed to a visual encoder that extracts spatio-temporal features for each clip. To learn time-aware representations for query and key clips, (b) we first pre-define a learnable orthogonal basis $\mathbf{D}_t$ ($\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_M}$) that represents the 'time-encoded component'. The video representations are expected to be time-aware along $\mathbf{D}_t$ in the training stage. To do so, we transform each query and key video representation (*i.e.*, $\mathbf{f}(q)$, $\mathbf{f}(k_p^+)$) by a linear combination of $\mathbf{D}_t$ and associated magnitudes learned from its time shift **dt** to a time-blended position (*i.e.*, $\mathbf{f'}(q, \mathbf{dt}_q)$, $\mathbf{f'}(k_p^+, \mathbf{dt}_p)$, abbreviated as $\mathbf{f'}(q)$, $\mathbf{f'}(k_p^+)$). Finally, we conduct (c) contrastive learning on top of **f'**, so that the learned representation from the visual encoder can maintain temporal awareness.

5.2 Related Work

Contrastive Learning: Contrastive learning and its variants [6, 15, 21, 59, 66, 71, 78, 167, 192] have established themselves as a pertinent direction for self-supervised representation learning for a number of tasks due to promising performances. Recent video representation learning methods [52, 73, 84] are inspired by image techniques. The objective of such techniques is to encourage representational invariances of different views (*i.e.*, positive pairs) of the same instance obtained by data augmentation, *e.g.*, random cropping [21, 192], rotation [116], while spreading representations of views from different instances (*i.e.*, negative pairs) apart. To further improve the representation capability, CMC [116] scaled contrastive learning to any number of views. MoCo [66] incorporated a dynamic dictionary with a queue and a moving-averaged encoder. To omit a large number of negative pairs, BYOL [59] and SwAV [15] were targeted to solely rely on positive pairs. However, these methods miss a crucial time element when they are straightforwardly applied to the *video* domain with views generated by *image* data augmentation technique. In our work, we adopt recent contrastive

learning frameworks [59, 66] and we focus on learning time-aware representations for videos by latent spatio-temporal decomposition and navigation in the representation space.

Self-supervised Video Representation Learning: Approaches for self-supervised video representation learning exploit spatio-temporal pretext tasks from numerous unlabeled data. Towards effective extraction of the pertinent motion information in the time dimension, a number of temporal pretext tasks were proposed, *e.g.*, pixel-level future generation [113, 159, 176, 177] and jigsaw-solving [80]. Additionally, in order to facilitate the learning process, numerous works focused on learning representations in a more abstract space including temporal order [117, 195] or arrow [190] prediction of video frames, future prediction [175], speed prediction [9], motion prediction [39] and a combination of these tasks [8]. These methods are highly constrained by the limited quality of pretext tasks. Recently, video contrastive learning methods [73, 84] have obtained promising results and a large-scale study [52] has been conducted to compare state-of-the-art image-based contrastive methods [15, 21, 59, 66] on videos using spatio-temporal cropping, color jitters and Gaussian blur data augmentation techniques to generate multiple video views. Further, to improve representation performance, [40, 74, 134] focused on view generation techniques, *e.g.*, context-motion decoupling [74], foreground-background merging [40], global and local sampling across space and time [134]. In addition, some specific designs are incorporated in spatio-temporal representation learning including Gaussian probabilistic representations [123], skeleton contrastive learning [97, 201, 37] and muti-modal learning with audio [10, 45, 124, 135, 147, 193] or with optical flow [62, 100]. Such contrastive methods aimed at learning video representations invariant to time shift. However, motion significantly changes with time shifts, leading to poor performance on downstream fine-grained action recognition tasks that highly rely on the motion variance. To address this issue, CATE [163] proposed to parameterize data augmentation relying on an additional Transformer head prior to contrastive learning. It demonstrated that awareness of the temporal data augmentation is particularly instrumental in fine-grained action recognition tasks. Deviating from CATE that shifts the entire visual representation along all dimensions by the time-shift values, even for the action with small motion variances, we study variant time-parameterization strategies and propose to encode the time-shift values partially on certain orthogonal directions instead of on the entire visual representation. With our proposed LTN, the impact of time can be video specific and controlled by the number of the orthogonal directions so that the visual encoder can better capture motions.

5.3 LTN: Latent Time Navigation

In this section we introduce our Latent Time Navigation (LTN) framework. We start with the overall architecture, then we proceed to describe the design strategies focusing on time parameterization that forces the learned video representation to be aware of motion variances.

5.3.1 Overall Architecture of LTN

Our objective is to train a generic visual encoder $\mathbf{f}$ to extract accurate spatio-temporal features of video clips. We design our visual encoder to be efficient for downstream fine-grained action recognition tasks. We illustrate the overview of the architecture in Fig. 5.2. To train the visual encoder, a general data augmentation technique including random temporal shifts is applied to generate multiple positive views for a given input video, allowing us to obtain multiple representations from different views. Deviating from previous methods [66, 167], which directly employ contrastive learning for these representations in order to make them invariant to spatio-temporal augmentation, we design an additional time parameterization module to blend temporal augmentation to a 'time-encoded component' prior to contrastive learning. We then perform the contrastive learning for the new time-blended representations in the training stage. The trained visual encoder can thus be aware of time shifts compared to other positive pairs and can capture the important motion variances of videos for improving fine-grained action recognition tasks.

View Generation and Embedding: Following the study [52], we first spatio-temporally crop a segment by randomly selecting a segment and cropping out a fixed-size box from the same video instance. We then pull together image-based augmentations including random horizontal flip, color distortion and Gaussian blur following [21, 66] to generate positive views of the input video at each training iteration. As demonstrated in [52], multiple positive samples with large time spans between them are beneficial in downstream performance. In our work, we sample a query clip noted as q and multiple positive keys with large time spans, noted as $k_1^+, ..., k_P^+$ (see Fig. 5.2 (a)). We utilize a 3D-CNN network [63] as the visual encoder to obtain dim-dimensional representations of all clips (*i.e.*, $\mathbf{f}(q), \mathbf{f}(k_1^+), ..., \mathbf{f}(k_P^+) \in \mathbb{R}^{1\times dim}$).

Awareness of Time in Latent Space: Large time spans between positive samples may depict significant changes in human motion. When directly matching $\mathbf{f}(q)$ to all positive pairs, the corresponding representations may lose pertinent motion variance caused by time shifts. This could compromise the accuracy of downstream tasks related to fine-grained

human motion (*e.g.*, classification of 'Leave/Enter', 'Stand up/Sit down'). Hence, we expect positive pairs to be partially similar to each other (due to static object, scene) while also partially aware of their time shifts to preserve temporal dynamic information (e.g., changes in motion). To do so, we design several time parameterization methods (see Sec. Time Parameterization in Latent Space) to encode the time shift value (denoted as $\mathbf{dt_q}$ for the query clip q) used for data augmentation to a part (several orthogonal directions) of the visual representation while keep the remaining part unchanged. Such time-encoded pretext representation of q and each positive key can be computed and denoted as $\mathbf{f'}(q, \mathbf{dt_q})$ and $\mathbf{f'}(k_p^+, \mathbf{dt_p})$. We then maximize the mutual information between the pretext representations $\mathbf{f'}(q, \mathbf{dt_q})$ and $\mathbf{f'}(k_p^+, \mathbf{dt_p})$ by contrastive learning. The original (target) visual representations from different segments (*e.g.*, $\mathbf{f}(q)$, $\mathbf{f}(k_p^+)$) will be sensitive to time along the time-encoded part after learning and can be transferred onto downstream tasks.

5.3.2 Time Parameterization in Latent Space

We first introduce the latent space decomposition approach to split the representation space into 'time-encoded component' and 'time-invariant components', and then we introduce time encoding which is used as a parameter to transform the visual representation only along the 'time-encoded component' to reach a new time-blended position.

Latent Space Decomposition: To decompose the representation space, we set a learnable orthogonal basis (*i.e.*, a subspace) $\mathbf{D}_t = \{\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_M}\}$ with $M \in [1, dim)$, and $\mathbf{d} \in \mathbb{R}^{dim \times 1}$ to represent the 'time-encoded component', where each vector indicates a basic visual transformation. Due to $\mathbf{D}_t$ entailing an orthogonal basis, any two directions $\mathbf{d_i}, \mathbf{d_j}$ follow the constraint in Eq. 5.1. We implement $\mathbf{D}_t \in \mathbb{R}^{dim \times M}$ as a learnable matrix following [189], and we apply the Gram-Schmidt algorithm during each forward pass in order to satisfy the orthogonality.

$$< \mathbf{d_i}, \mathbf{d_j} >= \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases} \tag{5.1}$$

Time Encoding: We decomposed the 'time-encoded component' $\mathbf{D}_t$ of the video representation from the latent space, to force the model to be aware of temporal variances along $\mathbf{D}_t$ with different time shifts. We propose to encode and parameterize the time shift values $\mathbf{dt}$ for the randomly selected query (and key) segments using their absolute starting point in seconds in the timestamps (*i.e.*, t_{start}). We used absolute time as we aimed at learning the representation of a single segment aware of time shift from a fixed 'reference view' (*i.e.*, the

video beginning).

$$\varepsilon(\mathbf{dt}, \mathbf{f}(q)) = \text{MLP}\left([\text{MLP}(t_{start}), \mathbf{f}(q)]\right). \tag{5.2}$$

Specifically, we encode **dt** into a high-dimensional vector $\varepsilon(\mathbf{dt}, \mathbf{f}(q))$ by simple MLP (see Eq. 5.2), with the purpose of parameterizing the time shift considering different time-blending variants followed by contrastive learning. The time encoder also accepts $\mathbf{f}(q)$ as the input by concatenating it with embedded **dt**, towards learning a video specific encoding. We explore the idea of effective modeling for time shifts by proposing and comparing three time parameterization variants for the transformation from **f** to **f'**. The first approach concerns straightforward linear addition on video representation $\mathbf{f}(q)$ with $\varepsilon(\mathbf{dt}, \mathbf{f}(q))$ (Variant 1). We then develop more efficient variants, which model the 'time-encoded component' more finely by learning the weights (variant 2) or the magnitudes (variant 3) only along the directions in the 'time-encoded component' $\mathbf{D}_t$.

Variant 1. Time-driven Linear Addition We implement $\varepsilon(\mathbf{dt_q}, \mathbf{f}(q)) \in \mathbb{R}^{1\times dim}$ as the offset, from which positive pairs need to be pulled away from the representation 'time-encoded component' to obtain the time-blended representation in the latent space. The linear addition can be described as Eq. 5.3.

$$\mathbf{f'}(q, \mathbf{dt}_q) = \mathbf{f}(q) + \varepsilon(\mathbf{dt_q}, \mathbf{f}(q)) \tag{5.3}$$

Variant 2. Time-driven Attention We then explicitly implement an attention mechanism to learn a set of attention weights for the positive pairs to be driven by $\mathbf{W} \in \mathbb{R}^{1\times M} = \{w_1, w_2, ..., w_M\} = \text{Softmax}\left(\varepsilon(\mathbf{dt_q}, \mathbf{f}(q))\right)$. The attention weights force $\mathbf{f}(q)$ to focus on the specific 'time-encoded component' in $\mathbf{D}_t$ according to different time encoding. This process can be described as follows

$$\mathbf{f'}(q, \mathbf{dt}_q) = \mathbf{f}(q) \cdot (\sum_{i=1}^{M} w_i \cdot \mathbf{d_i}). \tag{5.4}$$

Variant 3. Time-driven Linear Transformation: As shown in Fig. 5.2 (b), we finally propose a linear transformation method to encode the time shift information in the latent 'time-encoded component' $\mathbf{D}_t$. To implement linear transformation along $\mathbf{D}_t$, we learn the coefficient (*i.e.*, magnitude) on each direction of $\mathbf{D}_t$, noted as $\mathbf{A} \in \mathbb{R}^{1\times M} = \{a_1, a_2, ..., a_M\} = \varepsilon(\mathbf{dt}_q, \mathbf{f}(q))$, by the time encoder. This linear transformation is able to enforce time variance and to obtain different representations only along $\mathbf{D}_t$. The final time-blended representation

$\mathbf{f'}(q, \mathbf{dt}_q)$ can be described as follows

$$\mathbf{f'}(q, \mathbf{dt}_q) = \mathbf{f}(q) + \sum_{i=1}^{M} a_i \cdot \mathbf{d_i} = \mathbf{f}(q) + \mathbf{A} \times \mathbf{D}_t^T. \tag{5.5}$$

All proposed time parameterization variants are effective in learning video representations aware of temporal changes and can improve the target downstream tasks by capturing such motion variances. Associated analysis is presented in Sec. Ablation Study, where we compare the three variants on their performance of downstream tasks. We find that the Linear Transformation with an orthogonal basis is the most effective and is beneficial as a generic methodology for learning time-aware spatio-temporal representations.

5.3.3 Self-supervised Contrastive Learning

In this section, we omit the parameterized time of all samples in the notations to simplify formulations (*e.g.*, $\mathbf{f'}(q, \mathbf{dt_q})$ is abbreviated as $\mathbf{f'}(q)$), and we provide details on the contrastive loss function. We apply general contrastive learning (see Fig. 5.2 (c)) to train our visual encoder $\mathbf{f}$ to encourage similarities between the time-blended positive representations, $\mathbf{f'}(q), \mathbf{f'}(k_1^+), ..., \mathbf{f'}(k_P^+)$, and discourage similarities between negative representations, $\mathbf{f'}(k_1^-), ..., \mathbf{f'}(k_N^-)$. The InfoNCE [122] objective is defined as follows

$$\mathscr{L}_q = \sum_{p=1}^{P} \mathscr{L}_{NCE} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\mathrm{Sim}\left(\mathbf{f'}(q), \mathbf{f'}(k_p^+)\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathbf{f'}(q), \mathbf{f'}(k_n^-)\right)}} \Big), \tag{5.6}$$

where P represents the number of positive keys, N denotes the number of negative Keys, and the similarity can be computed as:

$$\mathrm{Sim}(x, y) = \frac{\phi(x) \cdot \phi(y)}{\|\phi(x)\| \cdot \|\phi(y)\|} \cdot \frac{1}{Temp}, \tag{5.7}$$

where $Temp$ refers to the temperature hyper-parameter [192], and ϕ is a learnable mapping function (*e.g.*, an MLP projection head [52]) that can substantially improve the learned representations.

