Language-Assisted Skeleton Action Understanding for Skeleton-Based Temporal Action Segmentation

Haoyu Ji, Bowen Chen, Xinglong Xu, Weihong Ren, Zhiyong Wang(✉), and Honghai Liu

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China
wangzhiyong@hit.edu.cn

Abstract. Skeleton-based Temporal Action Segmentation (STAS) aims to densely segment and classify human actions in long, untrimmed skeletal motion sequences. Existing STAS methods primarily model spatial dependencies among joints and temporal relationships among frames to generate frame-level one-hot classifications. However, these methods overlook the deep mining of semantic relations among joints as well as actions at a linguistic level, which limits the comprehensiveness of skeleton action understanding. In this work, we propose a Language-assisted Skeleton Action Understanding (LaSA) method that leverages the language modality to assist in learning semantic relationships among joints and actions. Specifically, in terms of joint relationships, the Joint Relationships Establishment (JRE) module establishes correlations among joints in the feature sequence by applying attention between joint texts and differentiates distinct joints by embedding joint texts as positional embeddings. Regarding action relationships, the Action Relationships Supervision (ARS) module enhances the discrimination across action classes through contrastive learning of single-class action-text pairs and models the semantic associations of adjacent actions by contrasting mixed-class clip-text pairs. Performance evaluation on five public datasets demonstrates that LaSA achieves state-of-the-art results. Code is available at https://github.com/HaoyuJi/LaSA.

Keywords: Video Understanding · Skeleton-based Action Segmentation · Language-Assisted Learning · Attention · Contrastive Learning

1 Introduction

Human Activity Recognition (HAR) is a critical field in computer vision, finding applications in healthcare services [5], surveillance systems [6], industrial assembly [37], interactive robotics [39], and virtual reality [23]. Temporal Action

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-72949-2_23.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15112, pp. 400–417, 2025.
https://doi.org/10.1007/978-3-031-72949-2_23

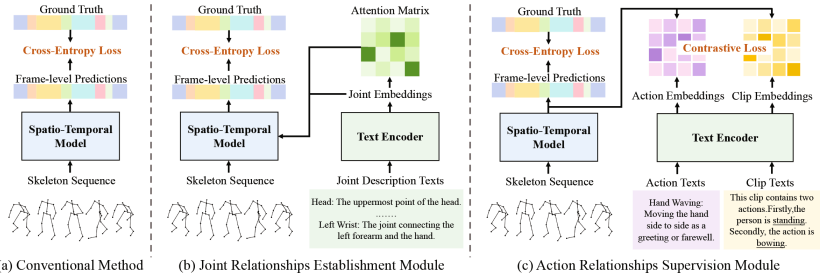


Fig. 1. Comparison of conventional skeleton-based action segmentation architecture and language-assisted skeleton action understanding (LaSA) architecture. The conventional architecture (a) solely employs spatio-temporal modeling of skeleton features to output frame-level representations, supervised with cross-entropy. The Joint Relationships Establishment (JRE) module (b) in LaSA utilizes joint texts to generate joint position embeddings and attention matrices, establishing dependencies among joints. Additionally, the Action Relationships Supervision (ARS) module (c) leverages single-class action-text and mixed-class clip-text contrastive losses to enhance inter-class discrimination and semantic correlations between adjacent actions.

Segmentation (TAS) stands as a challenging advanced task within HAR, aiming to classify each frame of untrimmed temporal action sequences [8]. Presently, the field of TAS is predominantly divided into two categories: Video-based methods (VTAS) and Skeleton-based methods (STAS), contingent upon whether the input features are derived from video RGB or skeleton data [24,25,51].

Skeleton-based temporal action segmentation approaches have recently attracted widespread research attention, owing to their ability to offer more semantically refined representations of motion [24,25] and robustness against background interference, appearance discrepancies, and diverse viewing conditions [24,34,51,57]. Current STAS methodologies predominantly leverage spatio-temporal modeling to establish spatial relationships among joints and temporal relationships across frames, thereby facilitating meaningful frame-level classification representations. Spatially, dependencies among joints are primarily constructed using Graph Convolutional Networks (GCN) [12,15,24,25,54] or Attention mechanisms [29,43]. Temporally, long-range sequential relationships among frames are primarily established through Temporal Convolutional Networks (TCN) [12,15,24,25] or Attention mechanisms [43].

Although existing research has encoded meaningful joint relationships and temporal dependencies among actions for classification, there has been limited exploration into leveraging linguistic priors to enhance representation learning. Specifically, graph or attention-based joint relationships may not fully capture the semantic dependencies among joints [21], while one-hot encoding supervision shown in Fig. 1(a) fails to distinguish the similarity and dissimilarity of actions in the category space [35,49]. Inspired by language prompts in some literature [20,35,46,49,50], we recognize that linguistic definitions encapsulate rich prior knowledge, including establishing connections and distinctions among joints and enhancing fine-grained similarities and differences among actions. For

instance, in the context of joints, the association between "left hand" and "right hand," "left elbow," and "head" varies, which cannot be explicitly provided by prior graphs. Similarly, in the realm of actions, elucidating the similarities (e.g., hand movements close to the head) and differences (variations in motion patterns) between actions such as "drink water" and "brushing teeth" is crucial, an aspect that one-hot encoding fails to capture. Therefore, these linguistic priors can offer fine-grained guidance for representation learning.

This paper introduces a Language-assisted Skeleton Action Understanding (LaSA) network to enhance skeleton-based action segmentation capabilities. Drawing inspiration from current language supervision [20,49,50] and benefiting from advancements [4,7,35] in Large Language Models, LaSA utilizes language modality to foster the understanding of relationships among joints and actions. In the context of Joint Relationships Establishment (JRE), as depicted in Fig. 1(b), attention matrices from joint textual embeddings are utilized to establish correlations, promoting semantic-level feature fusion across joints. Subsequently, during the fusion of spatio-temporal features, joint textual embeddings serve as positional embeddings to underscore semantic distinctions among joints. As for Action Relationships Supervision (ARS), illustrated in Fig. 1(c), contrastive learning is applied between single-class action sequence features and their corresponding action textual embeddings to enhance clustering and discrimination across action classes. Moreover, temporal semantic associations between adjacent actions are reinforced through contrastive learning between clip sequence features encompassing multiple adjacent actions and the sequential textual descriptions of these actions. Importantly, during inference, the ARS module is not utilized, thus incurring no additional computational cost.

