frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 23 August 2023
DOI 10.3389/fcomp.2023.1203901

OPEN ACCESS

EDITED BY
Mohamed Chetouani,
Sorbonne Université, France

REVIEWED BY
Séverine Dubuisson,
Aix-Marseille Université, France
Davide Boscaini,
Bruno Kessler Foundation (FBK), Italy

*CORRESPONDENCE
Cigdem Beyan
✉ cigdem.beyan@unitn.it

RECEIVED 11 April 2023
ACCEPTED 31 July 2023
PUBLISHED 23 August 2023

CITATION
Paoletti G, Beyan C and Del Bue A (2023) SKELTER: unsupervised skeleton action denoising and recognition using transformers. *Front. Comput. Sci.* 5:1203901. doi: 10.3389/fcomp.2023.1203901

SKELTER: unsupervised skeleton action denoising and recognition using transformers

Giancarlo Paoletti[1,2], Cigdem Beyan[1,3]* and Alessio Del Bue[1]

[1]Pattern Analysis and Computer Vision Research Line, Fondazione Istituto Italiano di Tecnologia, Genoa, Italy, [2]Electrical, Electronics, and Telecommunication Engineering and Naval Architecture Department, University of Genoa, Genoa, Italy, [3]Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Unsupervised Human Action Recognition (U-HAR) methods currently leverage large-scale datasets of human poses to solve this challenging problem. As most of the approaches are dedicated to reaching the best recognition accuracies, no attention has been put into analyzing the resilience of such methods given perturbed data, a likely occurrence in real in-the-wild testing scenarios. Our first contribution is to systematically validate the decrease in performance of current U-HAR state-of-the-art using perturbed or altered data (e.g., obtained by removing some skeletal joints, rotating the entire pose, and injecting geometrical aberrations). Then, we propose a novel framework based on a transformer encoder–decoder with remarkable de-noising capabilities to counter such perturbations effectively. Moreover, we also present additional losses to have robust representations against rotation variances and provide temporal motion consistency. Our model, SKELTER, shows limited drops in performance when skeleton noise is present compared with previous approaches, favoring its use in challenging in-the-wild settings.

KEYWORDS

action recognition, autoencoder, transformers, full-body movement, skeletal data, unsupervised feature learning

1. Introduction

Human Action Recognition (HAR), performed based on the visual analysis of the body, including posture and movements of individuals, brings in a rich source of information for numerous tasks, from video surveillance to biomedical applications. HAR from video sequences is a challenging problem for several reasons, such as the diverse camera viewpoints, similarity of visual contents, range of poses and size among subjects performing the action, different illumination, and weather conditions. Several approaches use 3D-skeleton data instead of raw videos due to the intrinsic properties of this pre-processed information: being lightweight, privacy-preserving, free of background noises, abrupt changes in lighting condition, and so forth (Cao et al., 2017; Yan et al., 2018; Si et al., 2019; Xu et al., 2020a,b; Beyan et al., 2021).

The successes in skeleton-based HAR (Zhang et al., 2017; Yan et al., 2018; Cheng et al., 2020) primarily rely on the supervised learning paradigm, which requires a significant amount of manually labeled data. However, data annotation is expensive, time-consuming, and prone to human errors (Paoletti et al., 2021a,b, 2022). Moreover, action classes may vary significantly from dataset to dataset, while several methods lack enough generalization to apply in different scenarios without requiring extra annotations. As (recent) alternative, unsupervised HAR approaches (Zheng et al., 2018; Lin et al., 2020; Nie et al., 2020; Su et al., 2020a; Paoletti et al., 2021b; Rao et al., 2021) (called U-HAR for the rest of this study) are tremendously increasing their impact while competing to reduce the performance gap with the fully supervised counterparts. As most of the approaches are dedicated to reaching the best recognition accuracies, no attention has been put into analyzing the resilience of such methods given perturbed data, a likely occurrence in real in-the-wild testing scenarios. The benchmark datasets, on which the existing U-HAR methods are tested, were recorded using depth sensors (Wang et al., 2019a) (e.g., by using Microsoft Kinect) in relatively controlled experimental settings, being free from several challenges such as noisy data and severe occlusions, thus being far from realistic scenarios. It is also important to notice that in real-world conditions, there can be errors in sensors resulting in missing frames and/or errors occurring due to the misdetection of the pose estimators. It is paramount to evaluate the resilience of proposed methodologies before deploying them in practical settings, particularly in applications such as those related to biomedical [e.g., action recognition-based elderly monitoring Petrushin et al. (2022), fall detection Cebanov et al. (2019), action-based anomaly detection Parvin et al. (2018)], where highly robustness and denoising capabilities are essential.