5.4 Experiments and Analysis

We conduct extensive experiments to evaluate LTN on four action classification datasets: **Toyota Smarthome**, **Kinetics-400**, **UCF101** and **HMDB51**. Firstly, we provide experimental results on tested variants, we investigate exhaustive ablations and further analyze on Toyota Smarthome (fine-grained action classification dataset) to better understand the design choices of our proposed time parameterization approaches. Secondly, we compare LTN with the best setting to state-of-the-art methods on all evaluated benchmarks: Toyota Smarthome, UCF101 and HMDB51 without additional training data and with pre-training on Kinetics-400.

5.4.1 Implementation Details

Backbone Model and Training Details: In this work, we use 3D-ResNets (R3D-50) as the backbone model for experiments and comparison to other state-of-the-art self-supervised models for comparison of backbone models. The design of our 3D-ResNets follows the 'slow' pathway of the SlowFast [51] network with details revealed in PySlowFast code base [47] (the license can be found in the code based [47]). The training clips have $T = 8$ frames sampled with stride $\tau = 8$ from 64 raw-frames of video. As demonstrated in [52], momentum encoders significantly help for spatio-temporal representation learning and more positive clips are beneficial. We use MoCo [66] (and also BYOL [59] for ablation study) with multiple positive pairs ($P = 4$) for the main experiments.

We follow [52] for all related hyper-parameter settings for training contrastive models [59, 66] on Kinetics-400 and evaluation on UCF101 and HMDB51. To train our framework on top of MoCo on Smarthome, we use a queue storing $N = 8192$ negatives with $m_{base} = 0.994$ for momentum encoder and we use a 3-layer projection MLP. The temperature $Temp$ is set as 0.1. We adopt a half-period cosine schedule [52] of learning rate decaying, with base learning rate 0.1 and the maximum training iterations is 200. We use 8 Tesla V100 GPUs for training LTN with batch size 64 on Smarthome (for $\sim$20 hours) and UCF101 (for $\sim$15 hours) and batch size 128 on Kinetics-400 (for $\sim$120 hours).

Time Encoder and 'Dynamic Component': Unless otherwise stated, we use a 2-layer MLP with hidden dimension 2048, ReLU activation and a Batch Normalization layer at the beginning for the time encoder. For Variant 2, we additionally place a Softmax layer before the output. For the 'Dynamic component' $\mathbf{D}_t$ designed in Variant 2 and Variant 3, unless otherwise stated, we set $M = 64$ orthogonal directions over the $dim = 2048$ dimensions in the latent space for all experiments. See Sec. 5.4.2 for more ablations.

Transformation	Top-1 (%)	Mean (%)
Base: w/o transformation	65.1	49.7
Variant 1: Linear w/o $\mathbf{D}_t$	66.0	49.8
Variant 2: Attention	66.7	51.6
Variant 3: Linear w/ $\mathbf{D}_t$		
w/o orthogonalization of $\mathbf{D}_t$	67.3	53.1
w/ orthogonalization of $\mathbf{D}_t$	**67.8**	**53.7**

Table 5.1 Top-1 accuracy and Mean accuracy on Smarthome CS in comparing proposed Time parameterization variants.

Method	P	Top-1 (%)	Mean (%)
MoCo [66]	2	61.5	47.2
ρMoCo [52]	4	65.1	49.7
ρBYOL [52]	4	61.7	42.4
LTN + MoCo	2	65.5	49.0
LTN + ρMoCo	4	**67.8**	**53.7**
LTN + ρBYOL	4	63.3	45.1

Table 5.2 Top-1 accuracy and Mean per-class accuracy on Smarthome CS signifying the impact of LTN on *different contrastive frameworks*. P: number of positive pairs.

Evaluation Protocols: For evaluating the learned representation, on Smarthome (for main ablation study), we first pre-train LTN using the training data while without the action labels. Then we conduct a *linear evaluation* by retraining only the fully-connected classifier with the backbone frozen for both Cross-subject (CS) and Cross-view2 (CV2) protocols without additional training data. On UCF101 and Kinetics-400, we also provide the linear results with self-supervised pre-training without extra data. To fully compare other state-of-the-art self-supervised models, we also report the transfer learning results in the setting with pre-training on the large-scale Kinetics-400 dataset followed by a *linear evaluation*, as well as a *fine-tuning evaluation i.e.*, further refining the whole network on Smarthome, UCF101 and HMDB51. We compare Mean per-class accuracy on Smarthome while Top-1 classification accuracy on other benchmarks following their respective evaluation protocols.

5.4.2 Ablation Study

As activities of Toyota Smarthome (Smarthome) have similar motion and high duration variance (*e.g.*, 'Leave', 'Enter', 'Clean dishes', 'Clean up'), the temporal information is generally crucial for action classification. To understand the contribution of LTN for video representation learning, we conduct ablation experiments on Smarthome Cross-Subject [36],

#Layers	#Dimensions	Top-1 (%)	Mean (%)
None	-	65.1	49.7
1	128	66.7	50.5
1	1024	67.3	52.3
2	1024	67.1	52.8
2	2048	**67.8**	**53.7**
3	2048	67.9	53.2

Table 5.3 Top-1 accuracy and Mean per-class accuracy on Smarthome CS *w.r.t.Time Encoder*.

Size of $\mathbf{D}_t$ (M)	Top-1 (%)	Mean (%)
$M = 16$	65.2	51.6
$M = 64$	**67.8**	**53.7**
$M = 128$	67.3	52.2
$M = 512$	67.6	52.1
$M = 1024$	67.5	51.1
$M = 2000$	66.9	50.5

Table 5.4 Top-1 and Mean accuracy on Smarthome CS for study on number of directions in the orthogonal basis $\mathbf{D}_t$.

with *linear evaluation* protocol (*i.e.*, pre-training without action labels, then training the classifiers only with the action labels) using RGB videos without additional modalities or training data. For the proposed $\mathbf{D}_t$, unless otherwise stated, we set $M = 64$ directions over the $dim = 2048$ dimensions. We report Top-1 and Mean per-class accuracy.

LTN Variants: The key module of LTN is the Time Parameterization method with three effective variants. To study the impact of each variant, we start from a baseline using MoCo [66] with multiple positive samples $P = 4$ as [52] and we then incorporate the time parameterization variants. The results in Tab. 5.1 indicate that leveraging time information is pertinent in improving the accuracy of fine-grained action classification. Specifically, in Variant 1, joint linear addition and visual representation related to time encoding without using $\mathbf{D}_t$ slightly boosts the Top-1 performance. We argue that the learned representation should code spatio-temporal data augmentation. If the entire representation is biased by time in the absence of $\mathbf{D}_t$, the static information that should be invariant is also shifted. This motivates us to use latent space decomposition to disentangle the 'time-encoded component' $\mathbf{D}_t$ coded in the learned representation. Using $\mathbf{D}_t$ to parameterize time encoding can significantly improve the performance (+1.9% by Variant 2 based on attention), especially by means of linear transformation (+4.0% by Variant 3).

Method	Supervision	Backbone	Mod.	Dataset	Frozen	**Toyota Smarthome** CS(%)	CV2(%)
From scratch	Supervised	R3D-50	V	SH	×	50.2	28.6
SimCLR [21]	Self-sup.	R3D-50	V	SH	✓	42.2	26.3
SwAV [15]	Self-sup.	R3D-50	V	SH	✓	41.4	25.6
MoCo [66]	Self-sup.	R3D-50	V	SH	✓	47.2	28.8
ρBYOL [52]	Self-sup.	R3D-50	V	SH	✓	42.4	26.8
LTN (Ours)	Self-sup.	R3D-50	V	SH	✓	**53.7**	**30.1**
LTN (Ours)	Self-sup.	R3D-50	V	K400	✓	**54.5**	**35.5**
STA [36]	Supervised	I3D+LSTM	V+P	K400	×	54.2	50.3
AssembleNet++ [140]	Supervised	R(2+1)D-50	V	K400	×	63.6	-
NPL [131]	Supervised	R3D-50	V	K400	×	-	54.6
ImprovedSTA [28]	Supervised	I3D+LSTM	V+P	K400	×	63.7	53.6
VPN [38]	Supervised	I3D+AGCNs	V+P	K400	×	60.8	53.5
MoCo [66]	Self-sup.	R3D-50	V	K400	×	61.8	52.7
LTN (Ours)	Self-sup.	R3D-50	V	K400	×	**65.9**	**54.6**

Table 5.5 Comparison of LTN to state-of-the-art methods on the Toyota Smarthome dataset (SH) with Cross-Subject (CS) and Cross-View2 (CV2) evaluation protocols. Mod: Modalities, V: RGB frames only, P: pre-extracted Pose data (skeleton keypoints coordinates), K400: the Kinetics-400 dataset. We classify methods *w.r.t.*supervision in the second column.

Impact of LTN for Different Contrastive Models: We compare two state-of-the-art momentum-based contrastive models [59, 66], a pair positive samples (P=2) and the improved versions [52] by leveraging multiple positive Keys (P=4) on the Smarthome dataset. Then, we incorporate the proposed LTN (Variant 3 with $M = 64$) into all models. The results in Tab. 5.2 demonstrate that LTN improves all three models and performs the best with ρ MoCo [52] for our target downstream action classification task.

Design of Time Encoder: We explore how many directions are required in $\mathbf{D}_t$. We empirically test six different values for M from 64 to 2000. Quantitative results in Tab. 5.4 show that when using 64 directions (out of all dim=2048 directions), the model achieves the best action classification results. Hence, we set $M = 64$ for the other experiments. For the design of the proposed time encoder, we investigate the effect of different numbers of hidden layers and dimensions for the time encoder across five architectures. The results shown in Tab. 5.3 suggest that 2-layer MLP with 2048 dimensions in the hidden layer is the most effective.

Method	Backbone	Mod.	**K400 (%)**
VTHCL [199]	R3D-50	V	37.8
CVRL [132]	R3D-50	V	66.1
SeCo [207]	R3D-50	V	61.9
MoCo [66]	R3D-50	V	66.6
ρBYOL [52]	R3D-50	V	70.0
MCL [100]	R3D-50	V+F	66.6
LTN (Ours)	R3D-50	V	**71.3**

Table 5.6 Comparison with state-of-the-art methods on Kinetics-400 by *Linear evaluation*. Mod: Modalities, V: RGB frames only, F: pre-extracted optical flow.

5.4.3 Comparison with State-of-the-art

We first compare our method on Smarthome. As we are the firsts to conduct the self-supervised action classification task on this dataset using only RGB data, we re-implement state-of-the-art models [15, 21, 52, 59, 66] and we compare the linear evaluation results without extra training data. We find that our proposed LTN, jointly with MoCo [66] achieves state-of-the-art performance, see Tab. 5.5. To further compare the results with skeleton-based methods [28, 38] trained with additional stream [200, 202], we conduct a self-supervised pre-training on Kinetics-400 and we transfer the model on Smarthome by linear evaluation and fine-tuning, see Tab. 5.5 bottom. In both settings, our model outperforms self-supervised state-of-the-art accuracy and many supervised approaches [28, 36, 38, 131, 140, 145].

We then compare our method to state-of-the-art approaches by linear evaluation on the general video understanding benchmark, Kinetics-400. For fair comparison, we mainly focus on the methods using R3D-50 and $T = 8$ sampled frames for training. The results are shown in Tab. 5.6 and demonstrate that our LTN improves the results upon previous methods [52, 66, 100, 132, 199, 207].

We also compare our LTN to state-of-the-art on HMDB51 and UCF101 (see Tab. 5.7). For fair comparison, we mainly focus on the model trained with the R3D-50 backbone used in our work with training frames $T = 8$. Using frozen features, our model outperforms all other works and even outperforms a number of works that adopt fine-tuning. For fine-tuning, the improvements are slight as the duration of these videos is small and the actions are not as sensitive as Smarthome to time variance. However, we still outperform all previously single RGB-based models and our model performs competitively with current multi-modal methods [62, 100, 135] combining information from pre-extracted optical flow and audio.