To evaluate the effectiveness of LaSA, we evaluated its performance on five public datasets: MCFS-22 [30], MCFS-130 [30], PKU-MMD (X-sub) [27], PKU-MMD (X-view) [27], and LARa [33]. Experimental results demonstrate that the proposed LaSA achieves state-of-the-art (SOTA) performance. The contributions of this work are summarized as follows: (i) LaSA is the first study to apply language-assisted learning to skeleton-based action segmentation, leveraging linguistic priors to enhance the understanding of relationships among joints and actions. (ii) We establish relationships among joints using language knowledge, leveraging attention matrices and position embeddings generated from joint text to delineate correlations and distinctions across joints. (iii) We utilize language supervision to establish relationships among actions, employing contrastive learning with action-text pairs and clip-text pairs to discern different actions and uncover semantic correlations between adjacent actions.

2 Related Works

Temporal Action Segmentation. Video-based action segmentation utilizes RGB [42] or optical flow [38] features extracted from videos. Early methods favored Recurrent Neural Networks (RNN) for temporal modeling [9,40]. Subsequently, TCN and various optimized versions [11,19,22,26] have gained prominence due to their effectiveness in capturing long-term temporal relationships.

Additionally, some TCN-based models enhance effectiveness through exploration of receptive field combinations [13,14], boundary-aware methods [16,48], multi-scale fusion strategies [41], and diffusion models [28]. On the other hand, various forms of Transformers [2,3,10,45,56] have been utilized for action segmentation, enabling adaptive context capturing through attention mechanisms.

Skeleton-based action segmentation utilizes skeleton sequences [32] obtained from motion capture devices [36] or pose estimation algorithms [58]. Current methods typically combine GCN and TCN for spatial and temporal modeling [12,15]. Moreover, some methods further explore the distinction [25] and decoupling [24] of spatial and temporal aspects. Various forms of attention [29,43] are also employed to model spatio-temporal correlations of features. Furthermore, some methods primarily explore processing strategies, such as trajectory primitives and geometric features [52], latent action composition [55], motion interpolation and action synthesis [51]. In comparison to these methods, our approach leverages linguistic priors within joints and actions to augment understanding of skeleton action sequences, providing more granular guidance.

Language Prompt Learning on Action Understanding. Advancements in Natural Language Processing (NLP), particularly LLM like GPT-3 [4] and BERT [7], have significantly influenced the development of multi-modal representation learning in computer vision. Models such as CLIP [35] and ALIGN [17] employ contrastive learning to align textual and visual information effectively, enhancing semantic understanding in downstream tasks. For action recognition, ActionCLIP [46] and GAP [49] utilize CLIP and motion description texts to facilitate learning in video and skeleton-based tasks, respectively. LA-GCN [50] constructs graphs using language texts to aid learning connections between actions and joints. In action segmentation, Bridge-Prompt [20] temporally models clip features contrasted with prompt text, while UnLoc [53] fuses video and category text information for modeling. Inspired by these methods, our LaSA introduces language prompt learning into skeleton-based action segmentation, providing detailed guidance in joint and action aspects through language modality.

3 Method

In skeleton-based action segmentation, skeleton sequences X serve as input, generating categorical label sequences Y. Here, $X \in \mathbb{R}^{C\times T\times V}$, where T, V and C denotes the number of frames, joints and channels, respectively. Meanwhile, $Y \in \mathbb{R}^{Q\times T}$, where Q signifies the number of action classes.

In this section, we introduce the architectural design of the proposed LaSA method, as depicted in Fig. 2. LaSA aims to leverage linguistic knowledge from a pre-trained text model to enhance the learning of relationships among joints and actions, thereby improving action segmentation performance.

3.1 Preliminary

In this section, we present the spatial and temporal modeling methods of the backbone of LaSA, with the specific structure based on DeST [24].

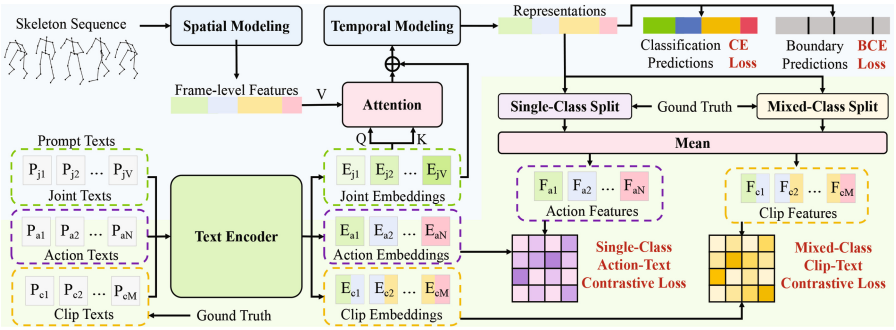


Fig. 2. Overview of the LaSA architecture. LaSA takes skeleton sequences as input, and after initial spatial feature modeling, establishes semantic relationships among joints through joint text embeddings. Upon completion of overall modeling, contrastive supervision is applied to the sequence representation using action and clip text features to enhance the discrimination and correlation among actions. The blue background section represents the architecture during inference, while the green background section denotes additional architecture employed during training.

Multi-scale GCN for Spatial Modeling. For spatial modeling, we adopt a multi-scale GCN inspired by MS-G3D [31] and DeST [24] to capture spatial dependencies among joints. Initially, we define a k-adjacency matrix $A^k \in \{0,1\}^{V\times V}$, connecting joints at a distance of k as follows:

$$A_{ij}^k = \begin{cases} 1, & \text{if } d(\alpha_i, \alpha_j) = k \text{ or } i = j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $d(\alpha_i, \alpha_j)$ represents the shortest distance between joints α_i and α_j. Subsequently, we concatenate all adjacency matrices from 0 to the maximum scale K to form the multi-scale adjacency matrix $A^{MS} \in \{0,1\}^{V\times KV}$:

$$A^{MS} = [(\tilde{D}^1)^{-\frac{1}{2}} A^1 (\tilde{D}^1)^{-\frac{1}{2}}] \oplus \cdots \oplus [(\tilde{D}^K)^{-\frac{1}{2}} A^K (\tilde{D}^K)^{-\frac{1}{2}})] \tag{2}$$

where $\oplus$ represents concatenation along the second dimension. $\tilde{D}^k$ is a diagonal matrix normalizing A^k, with $\tilde{D}_{ii}^k = \sum_j (A_{ij}^k) + \alpha$, where $\alpha = 0.001$. Given the sequence features $F_s \in \mathbb{R}^{C\times T\times V}$, the multi-scale spatial features F_{gcn} obtained after multi-scale GCN can be represented as:

$$F_{gcn} = \text{ReLU}[(A^{MS} + B)F_s W_s] \tag{3}$$

where $B \in \mathbb{R}^{V\times KV}$ is a trainable matrix which can adaptively learn the relationships between joints. $W_s \in \mathbb{R}^{1\times 1\times KC\times C}$ represents the convolution operator for channel adjustment.

