LAC - Latent Action Composition for Skeleton-based Action Segmentation

Di Yang[1] Yaohui Wang[1*] Antitza Dantcheva[1] Quan Kong[3] Lorenzo Garattoni[2]
Gianpiero Francesca[2] François Brémond[1]

[1]Inria, Université Côte d'Azur [2]Toyota Motor Europe [3]Woven by Toyota

{di.yang, yaohui.wang, antitza.dantcheva, francois.bremond}@inria.fr
{lorenzo.garattoni, gianpiero.francesca}@toyota-europe.com quan.kong@woven-planet.global

Abstract

Skeleton-based 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 this context, we propose Latent Action Composition (LAC)[1], 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. Specifically, we design a linear latent space in the generator to represent primitive motion. New composed motions can be synthesized by simply performing arithmetic operations on latent representations of multiple input skeleton sequences. LAC leverages such synthesized sequences, which have large diversity and complexity, for learning visual representations of skeletons in both sequence and frame spaces via contrastive learning. The resulting visual encoder has a high expressive power and can be effectively transferred onto action segmentation tasks by end-to-end fine-tuning without the need for additional temporal models. We conduct a study focusing on transfer-learning and we show that representations learned from pre-trained LAC outperform the state-of-the-art by a large margin on TSU, Charades, PKU-MMD datasets.

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 it is complementary to other modalities such as RGB [31, 7, 27, 23, 22, 48, 36, 63, 4] and optical flow [32, 25]. 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 [20, 19, 5, 70, 50, 11, 55, 73, 9, 38, 21, 74]. While aforementioned approaches have achieved remarkable success, such approaches often focus on *trimmed videos* containing *single actions*, which constitutes a highly simplified scenario. Deviating from this, 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.*, motion performed by a person generally comprises multiple actions (co-occurrence), each with the duration of a few seconds. Towards modeling *long-term dependency* among different actions, expressive skeleton features are required. Current approaches [33, 46, 45, 16] obtain such features through visual encoder such as AGCNs [50] pre-trained on trimmed datasets. However, due to the limited motion information in the trimmed samples, the performance of such features in classifying complex actions is far from satisfactory. Towards addressing 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 paper, we propose Latent Action Composition (LAC), a novel framework aiming at leveraging synthesized composable motion data for self-supervised action representation learning. As illustrated in Fig. 1 (left), as opposed to current self-supervised approaches [33, 46, 45, 16], LAC learns action representations in two steps: a first *action composition* step is then 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

*Corresponding author.
[1]Project website: https://walker1126.github.io/LAC/

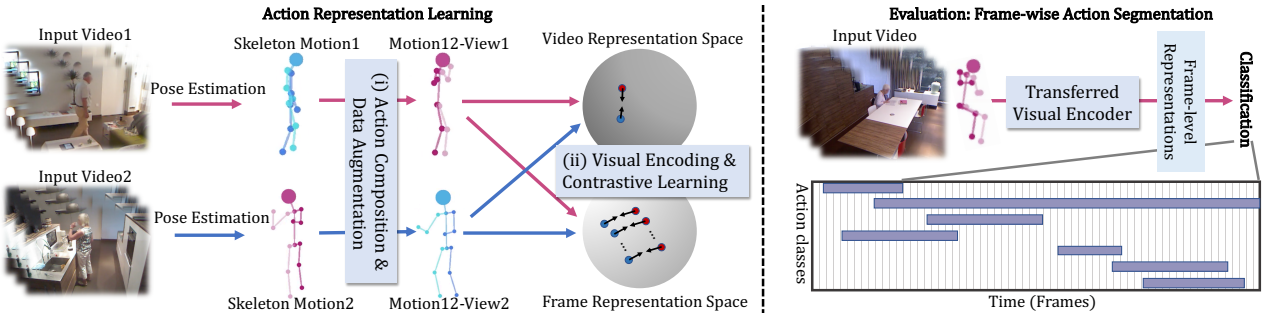

Figure 1. **General pipeline of LAC.** Firstly, in the representation learning stage (left), we propose (i) a novel action generation module to combine skeletons of multiple videos (*e.g.*, 'Walking' and 'Drinking' shown in the top and bottom respectively). We then adopt a (ii) contrastive module to 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 visual encoder is evaluated by transferring to action segmentation tasks.