Method	Backbone	Mod.	Data	Frozen	**UCF** (%)	**HMDB** (%)	Data	Frozen	**UCF** (%)	**HMDB** (%)
OPN [90]	VGG-M	V	-	✓	-	-	UCF	×	59.6	23.8
ClipOrder [195]	R(2+1)D	V	-	✓	-	-	UCF	×	72.4	30.9
CoCLR [62]	S3D	V	UCF	✓	70.2	39.1	UCF	×	81.4	52.1
LTN (Ours)	R3D-50	V	UCF	✓	**71.8**	**40.3**	UCF	×	**81.6**	**52.8**
SpeedNet [9]	S3D-G	V	-	✓	-	-	K400	×	81.1	48.8
VTHCL [199]	R3D-50	V	-	✓	-	-	K400	×	82.1	49.2
TaCo [8]	R3D-50	V	K400	✓	59.6	26.7	K400	×	85.1	51.6
MoCo [66]	R3D-50	V	-	✓	-	-	K400	×	92.8	67.5
CVRL [132]	R3D-50	V	-	✓	-	-	K400	×	92.2	66.7
ρBYOL [52]	R3D-50	V	-	✓	-	-	K400	×	94.2	72.1
SeCo [207]	R3D-50	V	K400	✓	-	-	K400	×	88.3	55.6
CATE [163]	R3D-50	V	K400	✓	84.3	53.6	K400	×	88.4	61.9
CORP [72]	R3D-50	V	K400	✓	90.2	58.7	K400	×	93.5	68.0
FAME [40]	I3D	V	K400	✓	-	-	K400	×	88.6	61.1
LTN (Ours)	R3D-50	V	K400	✓	**90.6**	**58.9**	K400	×	**94.5**	**72.3**
CoCLR [62]	S3D	V+F	K400	✓	77.8	52.4	K400	×	90.6	62.9
MCL [100]	R(2+1)D-50	V+F	-	✓	-	-	K400	×	93.4	69.1
BraVe [135]	TSM-50x2	V+F+A	AudioS	✓	92.8	70.6	AudioS	×	96.5	79.3

Table 5.7 Comparison with state-of-the-art methods on UCF101 and HMDB51 with pre-training on Kinetics-400 (K400). Mod: Modalities, V: RGB frames only, F: pre-extracted optical flow, A: Audio.

5.4.4 Further Analysis

Per-class Comparison with State-of-the-art: We list the Smarthome classes that benefit the most and the least from LTN (see Tab. 5.8) compared to the state-of-the-art model (MoCo). We find that our method is able to effectively classify the fine-grained actions (*e.g.*, 'Cook.Usestove' +47.1%, 'Makecoffee.Boilwater' +31.8%, 'Laydown' +25.9%, 'Leave' +22.4%) while being challenged in distinguishing some object-oriented activities (*e.g.*, 'Drink.Fromglass' -28.3%, 'Drink.Fromcan': -14.2%). We believe that this is due to the fact that we focus on temporal modeling using time encoding, which only places emphasis on humans and ignores object information. To tackle this challenge and to further improve classification performance, future work will extend our method to latent spatial information [163] in order to capture the object information, while maintaining time awareness, which is still an open problem.

Representation Analysis: To demonstrate that the learned presentations are aware of temporal augmentations, we randomly select 2 videos ('Leave' and 'Enter') that are correctly classified by our model and uniformly sample 20 segments for each video. Then, we visualize their time-aware (learned by the proposed LTN) and time-invariant (learned by MoCo) representations respectively with t-SNE (see Fig. 5.3). We find that, unlike the time-

Activity	Gain from LTN (%)
Cook.Usestove	+47.08
Maketea.Boilwater	+31.78
Laydown	+25.88
Cutbread	+25.42
Leave	+22.43
Mean Accuracy	+6.97
Walk	-5.07
Usetablet	-11.30
Cook.Cleandishes	-12.74
Drink.Fromcan	-14.24
Drink.Fromglass	-28.25

Table 5.8 Activities that benefit the most and the least from LTN, and Mean per-class accuracy gain on Smarthome CS.

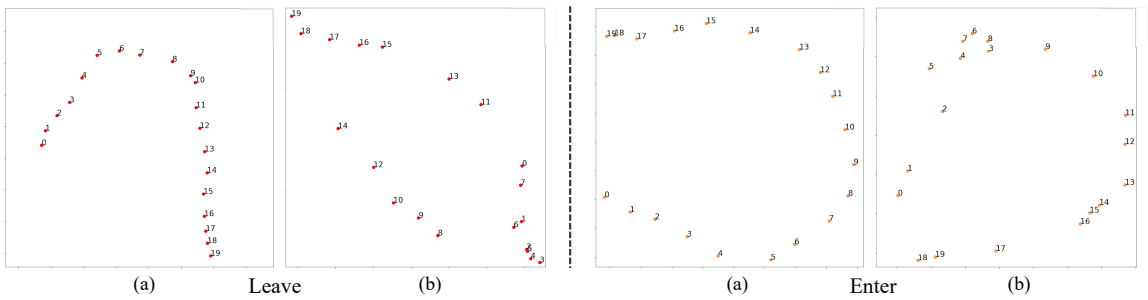

Fig. 5.3 Impact of LTN for video 'Leave' and 'Enter'. (a) Time-aware representations learned by LTN. (b) Their Time-invariant representations learned without LTN modules. The numbers indicate the time order of each uniformly sampled segment.

invariant representations of uniformly sampled segments learned by previous model [66] that are only regrouped together, the time-blended representations learned by our LTN are well aligned over the time order. Hence we conclude that LTN can learn the consistent amount of temporal changes with the video segments on their time-aware representations to benefit fine-grained motion-focused action classification.

5.5 Conclusions

In this chapter, we present LTN, a temporal parameterization approach that learns time-aware action representation. We show that embedding time information of each video segment into the contrastive model by time navigation through a time encoder and an orthogonal basis can

significantly improve the representation capability for videos. Experimental analysis confirms that a visual encoder extracting such representation can boost downstream action recognition. Future work will extend our time parameterization approach to spatial dimension, in order to better capture the object information that may also be crucial for fine-grained action recognition.

Chapter 6

Transferable Action Representation with Multi-Modal Learning

In previous chapters, we explored the human action representation based on single modality. In this chapter, we further improve the effectiveness and transferability of video representations by leveraging all the RGB, human motion and text features.

Transferable visual-language models, such as CLIP, have recently shown significant improvement in performance across various downstream vision tasks. Despite their success, these models mostly focus on visual-level pre-training with natural language supervision, often ignoring the subtle human motion dynamics crucial for complex, human-centric action recognition. Addressing this gap, we introduce T-MOR (Transferable Motion Representation), a novel framework designed to capture and analyze fine-grained human actions leveraging 2D human skeleton data. We employ a joint contrastive learning strategy, utilizing augmentations of skeleton sequences and their corresponding features extracted from a visual-language video foundation model. T-MOR is pre-trained on PoseCap-1M, a newly compiled large-scale human activity dataset featuring skeleton sequences. Remarkably, T-MOR, with such pre-training, can also conduct few-shot and zero-shot transfer. Our extensive transfer learning experiments demonstrate the versatility and effectiveness of T-MOR in many fine-grained, human-centric action recognition tasks (*e.g.*, on Toyota Smarthome, Penn Action, UAV-Human, TSU, Charades datasets) including action classification and segmentation.

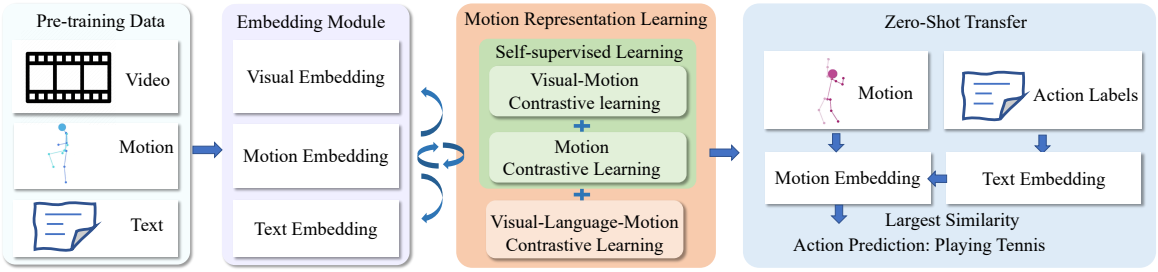

Fig. 6.1 **General pipeline of T-MOR for zero-shot transfer.** The pre-training data stages involve video, motion, and textual inputs. After the embedding module of each modality, the Motion Representation Learning phase includes Motion Contrastive Learning and Visual-Language-Motion Contrastive Learning, aimed at refining the motion model to better understand complex human actions. The final stage, Zero-Shot Transfer, demonstrates the model's capability to predict actions, such as "Playing Tennis," by comparing the largest similarities between motion and text embeddings.

6.1 Introduction

Central to real-world video understanding, *skeleton data*, represented through 2D or 3D human keypoints, plays an important role, as they are complementary to other modalities such as RGB [77, 18, 63, 51, 50, 182, 4]. The skeleton motion modality, robust against variations in camera perspectives and subject appearances, has significantly advanced the domain of activity recognition [43, 197, 145, 157, 202, 22, 99, 44]. However, existing research mainly focuses on training and testing the skeleton model solely on the target dataset. It is still an open question of how to learn a unified transferable skeleton representation that can benefit different downstream tasks especially without any re-training.

As recent intersection of computer vision and natural language processing has witnessed significant advancements, particularly with the emergence of transferable visual-language models like CLIP [133]. These models [106, 196, 148, 149, 198, 12, 114, 150, 178, 55] have revolutionized a myriad of downstream vision tasks, demonstrating unprecedented versatility and performance. Among these tasks, human-centric action recognition stands out as a critical domain, pivotal for applications such as health-care monitoring. Despite the progress, a critical examination reveals a persistent gap: the nuanced understanding of human motion dynamics, especially in complex, fine-grained human actions, remains under-explored. In this context, we hypothesis that a transferable skeleton model can be learned via video-language supervision, and in the transfer stage, only using such trained skeleton model can effectively understand the motion-focused human activities without additional re-training.

Based on this hypothesis, to improve the video representation ability of both skeleton motion models and visual-language models all together, we present a Skeleton Motion Representation model (T-MOR), a pioneering approach designed to bridge the gap of video-language models by focusing on the skeletal representation of human actions. However, as motion features are generally seen as incompatible with video and text modalities, making direct pre-training through feature similarity maximization across these modalities is insufficient [37, 44]. In this context, T-MOR tackles this challenge by employing a dual strategy on top of the three modalities: harnessing 2D skeletal data and integrating a joint contrastive learning mechanism. This method enables the model to capture and learn from the fine-grained subtleties of human motion, a feat that traditional visual-language models often overlook. Specifically, T-MOR includes two stages, a (1) pre-training stage with dual contrastive learning with both skeleton motion features and video-text features, and a (2) transfer learning stage to improve many action understanding tasks on smaller benchmarks. The main strength of T-MOR is that the skeleton model is sufficiently transferable to many downstream tasks without the need for additional modalities in the inference stage. To achieve this, our approach leverages multi-modal contrastive learning, allowing T-MOR to learn from skeleton sequence augmentations and their corresponding features derived from a foundational visual-language video model. This innovative training strategy is complemented by our compilation of PoseCap-1M, a newly collected comprehensive dataset specifically curated to foster the development of advanced human action recognition models. With pre-training on such a dataset, T-MOR can have a strong transfer ability to address different human-centric tasks even with few-shot and zero-shot transfer scenarios, as illustrated in the general pipeline for zero-shot transfer (see Fig. 6.1).

In summary, the contributions of this paper are the following. (i) We introduce T-MOR, a novel transferable skeleton motion model that can be generalized to real-world human-centric action understanding tasks. We propose to enhance the representation ability of recent visual-language video models using human motion data and multi-modal contrastive learning. (ii) We introduce PoseCap-1M, a new large-scale human motion-focused action dataset, featuring high-quality 2D and 3D human skeleton data, for learning generic skeleton motion models. (iii) We conduct a study and show that pre-training T-MOR on PoseCap-1M and transferring it onto an unseen target video dataset represents a generic and effective methodology for action understanding.

6.2 Related Work

Video Representation Learning exploits spatio-temporal pretext tasks from numerous unlabeled data. Towards effective extraction of the pertinent motion information in the time dimension, a number of temporal pretext tasks were proposed, *e.g.*, pixel-level future generation [177], jigsaw-solving [80] and temporal order [195] and a combination of these tasks [8]. These methods are highly constrained by the limited quality of pretext tasks. Recently, video contrastive learning methods [73, 84] have obtained promising results and a large-scale study [52] has been conducted to compare state-of-the-art image-based contrastive methods [15, 21, 59, 66] on videos using spatio-temporal cropping, color jitters, and Gaussian blur data augmentation techniques to generate multiple video views. Further, to improve representation performance, [40, 74, 134, 163] focused on data augmentation techniques, *e.g.*, context-motion decoupling [74], foreground-background merging [40], data augmentation parameterization [163]. Besides contrastive model, Masked visual modeling [65] has been proposed to learn effective visual representations based on the simple pipeline of masking and reconstruction. Based on this, VideoMAE-v2 [181] is shown as a data-efficient learner for self-supervised video pre-training. However, using only RGB video data is limited to understanding the human motions due to the noise caused by background clutter, appearances, etc. In this work, we enhance video contrastive learning by leveraging multiple features extracted from videos to benefit action understand tasks effectively.

Skeleton Motion Representations are learned by extracting spatio-temporal features from skeleton sequences. Compared to using RGB cues, skeleton-based approaches benefit from being less sensitive to variations in appearance, background, and lighting. Techniques such as Graph Convolutional Networks (GCNs) [197, 145, 22] and topology-free models [202, 44] have been applied to model the spatial relationships between joints in the human body, demonstrating improved performance in capturing the essence of human actions.

For self-supervised learning, current methods [97, 168, 112, 203, 110] adopt contrastive learning [192, 66] as the pretext task to learn skeleton representations invariant to data augmentation. Despite these advancements, existing skeleton-based methods have space for improvement, particularly in leveraging the rich contextual information available from video and textual data. Moreover, above methods have only shown promising results on laboratory datasets based on laboratory 3D skeleton sequences [143, 103]. Without sufficiently large pre-training data with large diversity and complexity and without alignment with semantics, they struggle to transfer onto real-world 2D datasets [36, 101, 213] and they are not available for

zero-shot transfer. Deviating from the mentioned methods, T-MOR leverages motion, video, and text features to learn real-world and zero-shot transferable skeleton action representation.