Linear Transformer for Temporal Modeling. Unlike TCN, which captures features within a fixed temporal receptive field, the attention mechanism in transformer can adaptively model dependencies among all frames. However,

due to the high dimensionality of the temporal features of action segmentation sequences $F_t \in \mathbb{R}^{C\times T}$, the memory requirements for global attention in conventional transformers are prohibitively high, leading to the necessity of using only local window attention [3,56]. In order to leverage global attention, we adopt the linear former [18,47] used in [24] as our temporal modeling method, which reduces the complexity of attention from $O(n^2)$ to $O(n)$, enabling global temporal modeling. Formally, the linear former layer can be computed as follows:

$$F_{t+1} = \text{ReLU}[\phi(Q_t)(\phi(K_t)^T V_t) \cdot W_t + F_t] \tag{4}$$

Here, Q_t, K_t, and V_t are transformed from F_t through a linear layer $W_{Qt}, W_{Kt}, W_{Vt} \in \mathbb{R}^{C\times C_t}$, where $W_t \in \mathbb{R}^{C_t\times C}$ is a linear layer for channel adjustment, and $\phi(\cdot)$ denotes the sigmoid activation function.

3.2 Language-Assisted Joint Relationships Establishment

Generation of Spatial Text Features. Initially, we utilize GPT4 [1] to generate textual descriptions for each joint P_{jv}, such as: "Left elbow: the bending joint between the upper and lower parts of the left arm." These joint descriptions for V joints are input into the text encoder of CLIP [35], consisting of an embedding layer and 12 transformer layers, yielding joint text embeddings $E_j \in \mathbb{R}^{C_0\times V}$.

Establishment of Spatial Feature Correlations. As depicted in Fig. 3, we utilize the attention matrix generated from joint embeddings E_j to guide the fusion of the multi-scale graph-modeled features F_{gcn}, resulting in features F_{ls}:

$$F_{ls} = \text{Softmax}(Q_j K_j^T) V_j \cdot W_j + F_{gcn} \tag{5}$$

where Q_j and K_j are transformed from E_j through $W_{Qj}, W_{Kj} \in \mathbb{R}^{C_0\times C_j}$, and V_j is transformed from F_{gcn}. W_j is a 1x1 convolutional kernel for channel adjustment. The attention matrix, derived from joint embeddings E_j, orchestrates the formation of spatial correlations within F_{gcn} at a granular linguistic level.

Establishment of Spatial Feature Differences. After establishing spatial correlations, the features $F_{ls} \in \mathbb{R}^{C\times T\times V}$ are obtained. To distinguish features across joints, inspired by positional embeddings in attention mechanisms [44], we convolve and expand joint embeddings $E_j \in \mathbb{R}^{C_0\times V}$ into positional embeddings $E'_j \in \mathbb{R}^{C\times T\times V}$, which are then added to F_{ls}.

Integration of Spatio-Temporal Features. Additionally, as depicted in the latter part of Fig. 3, we adopt methods from [24] for spatio-temporal fusion. Spatial features are adjusted dimensionally via convolution to $F_j \in \mathbb{R}^{L\times T\times V}$, then split along channels into spatial sub-features $F_{jt} \in \mathbb{R}^{V\times T}$, convolved spatially to $F'_{jt} \in \mathbb{R}^{C\times T}$. The first spatial sub-feature undergoes temporal modeling via linear former to F'_t, followed by spatio-temporal attention fusion before the next linear former layer to obtain $F_t \in \mathbb{R}^{C\times T}$:

$$F_t = \text{Softmax}(Q_{jt} K_{jt}^T) V_{jt} \cdot W_{jt} + F'_t \tag{6}$$

where Q_{jt} is derived from F'_{jt}, and K_{jt} and V_{jt} are derived from F'_t.

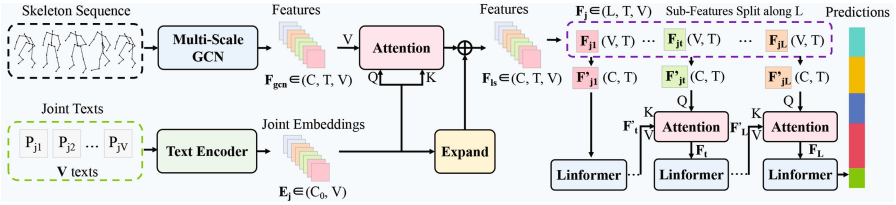


Fig. 3. The implementation of the JRE module within spatial-temporal modeling. After multi-scale graph spatial modeling, JRE utilizes an attention matrix generated from joint text embeddings to assist further integration of spatial features. Then positional text embeddings are incorporated to differentiate distinct joint features, facilitating subsequent attention fusion between spatial sub-features and temporal features.

3.3 Language-Assisted Action Relationship Supervision

The Generation of Action and Clip Text Features. For action text generation, we employ GPT4 [1] to generate descriptive text for each action P_{an}, such as "Clapping: Bringing the hands together repeatedly to express approval or appreciation." Subsequently, we match the corresponding action text for each action segment of ground truth, resulting in N action texts. As for clip text generation, we extract a set of adjacent action labels from one ground truth clip to form a label group LG, for example, $LG_m = [\text{Standing}, \text{Bowing}, \text{Reading}]$. We then use LG to generate textual descriptions for clips P_{cm}, which include the total number of action segments, their sequential order, and their names. For instance, "This clip contains three actions. Firstly, the human is standing. Secondly, the person is bowing. Thirdly, the action is reading." We extract a total of M clip texts with equal intervals and certain overlap. Finally, N action texts and M clip texts are input into the text encoder of CLIP [35], resulting in action text embeddings $E_a \in \mathbb{R}^{C_0 \times N}$ and clip text embeddings $E_c \in \mathbb{R}^{C_0 \times M}$.

Enhancing Inter-action Discrimination. To enhance the discrimination of different actions, we propose a supervised approach leveraging contrastive learning between action segments and their corresponding textual descriptions, as illustrated in Fig. 4. Initially, we segment the representations F_R into N action segments based on the boundaries of each action segment in each ground truth, resulting in features $F'_{a1} \in \mathbb{R}^{C \times T_{a1}}, F'_{a2} \in \mathbb{R}^{C \times T_{a2}}, \cdots F'_{aN} \in \mathbb{R}^{C \times T_{aN}}$. Subsequently, through convolution and mean pooling, we transform each F'_{an} into action features $F_{an} \in \mathbb{R}^{C_0}$, and then aggregate all N action features into $F_a \in \mathbb{R}^{C_0 \times N}$. We then construct action-text pairs between action features F_a and action embeddings E_a, facilitating contrastive learning supervision. This is achieved by computing cosine similarity $sim(\cdot)$ between F_{an} and $E_{an} \in \mathbb{R}^{C_0}$ along the N-dimensional directions, forming a similarity matrix:

$$S_a(F_a, E_a) = \begin{bmatrix} sim(F_{a1}, E_{a1}) & \cdots & sim(F_{a1}, E_{aN}) \\ \vdots & \ddots & \vdots \\ sim(F_{aN}, E_{a1}) & \cdots & sim(F_{aN}, E_{aN}) \end{bmatrix} \tag{7}$$

Fig. 4. The implementation of the ARS module for supervision over representations. ARS primarily employs contrastive learning supervision using single-class action text and mixed-class clip text to supervise the feature representations before prediction. This approach enhances both inter-action discrimination and correlations between adjacent actions. Notably, the ARS module is utilized only during training.