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 [30]. 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 [73] in a self-supervised manner, without the need for action labels (see Fig. 1 (middle)). It is designed for the resulting visual encoder to be able to maximize the similarity of different skeleton sequences, that are obtained via data augmentation from the same original sequence, across large-scale datasets. Unlike current methods [18, 29, 47, 29, 57, 37, 43, 74] 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 [16, 53].

To assess the performance of LAC, we train the skeleton visual encoder on the large-scale dataset Posetics [73] and we evaluate the quality of the learned skeleton representations (see Fig. 1 (right)) by fine-tuning onto unseen action segmentation datasets (*e.g.*, TSU [16], Charades [53], PKU-MMD [12]). 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.

In summary, the contributions of this paper 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 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.

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 [7]) 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 trans-

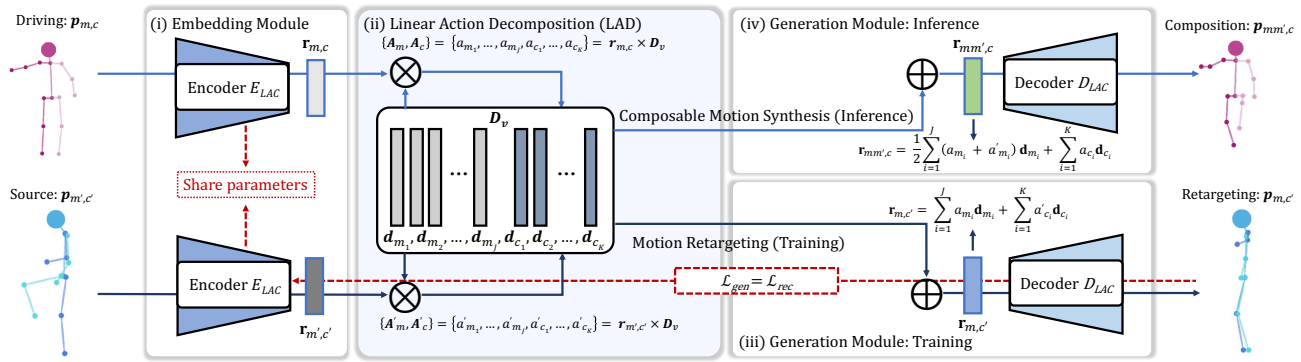

Figure 2. **Overview of the Composable Action Generation model in LAC.** The model consists of a visual encoder $\mathrm{E_{LAC}}$ and a decoder $\mathrm{D_{LAC}}$. In the latent space, we apply Linear Action Decomposition (LAD) by learning a visual action dictionary $\mathbf{D}_v$, which is an orthogonal 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.

late extracted visual features into per-frame action labels. Temporal Convolution Networks (TCNs) [33, 15, 77] and Transformers [14] are generally applied in the second step due to their ability to capture long-term dependencies.

Recently, few methods [13, 16] 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 [50] 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 [49]. 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 [16] and Charades [53]. 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 [73] which is sensitive to composable action. Unlike previous approaches, the pre-trained visual encoder in LAC has stronger representation capability for skeleton sequences compared to previous two-step methods [13, 16] 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.

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 [60, 65, 66, 76, 54, 67], previous image-based motion retargeting approaches explore to leverage structure representations such as 2D human keypoints [62, 3, 8, 74] and 3D human meshes [41, 64] as motion guidance. Recently, self-supervised methods [51, 52, 68] showed remarkable results on human bodies and faces by only relying on data without extract information.

Skeleton-based methods [1, 61, 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 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.

Self-supervised Skeleton Action Representation learning involves extracting spatio-temporal features from numerous unlabeled data. Current methods [72, 37, 59, 43, 74] adopt contrastive learning [58, 69, 28] as the pretext task to learn skeleton representations invariant to data augmentation. However, recent techniques [56, 75, 72, 37, 59, 43, 74] 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 [16, 53]. 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.