Multi-modal Video Representations enhance video representation ability by combining features extracted from multiple modalities. Current efforts in visual-motion models for videos are centered on the development of advanced computational frameworks that can effectively capture and analyze the intricate relationships between visual and motion data within video sequences [29]. State-of-the-art computer vision techniques with sophisticated motion modeling approaches [37, 30] have applied attention mechanisms [37] or distillations [30] to fuse the features from RGB and skeleton motion. However, they ignore the important semantic information learned from textual data. By adding semantics, this work strives to enhance the capacity of skeleton models to discern and interpret complex visual dynamics, leading to more precise and comprehensive video understanding.

On the other hand, the integration of visual and textual information has led to the development of video-language models. Recently, many methods have used language features [133] for video understanding [106, 196, 148, 149, 114, 178], video captioning [198] and visual question answering [12, 150]. However, these methods are designed to handle short temporal videos, and the challenge of handling actions over a long range of time for solving the task of action detection still persists. These models, especially InternVideo [188], aim to understand and generate descriptions of video content, facilitating a multi-modal understanding of visual data. However, the application of video-language models to action recognition, especially when incorporating skeleton data, remains an under-exploited avenue. MotionCLIP [166] explores to simply align 3D motion features to CLIP [133] features. However, the trained motion representation is still far from satisfactory due to limited training data and learning task. Different from aforementioned work, we have different objective. In our work, we aim at taking advantage of skeleton data and transferring only a light-weight skeleton motion encoder for downstream 2D tasks, to facilitate applications without the need for other modalities. Specifically, we go beyond visual-language representation [187] using large-scale real-world skeleton motion data using dual contrastive learning with both motion augmentation and multi-modal features from video foundation model.

6.3 T-MOR: Transferable Motion Representation

Our proposed Transferable Motion Representation learning framework, T-MOR, has two stages, a first pre-training stage is based on multi-modal contrastive learning (see Fig. 6.2),

Dataset	Real-world	2D	3D	#Videos	#Actions	Fine-grained	Type
Human3.6M [76]	×	✓	✓	209	15	No	Daily living
N-UCLA [180]	×	×	✓	1,475	10	No	Daily living
NTU-RGB+D 60 [143]	×	✓	✓	56,880	60	No	Daily living
NTU-RGB+D 120 [103]	×	✓	✓	114,480	120	No	Daily living
Penn Action [213]	✓	✓	×	2,326	15	No	Sport
UAV-Human [101]	✓	✓	×	21,224	155	No	UAV
Toyota Smarthome [36]	✓	✓	✓	16,115	31	Yes	Daily living
PKU-MMD [27]	×	✓	✓	1,076	51	No	Daily living
Charades [153]	✓	×	×	2,300	151	Yes	Daily living
TSU [34]	✓	✓	✓	536	51	Yes	Daily living
Mixamo [75]	×	✓	✓	2,400	15	No	Synthetics
Kinetics [18]	✓	×	×	400,000	400	No	General video
HT100M [115]	✓	×	×	136M	23K	No	Narrated video
Posetics [202]	✓	✓	✓	142,000	320	No	General activity
PoseCap-1M (Ours)	✓	✓	✓	**1,000,000**	811	Yes	Human-centric action

Table 6.1 A survey of recent datasets for human action classification (top), action segmentation (middle) and transferable action representation learning (bottom) including human skeleton locations.

with skeleton sequences, video (*i.e.*, RGB frame sequences) and texts (*i.e.*, the action names or descriptions generated from action labels). T-MOR is pre-trained on a newly collected large-scale dataset including video, text, and skeleton motion. The second stage is to transfer the pre-trained skeleton motion encoder onto different downstream action recognition tasks. In this section, we first introduce the newly collected dataset for training T-MOR. Secondly, we provide the model and training details for extracting the features encoded from the three types of input data from the proposed dataset. Thirdly, we present the proposed multi-modal contrastive learning details including self-supervised pre-training (with only visual features for pre-training) and supervised pre-training (with also textual features for pre-training). Finally, we show that learned skeleton motion encoder is sufficiently generic to improve many downstream action understanding tasks by transfer learning.

6.3.1 PoseCap-1M: Large-scale Skeleton Data for Training

Recent transferable vision foundation models are generally pre-trained on a huge number of data including images [133] and video clips [188, 106, 196] with their corresponding textual pairs. Similarly, to pre-train a generic and transferable skeleton motion model, such large-scale human-centric multi-modal dataset including videos, text and human skeleton sequences is also needed. However, such dataset is still missing. The current study on human 2D and 3D motion representation learning mostly focuses on laboratory indoor datasets such as NTU-RGB+D [143, 103] using the 3D skeleton data captured from RGBD sensors.

The state-of-the-art (SoTA) skeleton representation learning methods [111, 110, 97] are pre-trained and evaluated on the same or similar scenarios [180]. Since existing indoor laboratory datasets may not contain complex challenges from real-world (*e.g.*, occlusions, compositional actions, large viewpoints variations), it is difficult to use such datasets to pre-train a generic model that can be transferred onto real-world videos. To address this, the recent Posetics dataset [202] was proposed to be a pre-training dataset that can facilitate the study of transfer learning on skeleton-based action recognition. Specifically, it contains 142,000 real-world video clips with the corresponding 2D and 3D body joints. All video clips in Posetics dataset are filtered from Kinetics-400 [18], to contain at least one human pose over 50% of frames. Inspired by this, we scale-up the Posetics dataset to 1 million video clips with corresponding human 2D and 3D skeleton sequences and textual action descriptions by collecting new video clips from more publicly available datasets (*i.e.*, Consented Activities of People (CAP) [11] and Kinetics-700 [17]) and select the activities mainly focusing on human motion. The new extended visual-language-motion dataset is named PoseCap-1M. The comparison of PoseCap-1M with current video datasets including skeleton motion data is shown in Tab. 6.1 and to our knowledge, PoseCap-1M is the largest multi-modal and real-world transferable skeleton motion pre-training dataset.

6.3.2 Multi-modal Feature Extraction

As shown as Fig. 6.2, thanks to the PoseCap-1M dataset, for each video clip, we have its skeletons, video and text (action description) pairs, denoted as $\mathbf{sk}$, $\mathbf{v}$, $\mathbf{a}$ respectively. For the pre-training, in each training iteration, for the skeleton sequence $\mathbf{sk}$, we perform general data augmentations using random temporal cropping and random rotation to generate a positive sample sequence, denoted as $\mathbf{sk}^+$. Subsequently, we adopt a motion encoder $\mathrm{E_M}$ to extract their motion features, $\mathrm{E_M}(\mathbf{sk})$ and $\mathrm{E_M}(\mathbf{sk}^+)$. Different from previous works on skeleton motion representation learning methods [97, 168, 110], which are conducting contrastive learning only on the data augmented skeleton features, we perform multi-modal contrastive learning on top of not only $\mathrm{E_M}(\mathbf{sk})$ and $\mathrm{E_M}(\mathbf{sk}^+)$, but also their corresponding visual features $\mathrm{E_V}(\mathbf{v})$ and textual features $\mathrm{E_T}(\mathbf{a})$, embedded from $\mathbf{v}$ and $\mathbf{a}$ by a visual encoder $\mathrm{E_V}$ and a text encoder $\mathrm{E_T}$ respectively. In this section, we introduce the details of the three encoders to process the skeleton, video and text and to extract their corresponding features. We note that our objective is to train a generic skeleton motion encoder that can be transferred to downstream tasks using only skeleton data. Hence, in the training stage, the $\mathrm{E_M}$ is fully trained from scratch while the backbone encoders $\mathrm{E_V}$ and $\mathrm{E_T}$ are frozen.

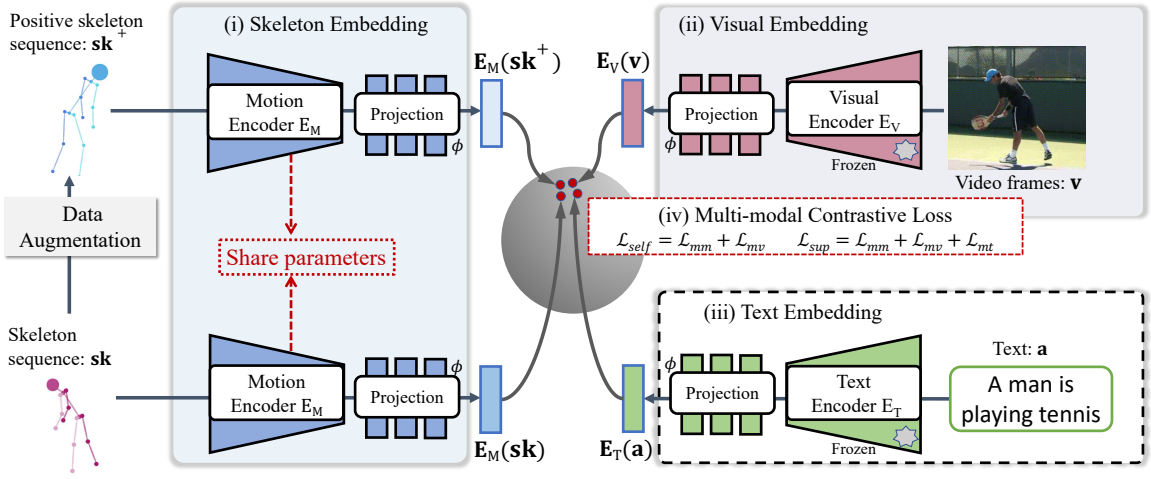

Fig. 6.2 **Overview of Skeleton Motion Representation (T-MOR) Framework.** Given the skeleton sequence **sk**, it begins with data augmentation to get $\mathbf{sk}^+$ to enrich the learning base. The core components include (i) Skeleton Embedding, utilizing a motion encoder $\mathrm{E_M}$ to capture nuanced human movements; (ii) Visual Embedding with a pre-trained encoder $\mathrm{E_V}$ for video frames **v**, enhancing the ability to correlate visual cues with motion data; (iii) Text Embedding with a pre-trained encoder $\mathrm{E_T}$, applying textual description **a** to refine the comprehension of actions; All three embeddings are followed by projection layers ϕ and then are sent to (iv) Multi-modal Contrastive module, a novel mechanism that synergizes skeleton, visual, and text embeddings to optimize the learning process.

Unified Skeleton Modeling and Embedding: We process skeleton sequences as spatio-temporal matrices, represented as $\mathbf{sk} \in \mathbb{R}^{T\times J\times C_{in}}$, where T is the video length, J is the number of body joints per frame, and C_{in} is the input channels (2 for 2D data and 3 for 3D skeletons). To effectively capture skeleton features through time and space, we use the advanced UNIK [202] network as our motion encoder $\mathrm{E_M}$.

For spatial features, we look at body joint movements over time by setting a sliding window across frames. This helps us understand spatial relationships and movements, combining information across multiple frames. At each step, the input **sk** across τ frames in the window becomes a matrix in $\mathbb{R}^{C_{in}\times T\times \tau J}$. For the purpose of spatial modeling, we use a multi-head and residual based processing and formulated as follows:

$$\mathbf{sk_{out}} = \sum_{i=1}^{H} \mathbf{E}_i \cdot \big(\mathbf{sk} \times (\mathbf{W}_i + \mathbf{A}_i)\big), \tag{6.1}$$

where H denotes the number of processing times, namely heads. $\mathbf{E}_i$ represents 2D convolutional weights with 1×1 kernel size, and $\mathbf{W}_i$ is a learnable matrix that captures the

dependencies between spatial joint-level features. $\mathbf{A}_i$ introduces an attention mechanism, adapting $\mathbf{W}_i$ dynamically based on the action being performed, allowing the model to focus on relevant joints for the action. The $\mathbf{W}_i$ is learnable and uniformly initialized as random values.

For the temporal dimension, handling temporal dimension directly with a large dependency weight (*i.e.*, setting the dependency weight to $T \times T$ weights for every pair of frames) would be computationally intensive. Instead, we use 2D convolutional layers with varying dilation rates d and kernel sizes t to learn dependencies over different time scales. described as:the video length is generally large. If we use the same method for spatial dimension , it will consume too much calculation. The temporal processing can be formulated as:

$$\mathbf{sk_{out}} = \mathrm{Conv}_{2D(t\times 1,d)}(\mathbf{sk}). \tag{6.2}$$

After processing spatial and temporal features separately, we merge these insights to form a comprehensive understanding of the skeleton sequence **sk** represented by $\mathrm{E_M}(\mathbf{sk})$. For downstream action prediction tasks, we add a pooling layer (spatial pooling for frame-wise segmentation, or spatio-temporal pooling for video-level classification) and a classifier on top of $\mathrm{E_M}(\mathbf{sk})$ to classify the actions based on the processed skeleton features.

Video-textual Feature Extraction: In this work, we apply ViCLIP [187] to extract video-textual features for all the video clips from the proposed PoseCap-1M. ViCLIP [187] is a general video foundation model. It applies the Vision Transformer (ViT) [42] with spatio-temporal attention as the video encoder and uses a Transformer-based text encoder following [133]. It develops its capabilities through a mix of self-supervised methods, including masked modeling [169] and cross-modal contrastive learning [122] for in-depth feature representation, allowing for efficient learning of transferable video-language representation. As the video and text encoders are well pre-trained on a web-scale video-language dataset [187] including 7 million videos, corresponding to 234 million clips each with the generated captions, we leverage the ViCLIP for initial feature extraction. Different from ViCLIP, we then propose to enhance these features with our novel skeleton motion representations via dual contrastive learning. By freezing the video and textual encoders, we refine the MLP-based projection layers and the full skeleton motion encoder through targeted contrastive learning, optimizing for both self-supervised and supervised pre-training.