Subsequently, applying softmax functions along the rows/columns of $S_a(F_a, E_a)$, we generate $S_a^T(F_a, E_a)$ and $S_a^V(F_a, E_a) \in \mathbb{R}^{N\times N}$ respectively, based on textual and action similarities. Simultaneously, we define a similarity matrix $S_a^{GT} \in \mathbb{R}^{N\times N}$ based on the ground truth, where positive pairs have similarity scores of 1 and negative pairs have scores of 0. We employ the Kullback-Leibler (KL) divergence, which was applied in [20,49], to maximize the similarity :

$$\mathcal{D}_{KL}(U\|W) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} U_{ij}\log\left(\frac{U_{ij}}{W_{ij}}\right) \tag{8}$$

where U, $W \in \mathbb{R}^{N\times N}$. The contrastive loss for action-text pairs is defined as:

$$\mathcal{L}_{action} = \frac{1}{2}[\mathcal{D}_{KL}(S_a^T\|S_a^{GT}) + \mathcal{D}_{KL}(S_a^V\|S_a^{GT})] \tag{9}$$

Enhancing Adjacent Action Correlations. To improve the semantic correlations between adjacent actions, we employ contrastive learning supervision between clips containing several adjacent action segments and their corresponding clip texts, as illustrated in Fig. 4. Initially, we segment representations to M clips based on the boundaries of M clips in the ground truth boundaries, yielding features $F'_{c1} \in \mathbb{R}^{C\times T_{c1}}, \cdots F'_{cM} \in \mathbb{R}^{C\times T_{cM}}$. Similarly, through convolution and mean pooling, we obtain aggregated features of M clips $F_c \in \mathbb{R}^{C_0\times M}$. Subsequently, akin to the previous method, we construct clip-text pairs between clip features F_c and clip embeddings E_c for contrastive learning supervision, forming a similarity matrix $S_c(F_c, E_c)$ and contrastive loss $\mathcal{L}_{clip}$.

In addition, both contrastive supervision methods mentioned above are exclusively applied during training and omitted during inference. This choice arises from the necessity of ground truth guidance for segmenting actions and clip features during training, rendering them unsuitable for inference scenarios. Furthermore, this supervision method has effectively bolstered the inference capacity of the backbone, allowing for the removal of this structure during inference to reduce memory consumption and enhance inference efficiency.

3.4 LaSA: Overall Framework

The overall framework of LaSA, as depicted in Fig. 5, commences with spatial-temporal modeling $\mathcal{ST}$ of the input $X \in \mathbb{R}^{C\times T\times V}$, yielding feature representa-

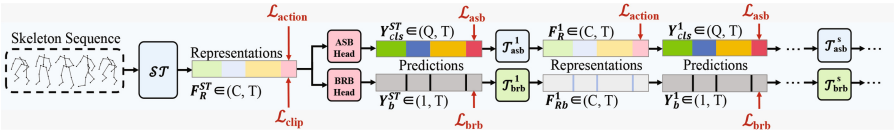


Fig. 5. The overall framework of LaSA. It encompasses a spatial-temporal component, followed by a multi-stage action segmentation branch and a multi-stage boundary regression branch. Multiple loss functions are applied to supervise the network.

tions $F_R^{ST} \in \mathbb{R}^{C \times T}$. Subsequently, F_R^{ST} are fed into two distinct heads to obtain class predictions $Y_{cls}^{ST} \in \mathbb{R}^{Q \times T}$ and boundary predictions $Y_b^{ST} \in \mathbb{R}^{1 \times T}$. Y_c^{ST} is then input into the action segmentation branch $\mathcal{T}_{asb}$ for refined category prediction, while Y_b^{ST} is input into the boundary regression branch $\mathcal{T}_{brb}$ [16] for further boundary prediction. Both branches consist of multiple stages, each comprising multiple layers, with linear former layers in $\mathcal{T}_{asb}$ and TCN layers in $\mathcal{T}_{brb}$.

For contrastive learning between action-text pairs, we apply it to supervise the representation of spatial-temporal modeling $\mathcal{ST}$ and each stage in the action segmentation branch $\mathcal{T}_{asb}$. However, for clip-text pairs, since the $\mathcal{T}_{asb}$ branch is detached from boundary prediction, temporal semantic relations between adjacent actions are not required. Therefore, we only utilize it to supervise the feature representation of $\mathcal{ST}$.

For the supervision of predictions from $\mathcal{ST}$ and $\mathcal{T}_{asb}$, we employ frame-level classification cross-entropy loss $\mathcal{L}_{cls}$ and smoothness loss $\mathcal{L}_{smo}$ [11]. The loss function $\mathcal{L}_{asb}$ is defined as:

$$\mathcal{L}_{asb} = \mathcal{L}_{cls} + \gamma \mathcal{L}_{smo} = -\frac{1}{T}\sum_{t=1}^{T} \log(\hat{y}_{t,\hat{c}}) + \gamma \frac{1}{TC}\sum_{t=1}^{T}\sum_{c=1}^{C} [log(\frac{\hat{y}_{t-1,c}}{\hat{y}_{t,c}})]^2 \quad (10)$$

where $\hat{y}_{t,\hat{c}}$ denotes the predicted probability of ground truth label $\hat{c}$ at time t, and the weight γ for smoothness loss is set to 0.15. For the boundary regression branch $\mathcal{T}_{brb}$, we utilize binary logistic regression loss $\mathcal{L}_{brb}$ [16] in each stage:

$$\mathcal{L}_{brb} = -\frac{1}{T}\sum_{t=1}^{T} (y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t)) \quad (11)$$

where y_t is the ground truth label (1 for the boundary frame and 0 for others), and $\hat{y}_t$ is the predicted boundary probability for the t-th frame. The contrastive losses are $\mathcal{L}_{action}$ and $\mathcal{L}_{clip}$. Thus, the overall loss function is as follows:

$$\mathcal{L} = \sum_{\mathcal{ST}+\mathcal{T}_{asb}} \mathcal{L}_{asb} + \lambda_1 \sum_{\mathcal{ST}+\mathcal{T}_{brb}} \mathcal{L}_{brb} + \lambda_2 \sum_{\mathcal{ST}+\mathcal{T}_{asb}} \mathcal{L}_{action} + \lambda_3 \sum_{\mathcal{ST}} \mathcal{L}_{clip} \quad (12)$$

where λ_1, λ_2, and λ_3 are the loss weights set to 0.1, 0.8, and 0.5 by default.