3. Proposed Approach

LAC is composed of two modules (see Fig. 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.

3.1. Composable Action Generation

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. 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. 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. 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. 1 and Supplementary Material (Appendix) 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. 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). Hence, we explicitly model the basic 'Static' and 'Motion' transformations using a unified action dictionary for the encoded latent skeleton features. Specifically, we first pre-define 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, any two directions $\mathbf{d_i}, \mathbf{d_j}$ follow the constraint:

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

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:

$$\begin{aligned} \mathbf{r}_m &= \sum_{i=1}^{J} a_{mi}\mathbf{d_{m}}_i, \quad & \mathbf{r}_c &= \sum_{i=1}^{K} a_{ci}\mathbf{d_{c}}_i, \\ \mathbf{r}_{m'} &= \sum_{i=1}^{J} a'_{mi}\mathbf{d_{m}}_i, \quad & \mathbf{r}_{c'} &= \sum_{i=1}^{K} a'_{ci}\mathbf{d_{c}}_i. \end{aligned} \tag{2}$$

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. 3:

$$\begin{aligned} a_{mi} &= \frac{< \mathbf{r}_{m,c} \cdot \mathbf{d_{m}}_i >}{\|\mathbf{d_{m}}_i\|^2}, \quad & a_{ci} &= \frac{< \mathbf{r}_{m,c} \cdot \mathbf{d_{c}}_i >}{\|\mathbf{d_{c}}_i\|^2}, \\ a'_{mi} &= \frac{< \mathbf{r}_{m',c'} \cdot \mathbf{d_{m}}_i >}{\|\mathbf{d_{m}}_i\|^2}, \quad & a'_{ci} &= \frac{< \mathbf{r}_{m',c'} \cdot \mathbf{d_{c}}_i >}{\|\mathbf{d_{c}}_i\|^2}. \end{aligned} \tag{3}$$

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 then

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

Table 1. **Main building blocks** of the autoencoder $\mathrm{E}_{\mathrm{LAC}}$, $\mathrm{D}_{\mathrm{LAC}}$ and the skeleton visual encoder 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.

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. 2 (iii)). The main training loss function is the *reconstruction loss*: $\mathcal{L}_{gen}$=$\mathcal{L}_{rec}$. 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 loss consists of two components: $\mathcal{L}_{rec} = \mathcal{L}_{self} + \mathcal{L}_{target}$. Specifically, at every training iteration, the decoder network $\mathrm{D}_{\mathrm{LAC}}$ 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 $\mathcal{L}_{self}$ and formulated as a standard autoencoder reconstruction loss (see Eq. 4).

$$\begin{aligned}\mathcal{L}_{self} &= \mathbb{E}[\|\mathrm{D}_{\mathrm{LAC}}(\mathbf{r}_m + \mathbf{r}_c) - \mathbf{p}_{m,c}\|^2],\\ \mathcal{L}_{target} &= \mathbb{E}[\|\mathrm{D}_{\mathrm{LAC}}(\mathbf{r}_m + \mathbf{r}_{c'}) - \mathbf{p}_{m,c'}\|^2].\end{aligned} \tag{4}$$

Moreover, at each iteration, the decoder is also encouraged to re-compose new combinations. As the generative module is trained on a synthetic dataset [30] including the cross-character motion retargeting ground-truth skeleton sequences, we can explicitly apply the cross reconstruction loss $\mathcal{L}_{target}$ (see Eq. 4) through the generation. The same reconstruction losses are also computed for $\mathbf{p}_{m',c'}$.

Inference (Motion Composition): 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. 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:

$$\begin{aligned}\mathbf{p}_{mm',c} &= \mathrm{D}_{\mathrm{LAC}}\left(\frac{1}{2}(\mathbf{r}_m + \mathbf{r}_{m'}) + \mathbf{r}_c\right),\\ \mathbf{p}_{mm',c'} &= \mathrm{D}_{\mathrm{LAC}}\left(\frac{1}{2}(\mathbf{r}_m + \mathbf{r}_{m'}) + \mathbf{r}_{c'}\right).\end{aligned} \tag{5}$$

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. 3.2.