6.3.3 Multi-modal Contrastive Learning

With the extracted motion, video, and text (*i.e.*, action annotations) features, the pre-training can be performed in a self-supervised manner via visual-motion contrastive learning without the need for action annotations. This self-supervised pre-training stag can improve the generalizability of the motion encoder and improve downstream tasks with additional fine-tuning. In addition, to realize zero-shot transfer to downstream tasks without any re-training, we propose to conduct a supervised pre-training stage on top of visual-motion-text features for aligning the motion features also with the text features.

Self-supervised Training (Visual-motion Learning): We here provide the details of the self-supervised training of the motion encoder $\mathrm{E_M}$ using only the visual features as supervision. We adopt general contrastive InfoNCE loss [122]. Specifically, we encourage similarities between the features of positive skeleton pairs $\mathrm{E_M}(\mathbf{sk})$ and $\mathrm{E_M}(\mathbf{sk}^+)$, and between $\mathrm{E_M}(\mathbf{sk})$ and its corresponding visual features $\mathrm{E_V}(\mathbf{v})$. Simultaneously, we discourage similarities between the skeleton and visual features and their negative representations encoded from other N samples (we use $N = 65,536$ for experiments) different from the given video clips in the dataset, denoted as $\mathrm{E_M}(\mathbf{sk}_1^-),...,\mathrm{E_M}(\mathbf{sk}_N^-)$ (for skeleton sequences), $\mathrm{E_V}(\mathbf{v}_1^-),...,\mathrm{E_V}(\mathbf{v}_N^-)$ (for the videos), and $\mathrm{E_M}(\mathbf{a}_1^-),...,\mathrm{E_M}(\mathbf{a}_N^-)$ (for the text) respectively. The visual-motion InfoNCE [122] objective is defined as: $\mathscr{L}_{self} = \mathscr{L}_{mm} + \mathscr{L}_{mv}$, where

$$\mathscr{L}_{mm} = -\mathbb{E}\Big(\log \frac{e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_M}(\mathbf{sk}^+)\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_M}(\mathbf{sk}_n^-)\right)}}\Big), \tag{6.3}$$

$$\mathscr{L}_{mv} = -\mathbb{E}\Big(\log \frac{e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_V}(\mathbf{v})\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_V}(\mathbf{v}_n^-)\right)}}\Big). \tag{6.4}$$

The similarity Sim is computed as Eq. 6.5, where *Temp* refers to the temperature hyper-parameter [192], and ϕ is a learnable mapping function (*e.g.*, a MLP projection head [52]) that can substantially improve the learned representations.

$$\mathrm{Sim}(x,y) = \frac{\phi(x)\cdot\phi(y)}{\|\phi(x)\|\cdot\|\phi(y)\|}\cdot\frac{1}{Temp}, \tag{6.5}$$

Supervised Training (Visual-motion-text learning): The supervised pre-training is for zero-shot transfer and is based on visual-motion-text contrastive learning with action an-

notations **a**. The InfoNCE loss function can be formulated as: $\mathscr{L}_{sup} = \mathscr{L}_{mm} + \mathscr{L}_{mv} + \mathscr{L}_{mt}$, where

$$\mathscr{L}_{mt} = -\mathbb{E}\Big(\log \frac{e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_T}(\mathbf{a})\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathrm{E_M}(\mathbf{sk}),\mathrm{E_T}(\mathbf{a}_n^-)\right)}} \Big). \tag{6.6}$$

We note that the negative samples are selected with different action categories as we have action labels. With such supervised pre-training, the motion features and textual features are also aligned in the projected representation space. The so trained T-MOR can be transferred for zero-shot action classification without additional fine-tuning.

6.3.4 Transfer-learning

In this section, we present the second stage, transfer learning of the pre-trained skeleton motion encoder.

Linear Transfer and Fine-tuning: For transferring the motion encoder $\mathrm{E_M}$ on downstream action recognition tasks, we attach $\mathrm{E_M}$ to a spatial-temporal average pooling layer (for classification) or spatial average pooling layer (for frame-wise segmentation) and a fully-connected classifier followed by a Softmax Layer to predict actions. Then, we fully re-train the motion encoder $\mathrm{E_M}$ with skeleton sequences and action labels for fine-tuning, and re-train only the classifier for linear transfer. For processing long sequences, we adopt a sliding window to extract features for a temporal segment and we use Binary Cross Entropy loss to optimize the motion encoder step by step. In this way, $\mathrm{E_M}$ can be re-trained end-to-end instead of pre-extracting features for all frames. In the inference stage, we combine the predictions of all the temporal sliding windows in an online manner [102].

Zero-shot Transfer: We also evaluate the pre-trained T-MOR by zero-shot transfer. Following [133], at test time the the learned skeleton motion encoder $\mathrm{E_M}$ embed skeleton features of the given video, then predict its action by searching the closed text embeddings encoded by text encoder $\mathrm{E_T}$ from names or descriptions of the classes in target datasets. In this work, we simply use action names to obtain text embeddings.

6.4 Experiments and Analysis

We conduct extensive experiments to evaluate T-MOR on both action classification and segmentation tasks. Firstly, we study the generalization ability of T-MOR by quantifying the

Methods	Pre-train	Smarthome			UAV-Human			Penn Action	
		#P.	CS(%)	CV2(%)	#P.	CS1(%)	CS2(%)	#P.	Top-1(%)
Random init. [202]	Scratch	7.97K	24.6	20.7	39.85K	3.8	4.1	3.85K	29.8
Previous SoTA [203]	M w/ labels	-	51.9	52.2	-	32.9	56.1	-	97.3
T-MOR (Ours)	V-M	7.97K	49.3	46.4	39.85K	27.6	43.4	3.85K	86.3
T-MOR (Ours)	V-M-T	7.97K	**52.3**	**53.4**	39.85K	**33.5**	**60.1**	3.85K	**97.8**
Random init. [202]	Scratch	3.45M	63.1	61.2	3.45M	39.2	67.3	3.45M	94.0
Previous SoTA [203]	M w/ labels	-	64.5	65.2	-	42.6	69.5	-	98.0
T-MOR (Ours)	V-M	3.45M	63.2	61.8	3.45M	40.4	67.8	3.45M	96.2
T-MOR (Ours)	V-M-T	3.45M	**66.2**	**66.7**	3.45M	**44.4**	**70.8**	3.45M	**98.2**

Table 6.2 Transfer learning results by **linear evaluation (top)** and **fine-tuning (bottom)** on Smarthome, UAV-Human and Penn Action with pre-training on **PoseCap-1M**. #P.: #Parameters. M/V/T indicates Motion/Visual/Text.

performance improvement obtained by transfer learning on real-world action classification (see Sec. 6.4.2) and action segmentation (see Sec. 6.4.3) datasets after both self-supervised (visual-motion) and supervised (visual-motion-text) pre-training on the large-scale dataset **PoseCap-1M**. Secondly, we evaluate the generalization ability of T-MOR by few-shot (see Sec. 6.4.4) and zero-shot (see Sec. 6.4.5) transfer after supervised pre-training on PoseCap-1M. We note that for transfer learning on downstream tasks, we only use skeleton data and skeleton encoder. Finally, we provide an exhaustive ablation study (see Sec. 6.4.6).

6.4.1 Datasets and Experimental Setting

The experiments are conducted on seven datasets for action understanding including both classification and segmentation tasks.

Toyota Smarthome (Smarthome) [36] contains 16,115 videos across 31 action classes, offering RGB and skeleton data. We utilize 2D skeleton data, following cross-subject (CS) and cross-view2 (CV2) protocols.

UAV-Human [101] features 22,476 UAV-captured sequences, using 2D skeleton data for Cross-subject evaluations (CS1 and CS2).

Penn Action [213] comprises 2,326 sequences of 15 actions, analyzed using 2D skeletons [139] from for standard train-test splits.

Toyota Smarthome Untrimmed (TSU) [34] extends the action classes and video counts, focusing on frame-wise segmentation tasks. We report per-frame mAP following Cross-Subject (CS) and Cross-View (CV) evaluation protocols.

Charades [153] focuses on fine-grained activities. We extract and only use the skeleton data [200] for action segmentation. We report per-frame mAP.

Methods	**Pre-train**	**TSU**			**Charades**	
		#Params	CS(%)	CV(%)	#Params	mAP(%)
Random init. [202]	Scratch	13.1K	8.1	6.9	40.2K	6.1
T-MOR (Ours)	Visual-Motion	13.1K	19.8	12.6	40.2K	11.3
T-MOR (Ours)	Visual-Motion-Text	13.1K	**23.2**	**19.4**	40.2K	**16.6**
Random init. [202]	Scratch	3.45M	28.2	11.0	3.45M	18.6
Previous SoTA	Motion w/ labels	-	26.7 [130]	22.4 [34]	-	9.8 [34]
T-MOR (Ours)	Visual-Motion	3.45M	33.4	21.9	3.45M	18.3
T-MOR (Ours)	Visual-Motion-Text	3.45M	**38.3**	**23.6**	3.45M	**26.0**

Table 6.3 Transfer-learning results by **linear evaluation (top)** and **fine-tuning (bottom)** on real-world datasets Toyota Smarthome Untrimmed (TSU) and Charades with pre-training on **PoseCap-1M.**

6.4.2 Evaluation on Skeleton based Action Classification

In this section, we study the transfer ability of T-MOR by both *linear* (*i.e.*, training only the fully-connected layer while keeping frozen the backbone) and *fine-tuning* (*i.e.*, refining the whole network) evaluations with pre-training on PoseCap-1M. We transfer only the motion encoder E_M onto three 2D skeleton action classification benchmarks (*i.e.*, **Smarthome, UAV-Human** and **Penn Action**) with no additional modalities.

Linear Evaluation: Tab. 6.2 (top) shows the linear results on the three 2D datasets. This experiment evaluates the effectiveness of transfer learning with fewer parameters (only the classifier is trained) compared to classification from random initialization. The results suggest that the weights of the model can be well pre-trained with both visual-motion and visual-motion-text pre-training, providing a strong transfer ability, especially on smaller benchmarks (*e.g.*, +32.7% Smarthome on CV2 and +68.0% on Penn Action compared to solely training from scratch) and the pre-trained skeleton motion encoder is generic enough to extract meaningful action features from skeleton sequences.

Fine-tuning: Tab. 6.2 (bottom) shows the fine-tuning results when the whole network is re-trained. These results suggest that pre-training can improve upon previous SoTA [203] which is supervised pre-trained with only skeleton motion data (*e.g.*, +1.5% on Smarthome CV2). The self-supervised visual-motion pre-trained model also performs competitively compared to supervised pre-trained models. From these results, we conclude that collecting a large-scale video dataset, even without action annotation, and using our proposed T-MOR (including visual-motion pre-training with skeleton data and RGB features from ViCLIP), can still be beneficial to downstream action classification tasks.

Methods	Modality	TSU		Charades
		CS(%)	CV(%)	mAP(%)
TGM [130]	RGB	26.7	-	13.4
SD-TCN [34]	RGB	29.2	18.3	21.6
PDAN [32] w/ I3D [18]	RGB	32.7	-	23.7
MS-TCT [31] w/ I3D [18]	RGB	33.7	-	25.4
PDAN [32] w/ ViCLIP [187]	RGB+Text	21.5	13.4	16.1
MS-TCT [31] w/ ViCLIP [187]	RGB+Text	15.8	-	16.4
Bi-LSTM [58]	Skeleton	17.0	14.8	8.2
TGM [130]	Skeleton	26.7	13.4	9.0
SD-TCN [34]	Skeleton	26.2	22.4	9.8
T-MOR (Ours)	Skeleton	**38.3**	**23.6**	**26.0**

Table 6.4 Frame-level mAP on TSU and Charades for comparison with SoTA action segmentation methods. RGB-based results (top) are shown for reference.

6.4.3 Evaluation on Skeleton based Action Segmentation

We evaluate the transfer ability of T-MOR also by both *linear evaluation* and *fine-tuning evaluation* on two action segmentation datasets **TSU** and **Charades** with pre-training on **PoseCap-1M**.

Linear Evaluation: Tab. 6.3 (top) shows the linear results on the two 2D datasets. The results suggest that the weights of the model can be well pre-trained with full visual-motion-text, providing a strong transfer ability (*e.g.*, +15.1% on TSU CS and +10.5% on Charades) and the pre-trained motion encoder is sufficiently generic to extract meaningful action features from only skeleton sequences in such challenging and complex task. Moreover, with only visual-motion pre-training, the transfer ability is also improved, showing that the visual information is complementary when action annotation is not available.

Fine-tuning: Tab. 6.3 (bottom) shows the fine-tuning results. The visual-motion-text pre-trained model also performs better compared to supervised pre-trained models (*e.g.*, +11.6% on TSU CS and +16.2% on Charades), and the self-supervised visual-motion pre-training performs competitively with previous SoTA [130, 34]. We note that in the transfer learning stage, all the results reported in Tab. 6.2 and Tab. 6.3 are using only skeleton data and the same for the backbone encoder [202].

To further demonstrate the comparison with methods using different modalities (*e.g.*, RGB [18]) and Visual-Text features [187] in the inference stage, we compare our fine-tuning results to other SoTA approaches [130, 34, 32, 31] on the challenging real-world segmentation datasets TSU and Charades (see Tab. 6.4). The results show that T-MOR,

Methods	Pre-train	Label	Smarthome		TSU		Charades
			CS(%)	CV2(%)	CS(%)	CV(%)	mAP(%)
Random init.	Scratch	5%	22.9	33.7	8.5	6.8	8.8
T-MOR	Visual-Motion-Text	5%	**43.7**	**44.6**	**28.0**	**18.8**	**15.8**
Random init.	Scratch	10%	33.8	39.5	12.9	9.5	9.3
T-MOR	Visual-Motion-Text	10%	**50.1**	**51.5**	**30.7**	**20.3**	**19.1**

Table 6.5 Transfer learning results by **fine-tuning** on action classification benchmarks of Toyota Smarthome Trimmed (Smarthome) and segmentation benchmarks of Toyota Smarthome Untrimmed (TSU) and Charades with randomly selected **5% (top)** and **10% (bottom)** of labeled training data after pre-training on **PoseCap-1M**.