4 Experiment

4.1 Datasets and Evaluation Metrics

Datasets. We evaluated the LaSA on MCFS-22 [30], MCFS-130 [30], PKU-MMD (X-sub and X-view) [27], and LARa [33] datasets. MCFS [30] consists of motion-centered figure skating data, categorized into 22 and 130 action classes for MCFS-22 and MCFS-130, respectively. PKU-MMD [27] consists of 52 action categories representing human daily behaviors, with distinct X-sub and X-view benchmarks for dataset partitioning. LARa [33] encompasses warehouse activities with 8 action categories. Evaluation was conducted using five-fold cross-validation for MCFS and single validation for PKU-MMD and LARa datasets.

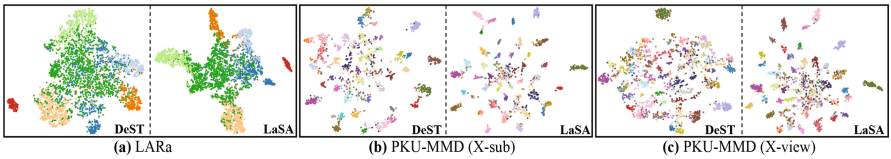
Evaluation Metrics. We employ frame-wise accuracy (Acc), segmental edit score, and segmental F1 score at Intersection over Union (IoU) thresholds of 10%, 25%, 50% (denoted as F1@{10, 25, 50}). Acc provides a direct metric but does not penalize over-segmentation errors. Segmental edit and F1 scores offer a more comprehensive evaluation, effectively penalizing over-segmentation errors.

4.2 Implementation Details

In LaSA, the $\mathcal{ST}$ and $\mathcal{T}_{asb}$ each comprises one stage, while the $\mathcal{T}_{asb}$ comprises two stages, each with 10 temporal modeling layers. The channel dimension for spatial or temporal modeling is $C = 64$, while for the text encoder, $C_0 = 512$. Training is performed on a single RTX 3090 GPU using the Adam optimizer. For MCFS-22 and MCFS-130, we employ a batch size of 1 and a learning rate of 0.0005, training for 300 epochs. For PKU-MMD and LARa, the batch size is 4, with a learning rate of 0.001, training for 300 and 120 epochs, respectively.

4.3 Comparisons with the State-of-the-Art

Quantitative Comparison. Our model is compared with state-of-the-art video-based and skeleton-based action segmentation methods on MCFS-22 [30], MCFS-130 [30], PKU-MMD (X-sub) [27], PKU-MMD (X-view) [27], and LARa [33] datasets, as shown in Tables 1 and 2. Across almost all evaluation metrics on these five datasets, our method achieves state-of-the-art performance, with particularly notable improvements in segment-level metrics, highlighting its



(a) LARa **(b)** PKU-MMD (X-sub) **(c)** PKU-MMD (X-view)

Fig. 6. Visualization of representation space. Each point represents an action segment feature, which is colored based on its class labels. As observed, in comparison to the previous SOTA, LaSA generates a more distinctly structured semantic feature space.

Table 1. Comparison with the state-of-the-art on MCFS-22 and MCFS-130 datasets. **Bold** and underline indicate the best and second-best results in each column.

	Dataset	MCFS-22					MCFS-130				
	Metric	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
VTAS	MS-TCN [11]	75.6	74.2	74.3	69.7	59.5	65.7	54.5	56.4	52.2	42.5
	ETSN [26]	77.0	79.8	81.4	77.6	66.8	64.6	64.6	64.5	61.0	52.3
	ASRF [16]	75.5	77.3	83.3	80.1	69.2	65.6	65.6	66.7	62.3	51.9
	ASFormer [56]	78.7	82.3	82.8	77.9	66.9	67.5	69.1	68.3	64.0	55.1
STAS	MS-GCN [12]	75.5	72.6	75.7	70.5	57.9	64.9	52.6	52.4	48.8	39.1
	SFA+ETSPNet [12]	**81.4**	80.8	82.1	78.3	68.6	-	-	-	-	-
	ID-GCN+ASRF [12]	78.1	81.6	86.4	83.4	73.0	67.1	68.2	68.7	65.6	56.9
	IDT-GCN [12]	79.9	84.5	88.0	84.9	74.9	68.6	70.2	70.7	67.3	58.6
	DeST-TCN [24]	78.7	82.3	86.6	83.5	73.2	70.5	73.8	74.0	70.7	61.8
	DeST-Former [24]	80.4	85.2	87.4	84.5	75.0	71.4	75.8	75.8	72.2	63.0
	LaSA	80.8	**86.7**	**89.3**	**86.2**	**76.3**	**72.6**	**79.3**	**79.3**	**75.8**	**66.6**

Table 2. Comparison with the state-of-the-art on PKU-MMD (X-sub), PKU-MMD (X-view) and LARa datasets. † indicates our implemented results.

Dataset	PKU-MMD (X-sub)					PKU-MMD (X-view)					LARa				
Metric	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
MS-TCN [11]	65.5	–	–	–	46.3	58.2	56.6	58.6	53.6	39.4	65.8	–	–	–	39.6
MS-TCN++† [22]	66.0	66.7	69.6	65.1	51.5	58.4	56.7	58.7	53.2	38.7	71.7	58.6	60.1	58.6	47.0
ETSN† [26]	68.4	67.1	70.4	65.5	52.0	60.7	57.6	62.4	57.9	44.3	71.9	58.4	64.3	60.7	48.1
ASRF† [16]	67.7	67.1	72.1	68.3	56.8	60.4	59.3	62.5	58.0	46.1	71.9	63.0	68.3	65.3	53.2
MS-GCN [12]	68.5	–	–	–	51.6	65.3	58.1	61.3	56.7	44.1	65.6	–	–	–	43.6
CTC [51]	69.2	–	69.9	66.4	53.8	–	–	–	–	–	–	–	–	–	–
DeST-TCN [24]	67.6	66.3	71.7	68.0	55.5	62.4	58.2	63.2	59.2	47.6	72.6	63.7	69.7	66.7	55.8
DeST-Former [24]	70.3	69.3	74.5	71.0	58.7	67.3	64.7	69.3	65.6	52.0	75.1	64.2	70.3	68.0	57.7
LaSA	**73.5**	**73.4**	**78.3**	**74.8**	**63.6**	**69.5**	**67.8**	**72.9**	**69.2**	**57.0**	**75.3**	**65.7**	**71.6**	**69.0**	**57.9**

significant potential in action segmentation. Furthermore, we observe a positive correlation between performance improvement and the number of action classes. Specifically, concerning the F1@50 metric, our model outperforms the previous state-of-the-art method DeST [24] by 0.2% on LARa (8 classes), by 1.3% on MCFS-22 (22 classes), by 4.9% (X-sub) and 5% (X-view) on PKU-MMD (52 classes), and by 3.6% on MCFS-130 (130 classes). This clearly indicates that LaSA can better utilize language priors to enhance discrimination and clustering across different actions, thus exhibiting stronger performance on datasets with a larger number of classes compared to previous methods. Visualization in Fig. 6 of action segment feature spaces further demonstrates the enhancement of action discriminability and clustering afforded by language-assisted learning.