3.2. Self-supervised Skeleton Contrastive 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. 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 [28] 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 [28], the queue (memory) [28] 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 can generalize 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 [73] as the skeleton visual encoder E_{V} (see Tab. 1 and Appendix for details). To obtain the global sequence space, we adopt 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. 1). Per-frame features can be obtained by $\mathrm{E_V}$ before the temporal average pooling layer (see Tab. 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 [44] 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 [44] objective is defined as: $\mathcal{L}_q = \mathcal{L}_{q-s} + \mathcal{L}_{q-f}$, where

$$\mathcal{L}_{q-s} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\mathrm{Sim}\left(\mathrm{E_{Vs}}(q), \mathrm{E_{Vs}}(k_p^+)\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathrm{E_{Vs}}(q), \mathrm{E_{Vs}}(k_n^-)\right)}} \Big), \tag{6}$$

$$\mathcal{L}_{q-f} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\sum_{\tau=1}^{T} \mathrm{Sim}\left(\mathrm{E_{Vf}}(q,\tau), \mathrm{E_{Vf}}(k_p^+,\tau)\right)}}{\sum_{n=1}^{N} e^{\sum_{\tau=1}^{T} \mathrm{Sim}\left(\mathrm{E_{Vf}}(q,\tau), \mathrm{E_{Vf}}(k_n^-,\tau)\right)}} \Big), \tag{7}$$

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:

$$\mathrm{Sim}(x, y) = \frac{\phi(x) \cdot \phi(y)}{\|\phi(x)\| \cdot \|\phi(y)\|} \cdot \frac{1}{Temp}, \tag{8}$$

where $Temp$ refers to the temperature hyper-parameter [69], and ϕ is a learnable mapping function (*e.g.*, a MLP projection head [24]) 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. 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 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 [39].

4. Experiments and Analysis

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. See Appendix for implementation details, limitation discussion and additional studies, *e.g.*, computational cost analysis.

Methods	**Mod.**	**TSU** CS(%)	CV(%)	**Charades** mAP(%)
TGM [46]	RGB	26.7	-	13.4
PDAN [15]	RGB	32.7	-	23.7
SD-TCN [16]	RGB	29.2	18.3	21.6
MS-TCT [14]	RGB	33.7	-	25.4
Bi-LSTM [26]	Skeleton	17.0	14.8	8.2
TGM [46]	Skeleton	26.7	13.4	9.0
SD-TCN [16]	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 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.

Methods	**Mod.**	**PKU-MMD** mAP@IoU 0.1(%)	0.3(%)	0.5(%)
GRU-GD [42]	RGB	82.4	81.3	74.3
SSTCN-GD [13]	RGB	83.7	82.1	76.5
Augmented-RGB [13]	RGB	86.3	84.5	81.1
JCRRNN [39]	Skeleton	45.2	-	32.5
Convolution Skeleton [12]	Skeleton	49.3	31.8	12.1
Skeleton boxes [34]	Skeleton	61.3	-	54.8
Hi-TRS [10]	Skeleton	-	-	67.3
Window proposal [35]	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 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.

4.1. Datasets and Evaluation Protocols

Posetics [73] contains 142,000 real-world trimmed video clips from Kinetics-400 [7] with corresponding 2D and 3D skeletons. We use Posetics to pre-train the contrastive model of LAC with skeleton data and we study the transfer-learning on skeleton-based action segmentation.

Toyota Smarthome Untrimmed (TSU) [16] 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 [71] for the experiments. For evaluation, we report *per-frame* mAP (mean Average Precision) as [15, 14] following the cross-subject (CS) and cross-view (CV) evaluation protocols.

Methods	Pre-training	Training data	Toyota Smarthome Untrimmed		PKU-MMD (IoU=0.1)		Charades
			CS(%)	CV(%)	CS(%)	CV(%)	mAP(%)
Random init. [73]	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. [73]	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. 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.