Methods	Pre-train	Smarthome		Penn Action
		CS(%)	CV2(%)	Top-1(%)
ViCLIP [187]	InternVid (V-T)	14.1	14.2	74.3
UNIK [202]	Posetics (M only)	12.1	2.7	14.2
T-MOR (Ours)	PoseCap-1M (M-T)	14.5	7.0	69.5
T-MOR (Ours) + ViCLIP	PoseCap-1M (V-M-T)	**21.9**	**17.4**	**80.9**

Table 6.6 Zero-shot transfer results without re-training on action classification benchmarks of Smarthome (Top-1 accuracy) and Penn Action. V/M/T: Visual/Motion/Text.

with Visual-Motion-Text pre-training, outperforms all previous supervised approaches by a large margin. We also implement previous methods [31, 32] using Video-Text features [187]. However, we find that Text features are not always assistive to Visual features for some specific tasks [153]. In contrast, the Motion features learned by T-MOR, with Video-Text supervision, are important to increase the expressive power of the representation and to benefit action segmentation tasks.

6.4.4 Evaluation on Few-shot Transfer

The few-shot transfer ability of T-MOR is shown in Tab. 6.5. Such a scenario is commendable, obtaining high accuracy with limited labeled data. This highlights the model's practicality in real-world applications where data scarcity is prevalent. The results show that our proposed T-MOR, with prior multi-modal learning using three modalities, achieves better performance compared to pretraining from scratch for both action understanding tasks in few-shot setting.

6.4.5 Evaluation on Zero-shot Transfer

The zero-shot transfer capability of T-MOR aims to showcase its ability to generalize to unseen actions, leveraging the knowledge gained from the PoseCap-1M dataset without direct

Methods	Pre-training	Loss	Smarthome CS(%)	TSU CS(%)
Baseline	Scratch	-	24.6	8.1
Motion only	PoseCap-1M	$\mathscr{L}_{mm}$	42.5	12.8
Visual-Motion	PoseCap-1M	$\mathscr{L}_{mv}$	46.3	16.3
Motion & Visual-Motion	Posetics	$\mathscr{L}_{mm}$+$\mathscr{L}_{mv}$	42.6	12.5
Motion & Visual-Motion	PoseCap-1M	$\mathscr{L}_{mm}$+$\mathscr{L}_{mv}$	49.3	19.8
Motion & Visual-Motion-Text	PoseCap-1M	$\mathscr{L}_{mm}$+$\mathscr{L}_{mv}$+$\mathscr{L}_{mt}$	**52.6**	**23.2**

Table 6.7 Ablation study on action classification and segmentation benchmarks of Smarthome and TSU in the linear evaluation setting.

training on specific action labels. Following [133], we employ a strategy where supervisedly trained T-MOR utilizes textual descriptions of actions as proxies for action classes, enabling it to predict actions in videos on which it has not been trained. We evaluate T-MOR on challenging real-world action classification datasets, Smarthome, UAV-Human, and Penn Action, comparing it against previous model [202] that is pre-trained with only skeleton motion data. T-MOR is the first model that is evaluated by zero-shot transfer with only skeletons on the three real-world action classification datasets. From the results in Tab. 6.6, T-MOR outperforms the previous skeleton model [202] by a large margin (*e.g.*, of +% 66.7 on Penn Action) and performs competitively with the current Visual-Text foundation model ViCLIP [187], underscoring its potential for practical applications when training data are not available. Moreover, we show that Motion features are complementary to Visual-Text features, by combining both T-MOR and ViCLIP features to achieve SoTA accuracy.

6.4.6 Further Studies

In this section, we provide a comprehensive ablation study on the different contrastive learning strategies followed by a study for the impact of the new proposed PoseCap-1M dataset. Finally, we discuss the limitations of T-MOR.

Ablation Study of Contrastive Loss: A key component of T-MOR is the utilization of a good strategy for contrastive loss with three modalities, which significantly enhances the model's ability to learn effective motion features without losing the pre-extracted visual-textual features for action recognition. By comparing different contrastive losses, we observe that motion features and visual-textual features are important to each other to be more discriminative to actions. Our analysis in Tab. 6.7 confirms that a balanced dual contrastive

loss with two or three modalities, which emphasizes pulling together similar examples and pushing apart dissimilar ones, achieves the best results.

Impact of Training Data: The diversity and volume of training data play a pivotal role in the generalization capabilities of T-MOR. By systematically varying the dataset size and composition, we evaluate the robustness and adaptability of T-MOR with different datasets. In Tab. 6.7, we find that T-MOR performs well with the previous large dataset [202], Posetics. However, its performance can be further improved by incorporating more varied and complex action sequences, highlighting the benefits of the proposed PoseCap-1M dataset.

Limitation Discussion: Despite T-MOR's impressive achievements showing effective zero-shot and few-shot learning capabilities, we acknowledge certain limitations that merit further exploration. One such limitation is the model's ability to handle fine-grained actions with subtle human-object interactions. Future work could explore more sophisticated models or learning techniques incorporating more modalities (*e.g.*, objects [37] and audio [135]) or context-aware mechanisms. Moreover, the model can be further improved by learning from more data including compositional activities [204] using generative models without the need for real data collection and action annotations.

6.5 Conclusion

This chapter introduces the T-MOR, a novel skeleton-based framework that enhances human action recognition by integrating video-textual features for pre-training. Our approach not only sets new benchmarks in recognizing complex actions with improved accuracy but also demonstrates the model's capability in few-shot and zero-shot learning scenarios using only skeleton data in the inference stage, addressing data scarcity challenges. Future work involves a learning of T-MOR with more modalities for action recognition.

Chapter 7

Perspective and Future Work

In this chapter, we conclude this thesis by providing a summary of contributions and by outlining future research directions, that build on our current action recognition algorithms.

7.1 Scientific Contributions

In this section we summarize the scientific contributions proposed in this thesis.

Generic Skeleton-based Action Recognition Framework: Our goal was to train a foundation skeleton model that can be generalized to different real-world applications *e.g.*, action classification and action segmentation. Firstly, we introduced a novel skeleton refinement method SSTA-PRS to obtain high-quality skeleton data in real-world videos by integrating *multi-expert pose estimators*. Second, we proposed Unified skeleton model UNIK, a novel skeleton-based action recognition method that effectively learns spatio-temporal features on human skeleton sequences and generalizes across datasets. This is achieved by learning an optimal dependency matrix from the *topology-free* distribution based on a *multi-head attention* mechanism. Training a generic model requires a sufficiently large-scale video dataset, which includes high-quality skeleton data. Motivated by this, we thirdly created a novel and larger real-world skeleton dataset, called Posetics, by estimating poses from real-world YouTube videos. We evaluated the proposed UNIK in the context of the Posetics dataset for action classification tasks. Experimental results demonstrate that UNIK, with pre-training on Posetics, outperforms the state-of-the-art when transferred onto multiple target action classification datasets.

Action Representation Learning From Generated Skeletons: Targeting the challenges in composable action segmentation and cross-view/subject action recognition, we proposed two joint generation and representation learnign approach for further improving the generalization ability of UNIK. Firstly, we proposed Latent Action Composition and representation learning framework, namely LAC, a self-supervised framework for *learning from synthesized composable motions* for skeleton-based action segmentation. LAC learns meaningful human primitive motions via an *orthogonal basis (action dictionary)*. Based on LAC, we further proposed a View-invariant Skeleton Action Representation Learning framework (ViA) for cross-view and cross-subject action recognition. Specifically, ViA leverages contrastive learning on top of the generated multi-view skeletons for the same action (using the action dictionary of the generation module of LAC). It facilitates cross-subject and cross-view action classification tasks and demonstrates an improved performance on various datasets.

RGB-based Video Representation Learning: As the most general modality, RGB data, has more information on the human-object interactions, we also aim at pre-training effective RGB-based action representation models. In this context, we proposed Time-aware Video Representation Learning networks. LTN is proposed as a *time-parameterized contrastive learning* strategy for capturing fine-grained motions in video representation learning, showcasing improved performance in action classification tasks.

Multi-modal Action Representation Learning: We believe that our proposed skeleton-based models can be an important complementary modality to benefit the RGB-based models. Hence, we introduced VPN++, a Video-pose Embedding Network [37] (published in IEEE Transactions on Pattern Analysis and Machine Intelligence) using an extension of the pose-driven attention mechanism. VPN++ integrates pose knowledge into RGB through feature-level distillation and mimics pose-driven attention through attention-level distillation, demonstrating superior performance on various datasets. To train the cross-modal action representation model in a self-supervised manner, we proposed Cross-modal Contrastive Learning, a novel visual-motion contrastive learning framework for action recognition.

We also initially explore the video representation learning using multiple modalities. We proposed to incorporate current visual-text pre-training models to improve skeleton motion features for transferable human-centric action recognition. Specifically, we propose T-MOR, a novel skeleton framework that enhances human action recognition by *visual-motion-textual contrastive learning*. Our approach not only sets new benchmarks in recognizing complex actions with improved accuracy but also demonstrates the capability of the model in few-shot

and zero-shot learning scenarios using only skeleton data in the inference stage, addressing data scarcity challenges.

7.2 Challenges and Future Work

In the thesis, we improve the video understanding performance by addressing the limitations caused by subtle motion variance, occlusions, view/subject variance, and action composition. The action classification and segmentation accuracy is significantly improved on fine-grained and motion-oriented scenarios (*e.g.*, indoor daily living activities, sports activities) by the proposed approaches. However, there are still limitations on performance and generalizability for human-object interaction activities, long-term composable activities, ego-centric activities, etc. Future work will focus more on generic multi-modal video understanding modeling to capture more semantic information and to generalize onto the mentioned complex tasks. Additionally, a large enough dataset is needed for learning effective video representations. Designing the synthetic data generation algorithm could facilitate the generic model training. To learn an effective video generation model, addressing the semantic gap between learned representations and human-understandable concepts remains a crucial research area. Striving for interpretable representations that reveal actionable insights from motion data is an ongoing pursuit.

7.2.1 Work in Progress

In response to the current challenges, our research is progressing in various directions, focusing on advancing video understanding tasks. Our ongoing efforts include the following:

Text-to-motion Generation: The text-to-motion generation [127, 128, 211, 5] project is dedicated to exploring the synthesis of realistic and contextually coherent motion sequences from textual descriptions. Compared GAN-based [204] and VAE-based [127] methods, diffusion-based [210, 211] methods are more stable and have higher generation quality. However, there still exist problems. Firstly, diffusion models require a large amount of diffusion steps during inference and it is challenging to generate motion sequences in real-time. Second, the current pipeline only accepts a single form of motion representation. Hence, instead of directly generating motions based on text features using GAN/Diffusion model, we are exploring more controllable but efficient ways. For instance, based on our LAC presented in chapter 4, we are trying to reconstruct motions, guided from a text-conditioned generated

meaningful magnitudes A_m along Motion Dictionary of the Latent Action Decomposition module using Diffusion architectures.

Skeleton Motion Foundation Model: Our work on the motion foundation model aims to establish a comprehensive framework for understanding the fundamental aspects of skeleton motion representation in videos. In chapter 3, we have proposed UNIK and Posetics to learn generic skeleton motion representations. To improve the model generalizability, we are extending Posetics by extracting more skeleton data on videos from real-world [115, 17] (like YouTube video). Moreover, to improve the diversity and the complexity of the actions, we also collect videos with more fine-grained activities (*e.g.*, Yoga [81]) and we generate composable actions using LAC on different collected videos. We will generate more skeleton actions with text descriptions with our text-to-motion generation model in future work. The remaining challenges are the model scaling up methods designs [83] based on our UNIK and the effective pre-text tasks designs [61, 65, 110, 111].

Multi-modal Models for Videos: Our efforts in Visual-motion Models for Videos are centered on the development of advanced computational frameworks that can effectively capture and analyze the intricate relationships between visual and motion data within video sequences [29]. More recently, several methods [106, 196, 148, 149, 198, 12, 114] have used language features [133] for video understanding [106, 196, 148, 149, 114], video captioning [198] and visual question answering [12]. In the domain of Visual-language models for Videos, our focus lies in furthering multi-modal fusion techniques to seamlessly integrate not only visual, textual information but also important skeleton motion information for holistic video comprehension. In complementary of using visual and textual features from videos, we are exploring an effective way of combining also skeleton motion features. Specifically, we adopt the contrastive learning (*e.g.*, T-MOR in chapter 6) on top of motion, semantic and contextual knowledge from all the visual features (*e.g.*, features from LTN in chapter 5), textual features (*e.g.*, action and object description [33]) and skeleton motion features (*e.g.*, features from LAC in chapter 4) to facilitate more robust and accurate video interpretation for multi-object and human-object interactions. Moreover, our research is directed towards enhancing fine-grained video-text alignment and reasoning techniques, enabling our visual-language models to generate coherent and contextually relevant descriptions and narratives for intricate video content.

Fig. 7.1 Ego-centric video understanding (left) and robot learning (right) are still challenging.

Long-term Temporal Dependencies Learning in Untrimmed Videos: Current action recognition methods are mostly designed to handle short temporal videos, and the challenge of handling actions over a long range of time for solving the task of action detection still persists. Therefore, temporal modeling is important for processing long-term sequential data. It is essential to model the temporal dependencies between different time steps in a video [49, 7, 91, 31]. We aim to design an online end-to-end temporal modeling to predict frame-wise actions for long-term videos. However, the global context is missing if we process only the local features from a sliding window. In this context, we propose a global context-aware temporal modeling. Specifically, we store global features after each training iteration on a memory structure to improve the local features during the training stage of the visual encoder.