Qualitative Comparison. In Fig. 7, we further present qualitative results. Compared to LaSA, the previous method DeST [24] exhibits certain boundary shift errors (Fig. 7a), and action category prediction errors (Figures 7b, 7c, 7d). On the other hand, MS-GCN [12] shows more errors in action category pre-

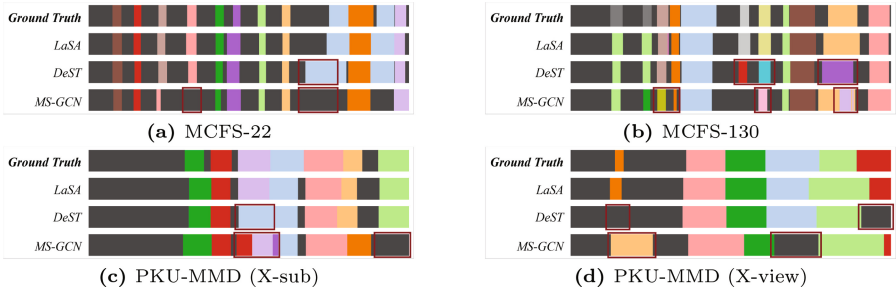


Fig. 7. Qualitative results of action segmentation on the MCFS-130, MCFS-22, PKU-MMD (X-sub), and PKU-MMD (X-view) datasets. Different colors represent distinct action classes. Red boxes highlight segmentation errors in other methods compared to LaSA, underscoring the greater potential of LaSA for action segmentation.

Table 3. Impact of joint relationships establishment module.

JRE Module		MCFS-130					PKU-MMD (X-sub)				
Attention	Embedding	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
✗	✗	72.0	77.6	78.7	74.9	65.9	72.8	71.6	77.5	74.2	62.9
✓	✗	72.2	79.0	79.1	75.5	66.1	73.3	72.8	78.2	74.8	63.5
✗	✓	72.2	78.2	78.3	75.1	65.9	73.2	72.3	77.9	74.4	63.3
✓	✓	**72.6**	**79.3**	**79.3**	**75.8**	**66.6**	**73.5**	**73.4**	**78.3**	**74.8**	**63.6**

diction (Figures 7a, 7b, 7d), and some over-segmentation errors (Figure 7c). In contrast, our method utilizes language priors to establish joint and action relationships, improving inter-action discrimination and semantic relations between adjacent actions. This leads to more accurate predictions of action categories and boundaries, yielding segmentation results closer to ground truth.

4.4 Ablation Studies

Impact of Joint Relationships Establishment Module. To validate the effectiveness of the JRE module, we evaluated LaSA with the inclusion of its language-assisted joint position embeddings and joint attention matrices, as shown in Table 3. The results indicate that performance is lowest when the attention and embeddings of the JRE module are not utilized. The attention mechanism of the JRE module assists in establishing joint correlations, while the position embeddings enhance differences among joints. Therefore, incorporating either attention alone or embeddings alone improves performance. Combining both mechanisms fully establishes joint relationships, resulting in the highest performance and demonstrating the effectiveness of the JRE module.

Impact of Action Relationships Supervision Module. To validate the effectiveness of the ARS module, we evaluated LaSA with the inclusion of its contrastive learning for action-text pairs and clip-text pairs, as shown in Table 4. Performance is lowest when the ARS module is not utilized. Contrastive learning

Table 4. Impact of action relationships supervision module.

ARS Module		MCFS-130					PKU-MMD (X-sub)				
Action-Text	Clip-Text	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
✗	✗	71.4	77.0	77.2	73.6	64.2	71.3	70.9	75.1	71.4	59.4
✓	✗	71.9	78.4	78.7	75.5	65.6	73.4	72.9	78.1	74.5	63.2
✗	✓	72.0	76.8	77.7	74.2	65.0	72.4	72.1	76.4	73.0	60.3
✓	✓	**72.6**	**79.3**	**79.3**	**75.8**	**66.6**	**73.5**	**73.4**	**78.3**	**74.8**	**63.6**

Table 5. Influence of language text prompts. *Detail* denotes description texts, *Name* denotes name texts, and *Simple* signifies "The joint/action of {*Name*}".

Text Prompts Type		MCFS-130					PKU-MMD (X-sub)				
Joint	Action	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
Detail	*Detail*	**72.6**	**79.3**	**79.3**	**75.8**	**66.6**	**73.5**	**73.4**	**78.3**	**74.8**	**63.6**
Simple	*Detail*	71.9	78.1	78.7	75.4	65.9	73.6	72.7	78.0	74.7	63.2
Detail	*Simple*	71.6	78.4	78.8	75.0	66.0	72.5	72.3	77.3	73.9	63.1
Simple	*Simple*	71.4	78.2	78.8	75.2	65.9	73.3	73.1	78.2	74.7	63.0
Name	*Detail*	71.7	79.0	79.0	76.0	66.5	73.5	73.1	78.1	75.0	63.4
Detail	*Name*	72.5	79.1	79.0	75.5	66.2	73.5	73.2	78.3	74.8	63.5
Name	*Name*	72.2	78.3	79.3	75.6	66.0	73.3	73.2	78.3	74.7	63.3

for action-text pairs in the ARS module enhances discrimination and clustering among different actions, while contrastive learning for clip-text pairs establishes correlations between adjacent actions. Therefore, incorporating either mechanism alone improves performance. Combining both mechanisms fully enhances relationships across actions, resulting in the highest performance and demonstrating the effectiveness of the ARS module.

Influence of Language Text Prompts. We explored three types of textual prompts for actions and joints: names of the action/joint *Name*, sentences with names *Simple*: "The action/joint is {*Name*}", and descriptive texts *Detail*, such as "Hopping: jumping on one foot repeatedly." The results shown in Table 5 indicate that when applying *Name* to either or both action and joint texts, the performance is moderate. However, when applying *Simple* to either or both, the performance is poorest, possibly because the sentence "The action/joint is" compresses distances between different action/joint classes in the text space. Performance is highest with *Detail* applied to both, showcasing the effectiveness of detailed descriptions in positioning and distinguishing classes in space. Consequently, we adopted *Detail* as the textual prompt for actions and joints.