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 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.

Charades [53] 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 [71]. 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 [12] 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 [42] on the frame-level predictions to get the action boundaries.

Mixamo [30] is a 3D animation collection, which contains elementary actions and various dancing moves. We use such a synthetic dataset for training and evaluating the generation module in LAC prior to contrastive learning on Posetics.

4.2. 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. 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. 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, without pre-training, the accuracy of the visual encoder [73] 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 [26, 46, 16, 39, 34, 12, 10, 35] on the real-world datasets TSU and Charades (see Tab. 2) and also laboratory dataset PKU-MMD (see Tab. 3). As previous approaches are based on supervised pre-training on large-scale datasets [40, 7], we also report our supervised results. The results in Tab. 2 show that LAC, even with self-supervised pre-training, outperforms all previous skeleton-based approaches [26, 46, 16] with supervised pre-training on our main target real-world datasets in a

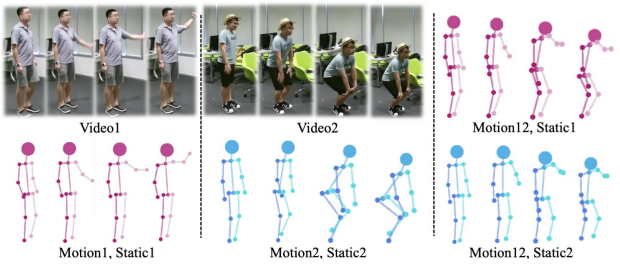

Figure 3. **Motion composition visualization.** The input pair of videos and corresponding skeleton sequences (left) have simple motions. The generated skeleton sequences (right) are composed by both motions while keeping their respective viewpoint and body size ('Static') invariant.

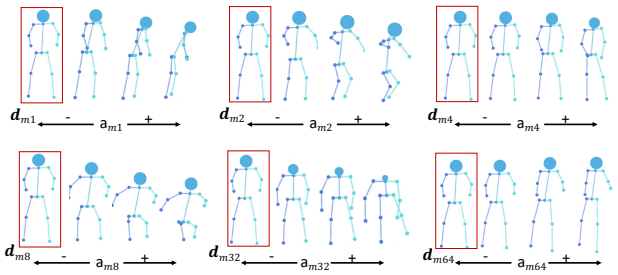

Figure 4. **Linear manipulation of six 'Motion' directions** in $\mathbf{D}_v$ on a skeleton sequence. Results indicate that each direction represents a meaningful motion transformation from a 'reference pose' marked in red (*e.g.*, $\mathbf{d}_{\mathbf{m}8}$ for squat, $\mathbf{d}_{\mathbf{m}32}$ for bending over).

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 [15, 16, 14, 46, 13]. 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 [13, 17] to further improve the performance.

4.3. 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 training and test sets on this dataset and follow the same setting and protocol described in [3, 74]. 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. 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 [62, 3, 74]. Hence, we set J=128 and K=32 for all other experiments.

Methods	Mean Square Error
NKN [62]	1.51
MotionRetargeting2D [3]	0.96
ViA [74]	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 6. Quantitative comparisons of LAC to other SoTA motion retargeting methods on the Mixamo dataset.

Motion Direction Interpretation and Visualization: We visualize an example of motion composition inference of two videos. Fig. 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 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. More real-world examples with different viewpoints are provided in the Appendix.

4.4. Ablation Study

To understand the contribution of the two individual components of LAC, we conduct ablation experiments on our

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

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 [73] that is pre-trained on the trimmed dataset (*i.e.*, Posetics) in a general contrastive learning strategy [28] without using composable motions and frame-level contrast for action segmentation. The results in Tab. 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 [16], combining motions from two skeleton sequences in the pre-training stage can significantly improve the representation ability of the visual encoder and generalize better to real-world untrimmed action segmentation tasks (see Tab. 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. 7 L2 suggest that frame-wise contrast with uniformly sampling every 4 frames is the most effective to improve the action segmentation accuracy.