7.2.2 More Challenges and Future Work

As human motion representation learning continues to evolve, several exciting directions beckon researchers and practitioners (see Fig. 7.1):

Ego-centric Video Understanding: Future research in ego-centric video understanding [35, 142, 57] should prioritize developing advanced models for capturing long-term temporal dependencies and exploring innovative multimodal fusion techniques. Efforts should also focus on devising fine-grained action segmentation methods, optimizing real-time processing, and implementing personalized learning for enhanced user experiences.

Robot Learning: In the field of Robot Learning for Videos [20], future work should emphasize enabling lifelong learning and enhancing robots' understanding of human actions.

Additionally, there is a need to improve spatio-temporal reasoning, investigate transfer learning techniques, and address ethical implications for the responsible integration of video-based learning into robotic systems.

Continual Learning: Exploring approaches for continual learning in the context of human motion representation can enable models to adapt to new actions and scenarios without catastrophic forgetting. This would be particularly valuable in dynamic environments.

Healthcare and Rehabilitation: Applying human motion representation learning in healthcare and rehabilitation settings holds transformative potential. Customizing models to analyze and guide patient movements can assist in designing personalized rehabilitation programs.

References

[1] Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., and Chen, B. (2020a). Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.*

[2] Aberman, K., Weng, Y., Lischinski, D., Cohen-Or, D., and Chen, B. (2020b). Unpaired motion style transfer from video to animation. *ACM Trans. Graph.*

[3] Aberman, K., Wu, R., Lischinski, D., Chen, B., and Cohen-Or, D. (2019). Learning character-agnostic motion for motion retargeting in 2d. *ACM TOG*.

[4] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. *ICCV*.

[5] Athanasiou, N., Petrovich, M., Black, M. J., and Varol, G. (2023). SINC: Spatial composition of 3D human motions for simultaneous action generation. In *ICCV*.

[6] Bachman, P., Hjelm, R. D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. In *NeurIPS*.

[7] Bahrami, E., Francesca, G., and Gall, J. (2023). How much temporal long-term context is needed for action segmentation? In *ICCV*.

[8] Bai, Y., Fan, H., Misra, I., Venkatesh, G., Lu, Y., Zhou, Y., Yu, Q., Chandra, V., and Yuille, A. (2020). Can temporal information help with contrastive self-supervised learning? In *arXiv:2011.13046*.

[9] Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W. T., Rubinstein, M., Irani, M., and Dekel, T. (2020). Speednet: Learning the speediness in videos. In *CVPR*.

[10] Bruno, K., Du, T., and Lorenzo, T. (2019). Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*.

[11] Byrne, J., Castanon, G., Li, Z., and Ettinger, G. (2023). Fine-grained activities of people worldwide. In *WACV*.

[12] Cadene, R., Ben-younes, H., Cord, M., and Thome, N. (2019). Murel: Multimodal relational reasoning for visual question answering. In *CVPR*.

[13] Caetano, C., Brémond, F., and Schwartz, W. (2019). Skeleton image representation for 3D action recognition based on tree structure and reference joints. *SIBGRAPI*.

[14] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE TPAMI*.

[15] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*.

[16] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *ICCV*.

[17] Carreira, J., Noland, E., Hillier, C., and Zisserman, A. (2019). A short note on the kinetics-700 human action dataset. *CoRR*.

[18] Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[19] Chan, C., Ginosar, S., Zhou, T., and Efros, A. A. (2019). Everybody dance now. In *ICCV*.

[20] Chane-Sane, E., Schmid, C., and Laptev, I. (2023). Learning video-conditioned policies for unseen manipulation tasks. In *ICRA*.

[21] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *ICML*.

[22] Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., and Hu, W. (2021a). Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *ICCV*.

[23] Chen, Y., Zhao, L., Yuan, J., Tian, Y., Xia, Z., Geng, S., Han, L., and Metaxas, D. N. (2022). Hierarchically self-supervised transformer for human skeleton representation learning. In *ECCV*.

[24] Chen, Z., Li, S., Yang, B., Li, Q., and Liu, H. (2021b). Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *AAAI*.

[25] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., and Zhang, L. (2020a). Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*.

[26] Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., and Lu, H. (2020b). Skeleton-based action recognition with shift graph convolutional network. In *CVPR*.

[27] Chunhui, L., Yueyu, H., Yanghao, L., Sijie, S., and Jiaying, L. (2017). Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv:1703.07475*.

[28] Climent-Pérez, P. and Florez-Revuelta, F. (2021). Improved action recognition with separable spatio-temporal attention using alternative skeletal and video pre-processing. *Sensors*.

[29] Crasto, N., Weinzaepfel, P., Alahari, K., and Schmid, C. (2019). Mars: Motion-augmented rgb stream for action recognition. In *CVPR*.

[30] Dai, R., Das, S., and Bremond, F. (2021a). Learning an augmented rgb representation with cross-modal knowledge distillation for action detection. In *ICCV*.

[31] Dai, R., Das, S., Kahatapitiya, K., Ryoo, M., and Bremond, F. (2022a). MS-TCT: Multi-Scale Temporal ConvTransformer for Action Detection. In *CVPR*.

[32] Dai, R., Das, S., Minciullo, L., Garattoni, L., Francesca, G., and Bremond, F. (2021b). Pdan: Pyramid dilated attention network for action detection. In *WACV*.

[33] Dai, R., Das, S., Ryoo, M. S., and Bremond, F. (2023). Aan: Attributes-aware network for temporal action detection. In *BMVC*.

[34] Dai, R., Das, S., Sharma, S., Minciullo, L., Garattoni, L., Bremond, F., and Francesca, G. (2022b). Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection. *IEEE TPAMI*.

[35] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*.

[36] Das, S., Dai, R., Koperski, M., Minciullo, L., Garattoni, L., Bremond, F., and Francesca, G. (2019). Toyota smarthome: Real-world activities of daily living. In *ICCV*.

[37] Das, S., Dai, R., Yang, D., and Bremond, F. (2021). Vpn++: Rethinking video-pose embeddings for understanding activities of daily living. *IEEE TPAMI*.

[38] Das, S., Sharma, S., Dai, R., Bremond, F., and Thonnat, M. (2020). Vpn: Learning video-pose embedding for activities of daily living. In *ECCV*.

[39] Diba, A., Sharma, V., Van Gool, L., and Stiefelhagen, R. (2019). Dynamonet: Dynamic action and motion network. In *ICCV*.

[40] Ding, S., Li, M., Yang, T., Qian, R., Xu, H., Chen, Q., Wang, J., and Xiong, H. (2022). Motion-aware contrastive video representation learning via foreground-background merging. In *CVPR*.

[41] Ding, W., Liu, K., Cheng, F., and Zhang, J. (2015). Stfc: Spatio-temporal feature chain for skeleton-based human action recognition. *JVCIR*.

[42] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

[43] Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *CVPR*.

[44] Duan, H., Zhao, Y., Chen, K., Shao, D., Lin, D., and Dai, B. (2022). Revisiting skeleton-based action recognition. In *CVPR*.

[45] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. (2019). Temporal cycle-consistency learning. In *CVPR*.

[46] Fabian Caba Heilbron, Victor Escorcia, B. G. and Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

[47] Fan, H., Li, Y., Xiong, B., Lo, W.-Y., and Feichtenhofer, C. (2020). Pyslowfast. https://github.com/facebookresearch/slowfast.

[48] Fang, H.-S., Xie, S., Tai, Y.-W., and Lu, C. (2017). RMPE: Regional multi-person pose estimation. In *ICCV*.

[49] Farha, Y. A. and Gall, J. (2019). Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*.

[50] Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. In *CVPR*.

[51] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *ICCV*.

[52] Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., and He, K. (2021). A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*.

[53] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *CVPR*.

[54] Gao, X., Hu, W., Tang, J., Liu, J., and Guo, Z. (2019). Optimized skeleton-based action recognition via sparsified graph regression. *ACM MM*.

[55] Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K. V., Joulin, A., and Misra, I. (2023). Imagebind: One embedding space to bind them all. In *CVPR*.

[56] Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. (2017). The "something something" video database for learning and evaluating visual common sense. In *ICCV*.

[57] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., González, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolář, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Ruiz, P., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbeláez, P., Crandall, D., Damen, D., Farinella, G. M., Fuegen, C., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. (2022). Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*.

[58] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *IJCNN*.

[59] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*.

[60] Güler, R. A., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. *CVPR*.

[61] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *CVPR*.

[62] Han, T., Xie, W., and Zisserman, A. (2020). Self-supervised co-training for video representation learning. In *NeurIPS*.

[63] Hara, K., Kataoka, H., and Satoh, Y. (2017). Learning spatio-temporal features with 3D residual networks for actio recognition. In *ICCVW*.

[64] Hara, K., Kataoka, H., and Satoh, Y. (2018). Can spatiotemporal 3D cnns retrace the history of 2D cnns and imagenet? In *CVPR*.

[65] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *CVPR*.

[66] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR*.

[67] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. In *ICCV*.

[68] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*.

[69] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. *CVPR*.

[70] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *CVPR*.

[71] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *ICLR*.

[72] Hu, K., Shao, J., Liu, Y., Raj, B., Savvides, M., and Shen, Z. (2021). Contrast and order representations for video self-supervised learning. In *ICCV*.

[73] Huang, J., Dong, Q., Gong, S., and Zhu, X. (2019). Unsupervised deep learning by neighbourhood discovery. In *ICML*.

[74] Huang, L., Liu, Y., Wang, B., Pan, P., Xu, Y., and Jin, R. (2021). Self-supervised video representation learning by context and motion decoupling. In *CVPR*.

[75] Inc., A. S. (2018). Mixamo. https://www.mixamo.com. https://www.mixamo.com. *Accessed: 2018-12-27*.

[76] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE TPAMI*.

[77] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE TPAMI*.

[78] Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., and Zhu, Y. (2020). Sub-graph contrast for scalable self-supervised graph representation learning. In *ICDM*.

[79] Karen, S. and Andrew, Z. (2014). Two-stream convolutional networks for action recognition in videos. In *NeurIPS*.

[80] Kim, D., Cho, D., and Kweon, I. S. (2019). Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*.

[81] Kim, S. (2023). 3dyoga90: A hierarchical video dataset for yoga pose understanding. *arXiv:2310.10131*.

[82] Kim, T. S. and Reiter, A. (2017). Interpretable 3D human action analysis with temporal convolutional networks. In *CVPRW*.

[83] Kim, Y. J., Awan, A. A., Muzio, A., Cruz-Salinas, A. F., Lu, L., Hendy, A., Rajbhandari, S., He, Y., and Awadalla, H. H. (2021). Scalable and efficient moe training for multitask multilingual models. *arXiv:2109.10465*.

[84] Kong, Q., Wei, W., Deng, Z., Yoshinaga, T., and Murakami, T. (2020). Cycle-contrast for self-supervised video representation learning. In *NeurIPS*.

[85] Kreiss, S., Bertoni, L., and Alahi, A. (2019). Pifpaf: Composite fields for human pose estimation. In *CVPR*.

[86] Kuehne, H., Arslan, A., and Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*.

[87] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: A large video database for human motion recognition. In *ICCV*.

[88] Kundu, J. N., Gor, M., Uppala, P. K., and Radhakrishnan, V. B. (2019). Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In *WACV*.

[89] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *CVPR*.

[90] Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2017). Unsupervised representation learning by sorting sequences. In *ICCV*.

[91] Lezama, J., Alahari, K., Sivic, J., and Laptev, I. (2011). Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*.

[92] Li, B., Chen, H., Chen, Y., Dai, Y., and He, M. (2017a). Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network. In *ICMEW*.

[93] Li, C., Hou, Y., Wang, P., and Li, W. (2017b). Joint distance maps based action recognition with convolutional neural networks. In *ICMEW*.

[94] Li, C., Zhong, Q., Xie, D., and Pu, S. (2018a). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*.

[95] Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. (2018b). Unsupervised learning of view-invariant action representations. In *NeurIPS*.

[96] Li, K., Li, X., Wang, Y., Wang, J., and Qiao, Y. (2021a). Ct-net: Channel tensorization network for video classification. In *ICLR*.

[97] Li, L., Wang, M., Ni, B., Wang, H., Yang, J., and Zhang, W. (2021b). 3d human action representation learning via cross-view consistency pursuit. In *CVPR*.

[98] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*.

[99] Li, M., Chen, S., Liu, Z., Zhang, Z., Xie, L., Tian, Q., and Zhang, Y. (2021c). Skeleton graph scattering networks for 3d skeleton-based human motion prediction. In *ICCVW*.

[100] Li, R., Zhang, Y., Qiu, Z., Yao, T., Liu, D., and Mei, T. (2021d). Motion-focused contrastive learning of video representations. In *ICCV*.

[101] Li, T., Liu, J., Zhang, W., Ni, Y., Wang, W., and Li, Z. (2021e). Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *CVPR*.

[102] Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., and Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. In *ECCV*.

[103] Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L. Y., and Kot, A. C. (2020). Ntu rgb+d 120: A large-scale benchmark for 3D human activity understanding. *IEEE TPAMI*.

[104] Liu, W., Piao, Z., Min, J., Luo, W., Ma, L., and Gao, S. (2019). Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*.

[105] Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020). Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*.

[106] Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., and Li, T. (2021). Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*.

[107] Luo, Z., Hsieh, J.-T., Jiang, L., Niebles, J. C., and Fei-Fei, L. (2018). Graph distillation for action detection with privileged modalities. In *ECCV*.

[108] Luvizon, D., Picard, D., and Tabia, H. (2020). Multi-task deep learning for real-time 3D human pose estimation and action recognition. *IEEE TPAMI*.