Influence of Attention Methods in JRE Module. We evaluated different attention methods in LaSA to assess the impact of textual priors on attention, as shown in Table 6. For the generation of attention matrices, using only features F_j without incorporating textual embeddings E_j yields poor performance. However, introducing joint textual embeddings E_j through various means during atten-

Table 6. Influence of attention methods in JRE module. F_j denotes action features split from representations, while E_j denotes action embeddings from text encoder.

Attention Methods		MCFS-130					PKU-MMD (X-sub)				
Query	Key	Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
F_j	F_j	72.2	78.6	78.8	75.2	66.0	73.4	72.9	78.2	74.1	62.9
$E_j + F_j$	$E_j + F_j$	72.0	78.6	79.0	75.3	66.0	73.3	71.9	77.5	74.1	63.4
E_j	F_j	72.5	78.8	79.2	75.7	66.3	73.6	73.3	78.3	74.7	63.3
E_j	$E_j + F_j$	72.5	78.9	**79.4**	75.7	66.4	**73.9**	**73.5**	78.3	**75.0**	**63.7**
E_j	E_j	**72.6**	**79.3**	79.3	**75.8**	**66.6**	73.5	73.4	**78.3**	74.8	63.6

Table 7. Influence of clip features in ARS module. $\oplus$ indicates comparisons within separate matrices, while + denotes comparisons within the same matrix.

Clip Features		MCFS-130					PKU-MMD (X-sub)				
Split Method		Acc	Edit	F1@{10,25,50}			Acc	Edit	F1@{10,25,50}		
Split Equally	32 clips/video	72.1	78.4	78.3	74.9	65.6	73.5	71.8	76.6	73.9	62.9
	48 clips/video	71.7	77.6	77.8	74.3	64.6	72.9	73.2	77.8	74.5	63.2
Split by Boundaries	2 actions/clip	72.6	**79.3**	79.3	75.8	66.6	**73.5**	**73.4**	78.3	74.8	**63.6**
	3 actions/clip	**72.7**	78.4	**79.5**	**76.0**	**67.0**	73.5	73.2	**78.5**	**75.7**	63.4
	2+3 actions/clip	72.0	78.1	78.3	74.6	66.0	73.4	72.3	77.7	74.4	63.4
	2$\oplus$3 actions/clip	72.4	78.7	79.1	75.4	66.6	73.1	72.6	77.8	74.3	63.2

tion matrix generation leads to varying degrees of performance improvement. Notably, optimal performance was achieved when employing mutual attention between E_j alone or between E_j and F_j+E_j. Thus, we adopted mutual attention between textual embeddings E_j alone to establish inter-joint correlations.

Influence of Clip Features in ARS Module. We evaluated various clip segmentation methods to determine the optimal approach, as shown in Table 7. Two methods were explored: average segmentation and segmentation based on ground truth action boundaries. For average segmentation, we divided one sequence into 32 or 48 clips. For segmentation along action boundaries, we select M clips containing either 2 actions or 3 actions for separate contrastive learning or combine them together in different ways. The results indicate that segmentation along action boundaries outperforms average segmentation, as it captures more complete action information. Among them, contrastive learning with clips containing only 2 or 3 actions achieves near-optimal performance with minimal memory usage. Therefore, we selected segmentation along action boundaries with 2-action clips for contrastive learning as the final approach.

5 Conclusion

In this study, we propose a Language-assisted Skeleton Action Understanding (LaSA) network, which utilizes language modality to assist in establishing con-

nections and distinctions among joints, and enhancing inter-action discrimination and adjacent action correlations. Our model achieves state-of-the-art performance on five challenging datasets. However, it still exhibits some category prediction errors and boundary prediction offsets, leaving room for performance improvement. Future work should incorporate finer-grained motion descriptions for action supervision, providing temporal and spatial subdivision guidance.

Acknowledgements. This work is supported by the National Key Research and Development Program of China under Grant 2022YFB4703200, by the National Natural Science Foundation of China under Grant 62261160652, 52275013, 62206075 and 61733011, by the Guangdong Science and Technology Research Council under Grant 2020B1515120064.

References

1. Achiam, J., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Bahrami, E., Francesca, G., Gall, J.: How much temporal long-term context is needed for action segmentation? In: ICCV, pp. 10351–10361 (2023)
3. Behrmann, N., Golestaneh, S.A., Kolter, Z., Gall, J., Noroozi, M.: Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. ECCV 2022. LNCS, vol. 13695, pp. 52–68. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19833-5_4
4. Brown, T., et al.: Language models are few-shot learners. In: NeurIPS, pp. 1877–1901 (2020)
5. Chen, B., et al.: Autoenp: an auto rating pipeline for expressing needs via pointing protocol. In: ICPR, pp. 3280–3286. IEEE (2022)
6. Dave, I., Scheffer, Z., Kumar, A., Shiraz, S., Rawat, Y.S., Shah, M.: Gabriellav2: towards better generalization in surveillance videos for action detection. In: WACV, pp. 122–132 (2022)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL, vol. 1, pp. 4171–4186 (2019)
8. Ding, G., Sener, F., Yao, A.: Temporal action segmentation: an analysis of modern techniques. IEEE TPAMI **46**(2), 1011–1030 (2024)
9. Ding, L., Xu, C.: Tricornet: a hybrid temporal convolutional and recurrent network for video action segmentation. arXiv preprint arXiv:1705.07818 (2017)
10. Du, D., Su, B., Li, Y., Qi, Z., Si, L., Shan, Y.: Do we really need temporal convolutions in action segmentation? In: ICME, pp. 1014–1019. IEEE (2023)
11. Farha, Y.A., Gall, J.: Ms-tcn: multi-stage temporal convolutional network for action segmentation. In: CVPR, pp. 3575–3584 (2019)
12. Filtjens, B., Vanrumste, B., Slaets, P.: Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks. IEEE Trans. Emerg. Top. Comput. 1–11 (2022)
13. Gao, S.H., Han, Q., Li, Z.Y., Peng, P., Wang, L., Cheng, M.M.: Global2local: efficient structure search for video action segmentation. In: CVPR, pp. 16805–16814 (2021)