4.5. Further Discussion

Transfer Learning vs. Self Pre-training: Our target is to train a generic skeleton encoder that can fit different downstream tasks. Hence, similar to current RGB-based methods using large-scale dataset such as Kinetics [7, 6] for pre-training, our model is pre-trained on the large-scale Posetics dataset to learn a generic skeleton representation. Such 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

Dataset	TGM [46]	SD-TCN [16]	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 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.

LAC with SoTA methods [46, 16] in Tab. 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 [46, 16], 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 [46, 16]. Moreover, current untrimmed datasets are not large enough, the generated actions have less diversity, so the representation ability of the skeleton encoder is less impressive than pre-training on Posetics.

5. Conclusion

In this work, 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. 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.

Acknowledgements: This work was supported by Toyota Motor Europe (TME) and the French government, through the 3IA Cote d'Azur Investments In the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

References

[1] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.*, 2020. 3

[2] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM Trans. Graph.*, 2020. 3

[3] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *ACM TOG*, 2019. 3, 5, 8

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *ICCV*, 2021. 1

[5] C. Caetano, F. Brémond, and W. Schwartz. Skeleton image representation for 3D action recognition based on tree structure and reference joints. *SIBGRAPI*, 2019. 1

[6] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *CoRR*, 2019. 9

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 2, 6, 7, 9

[8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. In *ICCV*, 2019. 3

[9] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *ICCV*, 2021. 1

[10] Yuxiao Chen, Long Zhao, Jianbo Yuan, Yu Tian, Zhaoyang Xia, Shijie Geng, Ligong Han, and Dimitris N. Metaxas. Hierarchically self-supervised transformer for human skeleton representation learning. In *ECCV*, 2022. 6, 7

[11] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *AAAI*, 2021. 1

[12] Liu Chunhui, Hu Yueyu, Li Yanghao, Song Sijie, and Liu Jiaying. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv:1703.07475*, 2017. 2, 6, 7

[13] Rui Dai, Srijan Das, and François Bremond. Learning an augmented rgb representation with cross-modal knowledge distillation for action detection. In *ICCV*, 2021. 3, 6, 8

[14] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael Ryoo, and Francois Bremond. MS-TCT: Multi-Scale Temporal ConvTransformer for Action Detection. In *CVPR*, 2022. 3, 6, 8

[15] Rui Dai, Srijan Das, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Pdan: Pyramid dilated attention network for action detection. In *WACV*, 2021. 3, 6, 8

[16] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection. *IEEE TPAMI*, 2022. 1, 2, 3, 4, 6, 7, 8, 9

[17] Srijan Das, Rui Dai, Di Yang, and Francois Bremond. Vpn++: Rethinking video-pose embeddings for understanding activities of daily living. *IEEE TPAMI*, 2021. 8

[18] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *CVPR*, 2022. 2

[19] Wenwen Ding, Kai Liu, Fei Cheng, and Jin Zhang. Stfc: Spatio-temporal feature chain for skeleton-based human action recognition. *JVCIR*, 2015. 1

[20] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 1

[21] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, 2022. 1

[22] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 1

[23] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1

[24] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, 2021. 6

[25] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1

[26] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *IJCNN*, 2005. 6, 7

[27] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3D residual networks for actio recognition. In *ICCVW*, 2017. 1

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 3, 5, 9

[29] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. Self-supervised video representation learning by context and motion decoupling. In *CVPR*, 2021. 2

[30] Adobe Systems Inc. Mixamo. https://www.mixamo.com. https://www.mixamo.com. *Accessed: 2018-12-27.*, 2018. 2, 5, 7

[31] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 2013. 1

[32] Simonyan Karen and Zisserman Andrew. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1

[33] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017. 1, 3

[34] Bo Li, Huahui Chen, Yucheng Chen, Yuchao Dai, and Mingyi He. Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network. In *ICMEW*, 2017. 6, 7

[35] Chuankun Li, Yonghong Hou, Pichao Wang, and Wanqing Li. Joint distance maps based action recognition with convolutional neural networks. In *ICMEW*, 2017. 6, 7