[109] Mahasseni, B. and Todorovic, S. (2016). Regularizing long short term memory with 3D human-skeleton sequences for action recognition. *CVPR*.

[110] Mao, Y., Deng, J., Zhou, W., Fang, Y., Ouyang, W., and Li, H. (2023a). Masked motion predictors are strong 3d action representation learners. In *ICCV*.

[111] Mao, Y., Deng, J., Zhou, W., Lu, Z., Ouyang, W., and Li, H. (2023b). I^2md: 3d action representation learning with inter- and intra-modal mutual distillation. *arXiv:2310.15568*.

[112] Mao, Y., Zhou, W., Lu, Z., Deng, J., and Li, H. (2022). Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In *ECCV*.

[113] Mathieu, M., Couprie, C., and LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. In *ICLR*.

[114] Mezghani, L., Bojanowski, P., Alahari, K., and Sukhbaatar, S. (2023). Think before you act: Unified policy for interleaving language reasoning with actions. *arXiv:2304.11063*.

[115] Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019). HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

[116] Misra, I. and van der Maaten, L. (2020). Self-supervised learning of pretext-invariant representations. In *CVPR*.

[117] Misra, I., Zitnick, C. L., and Hebert, M. (2016). Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*.

[118] Moon, G., Chang, J., and Lee, K. M. (2019). Camera distance-aware top-down approach for 3D multi-person pose estimation from a single rgb image. In *ICCV*.

[119] Nie, Q. and Liu, Y. (2021). View transfer on human skeleton pose: Automatically disentangle the view-variant and view-invariant information for pose representation learning. *IJCV*.

[120] Nie, Q., Liu, Z., and Liu, Y. (2020). Unsupervised human 3D pose representation with viewpoint and pose disentanglement. In *ECCV*.

[121] Ning, G., Zhang, Z., and He, Z. (2017). Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE TMM*.

[122] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. In *arXiv:1807.03748*.

[123] Park, J., Lee, J., Kim, I.-J., and Sohn, K. (2022). Probabilistic representations for video contrastive learning. In *CVPR*.

[124] Patrick, M., Asano, Y. M., Huang, B., Misra, I., Metze, F., Henriques, J., and Vedaldi, A. (2021). Space-time crop & attend: Improving cross-modal video representation learning. In *ICCV*.

[125] Pavllo, D., Feichtenhofer, C., Grangier, D., and Auli, M. (2019). 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*.

[126] Peng, W., Hong, X., Chen, H., and Zhao, G. (2020). Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *AAAI*.

[127] Petrovich, M., Black, M. J., and Varol, G. (2022). TEMOS: Generating diverse human motions from textual descriptions. In *ECCV*.

[128] Petrovich, M., Black, M. J., and Varol, G. (2023). TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *ICCV*.

[129] Piergiovanni, A. and Ryoo, M. S. (2018). Learning latent super-events to detect multiple activities in videos. In *CVPR*.

[130] Piergiovanni, A. and Ryoo, M. S. (2019). Temporal gaussian mixture layer for videos. In *ICML*.

[131] Piergiovanni, A. and Ryoo, M. S. (2021). Recognizing actions in videos from unseen viewpoints. In *CVPR*.

[132] Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., and Cui, Y. (2021). Spatiotemporal contrastive video representation learning. In *CVPR*.

[133] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *ICML*.

[134] Ranasinghe, K., Naseer, M., Khan, S., Khan, F. S., and Ryoo, M. (2022). Self-supervised video transformer. In *CVPR*.

[135] Recasens, A., Luc, P., Alayrac, J.-B., Wang, L., Strub, F., Tallec, C., Malinowski, M., Pătrăucean, V., Altché, F., Valko, M., Grill, J.-B., van den Oord, A., and Zisserman, A. (2021). Broaden your views for self-supervised video learning. In *ICCV*.

[136] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *ArXiv*.

[137] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

[138] Rockwell, C. and Fouhey, D. F. (2020). Full-body awareness from partial observations. *ECCV*.

[139] Rogez, G., Weinzaepfel, P., and Schmid, C. (2019). LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE TPAMI*.

[140] Ryoo, M., Piergiovanni, A., Kangaspunta, J., and Angelova, A. (2020). Assemblenet++: Assembling modality representations via attention connections. *ECCV*.

[141] Sardari, F., Ommer, B., and Mirmehdi, M. (2021). Unsupervised view-invariant human posture representation. In *BMVC*.

[142] Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., and Yao, A. (2022). Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *CVPR*.

[143] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). Ntu rgb+d: A large scale dataset for 3D human activity analysis. *CVPR*.

[144] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019a). Skeleton-based action recognition with directed graph neural networks. *CVPR*.

[145] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019b). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*.

[146] Shi, L., Zhang, Y., Cheng, J., and LU, H. (2020). Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. *IEEE TIP*.

[147] Shuang, M., Zhaoyang, Z., Daniel, M., and Yale, S. (2021). Active contrastive learning of audio-visual video representations. In *ICLR*.

[148] Shukor, M., Dancette, C., and Cord, M. (2023a). ep-alm: Efficient perceptual augmentation of language models. In *ICCV*.

[149] Shukor, M., Dancette, C., Rame, A., and Cord, M. (2023b). Unified model for image, video, audio and language tasks. In *ICCVW*.

[150] Shukor, M., Dancette, C., Rame, A., and Cord, M. (2023c). UnIVAL: Unified model for image, video, audio and language tasks. *Transactions on Machine Learning Research*.

[151] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., and Sebe, N. (2019). First order motion model for image animation. *Advances in Neural Information Processing Systems*.

[152] Siarohin, A., Woodford, O. J., Ren, J., Chai, M., and Tulyakov, S. (2021). Motion representations for articulated animation. In *CVPR*.

[153] Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., and Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*.

[154] Skorokhodov, I., Tulyakov, S., and Elhoseiny, M. (2022). Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*.

[155] Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*.

[156] Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2020a). Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACMMM*.

[157] Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2020b). Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM MM*.

[158] Soomro, K., Zamir, A., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv: 1212.0402*.

[159] Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised learning of video representations using lstms. In *ICML*.

[160] Stein, S. and McKenna, S. J. (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*.

[161] Strizhkova, V., Wang, Y., Anghelone, D., Yang, D., Dantcheva, A., and Brémond, F. (2021). Emotion editing in head reenactment videos using latent space manipulation. In *FG*.

[162] Su, Y., Lin, G., and Wu, Q. (2021). Self-supervised 3d skeleton action representation learning with motion consistency and continuity. In *ICCV*.

[163] Sun, C., Nagrani, A., Tian, Y., and Schmid, C. (2021). Composable augmentation encoding for video representation learning. In *ICCV*.

[164] Sun, J. J., Zhao, J., Chen, L.-C., Schroff, F., Adam, H., and Liu, T. (2020). View-invariant probabilistic embedding for human pose. In *ECCV*.

[165] Tanfous, A. B., Drira, H., and Amor, B. B. (2019). Sparse coding of shape trajectories for facial expression and action recognition. *IEEE TPAMI*.

[166] Tevet, G., Gordon, B., Hertz, A., Bermano, A. H., and Cohen-Or, D. (2022). Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*.

[167] Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *ECCV*.

[168] Tianyu, G., Hong, L., Zhan, C., Mengyuan, L., Tao, W., and Runwei, D. (2022). Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *AAAI*.

[169] Tong, Z., Song, Y., Wang, J., and Wang, L. (2022). Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*.

[170] Tran, D., Wang, H., Feiszli, M., and Torresani, L. (2019). Video classification with channel-separated convolutional networks. In *ICCV*.

[171] Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). MoCoGAN: Decomposing motion and content for video generation. In *CVPR*.

[172] Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3D skeletons as points in a lie group. *CVPR*.

[173] Villegas, R., Ceylan, D., Hertzmann, A., Yang, J., and Saito, J. (2021). Contact-aware retargeting of skinned motion. In *ICCV*.

[174] Villegas, R., Yang, J., Ceylan, D., and Lee, H. (2018). Neural kinematic networks for unsupervised motion retargetting. In *CVPR*.

[175] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016a). Anticipating visual representations from unlabeled video. In *CVPR*.

[176] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016b). Generating videos with scene dynamics. In *NeurIPS*.

[177] Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., and Murphy, K. (2018). Tracking emerges by colorizing videos. In *ECCV*.

[178] Wang, A. J., Ge, Y., Yan, R., Yuying, G., Lin, X., Cai, G., Wu, J., Shan, Y., Qie, X., and Shou, M. Z. (2023a). All in one: Exploring unified video-language pre-training. In *CVPR*.

[179] Wang, H., An, W. P., Wang, X., Fang, L., and Yuan, J. (2018). Magnify-net for multi-person 2D pose estimation. *ICME*.

[180] Wang, J., Nie, X., Xia, Y., Wu, Y., and Zhu, S.-C. (2014). Cross-view action modeling, learning and recognition. In *CVPR*.

[181] Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., and Qiao, Y. (2023b). Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*.

[182] Wang, L., Tong, Z., Ji, B., and Wu, G. (2021a). Tdn: Temporal difference networks for efficient action recognition. In *CVPR*.

[183] Wang, T. Y., Ceylan, D., Singh, K. K., and Mitra, N. J. (2021b). Dance in the wild: Monocular human animation with neural dynamic appearance synthesis. In *3DV*.

[184] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020). G3AN: Disentangling appearance and motion for video generation. In *CVPR*.

[185] WANG, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020). ImaGINator: Conditional spatio-temporal gan for video generation. In *WACV*.

[186] Wang, Y., Bremond, F., and Dantcheva, A. (2021c). Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. *arXiv:2101.03049*.

[187] Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Chen, X., Wang, Y., Luo, P., Liu, Z., Wang, Y., Wang, L., and Qiao, Y. (2024). Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*.

[188] Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., Xing, S., Chen, G., Pan, J., Yu, J., Wang, Y., Wang, L., and Qiao, Y. (2022a). Internvideo: General video foundation models via generative and discriminative learning. *arXiv:2212.03191*.

[189] Wang, Y., Yang, D., Bremond, F., and Dantcheva, A. (2022b). Latent image animator: Learning to animate images via latent space navigation. In *ICLR*.

[190] Wei, D., Lim, J., Zisserman, A., and Freeman, W. T. (2018). Learning and using the arrow of time. In *CVPR*.

[191] Weinzaepfel, P. and Rogez, G. (2021). Mimetics: Towards understanding human actions out of context. *IJCV*.

[192] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*.

[193] Xiaolong, W., Allan, J., and Alexei A, E. (2019). Learning correspondence from the cycle-consistency of time. In *CVPR*.

[194] Xie, C., Li, C., Zhang, B., Chen, C., Han, J., Zou, C., and Liu, J. (2018). Memory attention networks for skeleton-based action recognition. *IJCAI*.

[195] Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., and Zhuang, Y. (2019). Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*.

[196] Xu, H., Ghosh, G., Huang, P.-Y., Arora, P., Aminzadeh, M., Feichtenhofer, C., Metze, F., and Zettlemoyer, L. (2021). Vlm: Task-agnostic video-language model pre-training for video understanding. *arXiv preprint arXiv:2105.09996*.

[197] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*.

[198] Yang, A., Nagrani, A., Seo, P. H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., and Schmid, C. (2023a). Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*.

[199] Yang, C., Xu, Y., Dai, B., and Zhou, B. (2020). Video representation learning with visual tempo consistency. In *arXiv:2006.15489*.

[200] Yang, D., Dai, R., Wang, Y., Mallick, R., Minciullo, L., Francesca, G., and Bremond, F. (2021a). Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos. In *WACV*.

[201] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., and Bremond, F. (2021b). Self-supervised video pose representation learning for occlusion-robust action recognition. In *FG*.

[202] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., and Bremond, F. (2021c). Unik: A unified framework for real-world skeleton-based action recognition. In *BMVC*.

[203] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., and Bremond, F. (2024). Via: View-invariant skeleton action representation learning via motion retargeting. *IJCV*.

[204] Yang, D., Wang, Y., Dantcheva, A., Kong, Q., Garattoni, L., Francesca, G., and Bremond, F. (2023b). Lac - latent action composition for skeleton-based action segmentation. In *ICCV*.

[205] Yang, D., Wang, Y., Kong, Q., Dantcheva, A., Garattoni, L., Francesca, G., and Bremond, F. (2023c). Self-supervised video representation learning via latent time navigation. In *AAAI*.

[206] Yang, S., Liu, J., Lu, S., Er, M. H., and Kot, A. C. (2021d). Skeleton cloud colorization for unsupervised 3d action representation learning. In *ICCV*.

[207] Yao, T., Zhang, Y., Qiu, Z., Pan, Y., and Mei, T. (2021). Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*.

[208] Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J.-W., and Shin, J. (2022). Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*.

[209] Zhang, C., Gputa, A., and Zisserman, A. (2021). Temporal query networks for fine-grained video understanding. In *CVPR*.

[210] Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., and Liu, Z. (2022). Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv:2208.15001*.

[211] Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., and Liu, Z. (2023). Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*.

[212] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2019). View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE TPAMI*.

[213] Zhang, W., Zhu, M., and Derpanis, K. G. (2013). From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*.

[214] Zhao, L., Peng, X., Tian, Y., Kapadia, M., and Metaxas, D. N. (2019). Semantic graph convolutional networks for 3d human pose regression. In *CVPR*.

[215] Zhao, L., Wang, Y., Zhao, J., Yuan, L., Sun, J. J., Schroff, F., Adam, H., Peng, X., Metaxas, D., and Liu, T. (2021). Learning view-disentangled human pose representation by contrastive cross-view mutual information maximization. In *CVPR*.

[216] Zheng, W., Li, L., Zhang, Z., Huang, Y., and Wang, L. (2019). Relational network for skeleton-based action recognition. In *ICME*.