14. Gao, S., Li, Z.Y., Han, Q., Cheng, M.M., Wang, L.: RF-Next: efficient receptive field search for convolutional neural networks. IEEE TPAMI **45**(3), 2984–3002 (2023)
15. Ghosh, P., Yao, Y., Davis, L., Divakaran, A.: Stacked spatio-temporal graph convolutional networks for action segmentation. In: WACV, pp. 576–585 (2020)
16. Ishikawa, Y., Kasai, S., Aoki, Y., Kataoka, H.: Alleviating over-segmentation errors by detecting action boundaries. In: WACV, pp. 2322–2331 (2021)
17. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML, pp. 4904–4916. PMLR (2021)
18. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: fast autoregressive transformers with linear attention. In: ICML, pp. 5156–5165. PMLR (2020)
19. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR, pp. 156–165 (2017)
20. Li, M., et al.: Bridge-prompt: towards ordinal action understanding in instructional videos. In: CVPR, pp. 19880–19889 (2022)
21. Li, Q., Wang, Y., Lv, F.: Semantic correlation attention-based multiorder multi-scale feature fusion network for human motion prediction. IEEE Trans. Cybern. **54**(2), 825–838 (2024)
22. Li, S.J., AbuFarha, Y., Liu, Y., Cheng, M.M., Gall, J.: MS-TCN++: multi-stage temporal convolutional network for action segmentation. IEEE TPAMI **45**(6), 6647–6658 (2023)
23. Li, X., et al.: Action recognition based on multimode fusion for VR online platform. Virtual Reality, pp. 1–16 (2023)
24. Li, Y., Li, Z., Gao, S., Wang, Q., Qibin, H., Mingming, C.: A decoupled spatio-temporal framework for skeleton-based action segmentation. arXiv preprint arXiv:2312.05830 (2023)
25. Li, Y.H., Liu, K.Y., Liu, S.L., Feng, L., Qiao, H.: Involving distinguished temporal graph convolutional networks for skeleton-based temporal action segmentation. IEEE TCSVT **34**(1), 647–660 (2024)
26. Li, Y., et al.: Efficient two-step networks for temporal action segmentation. Neurocomputing **454**, 373–381 (2021)
27. Liu, C., Hu, Y., Li, Y., Song, S., Liu, J.: PKU-MMD: a large scale benchmark for skeleton-based human action understanding. In: ACM VSCC, pp. 1–8 (2017)
28. Liu, D., Li, Q., Dinh, A.D., Jiang, T., Shah, M., Xu, C.: Diffusion action segmentation. In: ICCV, pp. 10139–10149 (2023)
29. Liu, K., Li, Y., Xu, Y., Liu, S., Liu, S.: Spatial focus attention for fine-grained skeleton-based action tasks. IEEE Signal Process. Lett. **29**, 1883–1887 (2022)
30. Liu, S., et al.: Temporal segmentation of fine-gained semantic action: a motion-centered figure skating dataset. In: AAAI, pp. 2163–2171 (2021)
31. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: CVPR, pp. 143–152 (2020)
32. Nguyen, H.C., Nguyen, T.H., Scherer, R., Le, V.H.: Deep learning-based for human activity recognition on 3d human skeleton: Survey and comparative study. Sensors **23**(11), 5121 (2023)
33. Niemann, F., et al.: LARa: creating a dataset for human activity recognition in logistics using semantic attributes. Sensors **20**(15), 4083 (2020)
34. Qiu, H., Hou, B., Ren, B., Zhang, X.: Spatio-temporal segments attention for skeleton-based action recognition. Neurocomputing **518**, 30–38 (2023)

35. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763. PMLR (2021)
36. Salisu, S., Ruhaiyem, N.I.R., Eisa, T.A.E., Nasser, M., Saeed, F., Younis, H.A.: Motion capture technologies for ergonomics: a systematic literature review. Diagnostics **13**(15), 2593 (2023)
37. Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A.: Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: CVPR. pp. 21096–21106 (2022)
38. Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., Black, M.J.: On the integration of optical flow and action recognition. In: Brox, T., Bruhn, A., Fritz, M. (eds.) GCPR 2018. LNCS, vol. 11269, pp. 281–297. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12939-2_20
39. Siam, M., et al.: Video object segmentation using teacher-student adaptation in a human robot interaction (hri) setting. In: ICRA, pp. 50–56. IEEE (2019)
40. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: A multi-stream bi-directional recurrent neural network for fine-grained action detection. In: CVPR, pp. 1961–1970 (2016)
41. Singhania, D., Rahaman, R., Yao, A.: C2F-TCN: a framework for semi-and fully-supervised temporal action segmentation. IEEE TPAMI **45**(10), 11484–11501 (2023)
42. Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human action recognition from various data modalities: a review. IEEE TPAMI **45**(3), 3200–3225 (2023)
43. Tian, X., Jin, Y., Zhang, Z., Liu, P., Tang, X.: STGA-Net: spatial-temporal graph attention network for skeleton-based temporal action segmentation. In: ICMEW, pp. 218–223. IEEE (2023)
44. Vaswani, A., et al.: Attention is all you need. NeurIPS **30**, 6000–6010 (2017)
45. Wang, J., Wang, Z., Zhuang, S., Hao, Y., Wang, H.: Cross-enhancement transformer for action segmentation. Multimed. Tools Appl. 1–14 (2023)
46. Wang, M., Xing, J., Liu, Y.: Actionclip: a new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021)
47. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020)
48. Wang, Z., Gao, Z., Wang, L., Li, Z., Wu, G.: Boundary-aware cascade networks for temporal action segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 34–51. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_3
49. Xiang, W., Li, C., Zhou, Y., Wang, B., Zhang, L.: Generative action description prompts for skeleton-based action recognition. In: ICCV, pp. 10276–10285 (2023)
50. Xu, H., Gao, Y., Hui, Z., Li, J., Gao, X.: Language knowledge-assisted representation learning for skeleton-based action recognition. arXiv preprint arXiv:2305.12398 (2023)
51. Xu, L., Wang, Q., Lin, X., Yuan, L.: An efficient framework for few-shot skeleton-based temporal action segmentation. Comput. Vis. Image Underst. **232**, 103707 (2023)
52. Xu, L., Wang, Q., Lin, X., Yuan, L., Ma, X.: Skeleton-based tai chi action segmentation using trajectory primitives and content. Neural Comput. Appl. **35**(13), 9549–9566 (2023)
53. Yan, S., et al.: Unloc: a unified framework for video localization tasks. In: ICCV, pp. 13623–13633 (2023)

54. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (2018)
55. Yang, D., et al.: Lac-latent action composition for skeleton-based action segmentation. In: ICCV, pp. 13679–13690 (2023)
56. Yi, F., Wen, H., Jiang, T.: Asformer: transformer for action segmentation. In: BMVC (2021)
57. Zhang, J., Jia, Y., Xie, W., Tu, Z.: Zoom transformer for skeleton-based group activity recognition. IEEE TCSVT **32**(12), 8646–8659 (2022)
58. Zheng, C., et al.: Deep learning-based human pose estimation: a survey. ACM Comput. Surv. **56**(1), 1–37 (2023)