[36] Kunchang Li, Xianhang Li, Yali Wang, Jun Wang, and Yu Qiao. Ct-net: Channel tensorization network for video classification. In *ICLR*, 2021. 1

[37] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3d human action representation learning via cross-view consistency pursuit. In *CVPR*, 2021. 2, 3

[38] Maosen Li, Siheng Chen, Zihui Liu, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. Skeleton graph scattering networks for 3d skeleton-based human motion prediction. In *ICCVW*, 2021. 1

[39] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *ECCV*, 2016. 6, 7

[40] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3D human activity understanding. *IEEE TPAMI*, 2020. 7

[41] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019. 3

[42] Zelun Luo, Jun-Ting Hsieh, Lu Jiang, Juan Carlos Niebles, and Li Fei-Fei. Graph distillation for action detection with privileged modalities. In *ECCV*, 2018. 6, 7

[43] Yunyao Mao, Wengang Zhou, Zhenbo Lu, Jiajun Deng, and Houqiang Li. Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In *ECCV*, 2022. 2, 3

[44] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *arXiv:1807.03748*, 2018. 6

[45] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018. 1

[46] AJ Piergiovanni and Michael S. Ryoo. Temporal gaussian mixture layer for videos. In *ICML*, 2019. 1, 6, 7, 8, 9

[47] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael Ryoo. Self-supervised video transformer. In *CVPR*, 2022. 2

[48] M. Ryoo, A. Piergiovanni, Juhana Kangaspunta, and A. Angelova. Assemblenet++: Assembling modality representations via attention connections. *ECCV*, 2020. 1

[49] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 3

[50] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 1, 3

[51] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 2019. 3

[52] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *CVPR*, 2021. 3

[53] Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 3, 4, 7

[54] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2022. 3

[55] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM MM*, 2020. 1

[56] Yukun Su, Guosheng Lin, and Qingyao Wu. Self-supervised 3d skeleton action representation learning with motion consistency and continuity. In *ICCV*, 2021. 3

[57] Chen Sun, Arsha Nagrani, Yonglong Tian, and Cordelia Schmid. Composable augmentation encoding for video representation learning. In *ICCV*, 2021. 2

[58] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 3

[59] Guo Tianyu, Liu Hong, Chen Zhan, Liu Mengyuan, Wang Tao, and Ding Runwei. Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *AAAI*, 2022. 3

[60] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 3

[61] Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. Contact-aware retargeting of skinned motion. In *ICCV*, 2021. 3

[62] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *CVPR*, 2018. 3, 8

[63] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, 2021. 1

[64] Tuanfeng Y Wang, Duygu Ceylan, Krishna Kumar Singh, and Niloy J Mitra. Dance in the wild: Monocular human animation with neural dynamic appearance synthesis. In *3DV*, 2021. 3

[65] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G3AN: Disentangling appearance and motion for video generation. In *CVPR*, 2020. 3

[66] Yaohui WANG, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. ImaGINator: Conditional spatio-temporal gan for video generation. In *WACV*, 2020. 3

[67] Yaohui Wang, Francois Bremond, and Antitza Dantcheva. Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. *arXiv:2101.03049*, 2021. 3

[68] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. In *ICLR*, 2022. 3

[69] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3, 6

[70] S. Yan, Yuanjun Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*, 2018. 1

[71] Di Yang, Rui Dai, Yaohui Wang, Rupayan Mallick, Luca Minciullo, Gianpiero Francesca, and Francois Bremond. Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos. In *WACV*, 2021. 6, 7

[72] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Self-supervised video pose representation learning for occlusion-robust action recognition. In *FG*, 2021. 3

[73] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Unik: A unified framework for real-world skeleton-based action recognition. In *BMVC*, 2021. 1, 2, 3, 5, 6, 7, 9

[74] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Via: View-invariant skeleton action representation learning via motion retargeting. *arXiv:2209.00065*, 2022. 1, 2, 3, 8

[75] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C. Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *ICCV*, 2021. 3

[76] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 3

[77] Chuhan Zhang, Ankush Gputa, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *CVPR*, 2021. 3