The first contribution of this study is to provide a systematic analysis of the state-of-the-art (SOTA) skeleton-based U-HAR methods when evaluated on perturbed and altered data, simulating several real-world challenges, e.g., noise, clutter, occlusions, and geometrical distortions. Consequently, we present an extensive set of perturbations and alterations to simulate in-the-wild scenarios for HAR (e.g., obtained by removing some skeletal joints, rotating the entire pose, injecting geometrical aberrations, etc., see Section 3.2) and verifying the decrease in performance of current SOTA, evidencing cases where such loss is more predominant.

On the other hand, we also propose a novel framework, SKELTER (SKELeton TransformER), which is capable of learning robust representations from the spatiotemporal 3D-skeletal data in an unsupervised fashion (Section 3.3). Our proposed encoder–decoder architecture uses transformers that get inputs as 3D-skeletal data over time. A transformer-based encoder–decoder architecture is chosen due to its superior ability to encode skeletal joint information across the entire temporal span. At the same time, its attention modules provide context for any position in the input sequence of sequential data, weighing their influence on different temporal parts. We also devise additional components to SKELTER to obtain robustness toward skeletal rotations and temporal consistencies w.r.t. time-frames alterations (Section 3.2).

Overall, the performance of our method is compared with SOTA skeleton-based U-HAR when tested on perturbed and altered data, which are applied on NTU-60 (Shahroudy et al., 2016) and NTU-120 (Liu et al., 2019) datasets' cross-subject cross-view and cross-setup splits. The comprehensive analysis shows that SKELTER is more robust against several data perturbations and alterations than SOTA, showing its better denoising capability.
The main contributions of this study are listed as follows.

- For the first time, we evaluate SOTA skeleton-based U-HAR methods on perturbed and altered data, which simulate in-the-wild challenging scenarios. We believe that the results allow the community better to understand the existing methods' applicability to real-world scenarios.
- We present a novel method based on transformers (SKELTER), which processes the skeletal data within a spatiotemporal pipeline by integrating a multi-attention mechanism. This encoder–decoder structure relies on mean squared error (MSE), so the feature learning is fully unsupervised. In addition, we devise two additional losses: one for resulting in more robust representations against rotation variances (Section 3.3.5) and the other is to handle the possible temporal motion consistency by integrating triplet loss (Section 3.3.6).
- We show that SKELTER is more resilient than the SOTA skeleton-based U-HAR methods when subject to data perturbations and alterations, showing that it can handle various real-world challenges (reconstructing smoother and uncorrupted skeleton poses) compared with other approaches.

The rest of the study is organized as follows. Section 2 summarizes SOTA unsupervised skeleton-based human action recognition approaches and transformers. The proposed method is introduced in Section 3, specifying the application scenarios for skeleton-based HAR and the related data perturbation and alteration which could likely occur. Section 4 presents the experimental analysis, datasets, implementation details, and results. Section 5 reports qualitative analysis and a case study of a real-life scenario. Finally, we conclude the study with a summary and discussions in Section 6.

2. Related work

The HAR literature is very extensive; we refer to the following surveys (Poppe, 2010; Presti and La Cascia, 2016; Xing and Zhu, 2021) for a general overview. This study focuses on skeleton-based U-HAR approaches and reviews transformers as strictly related to our backbone architecture.

2.1. Skeleton-based U-HAR

To solve *supervised* skeleton-based HAR task, early approaches leveraged upon hand-craft features (Ni et al., 2011; Wang et al., 2012; Ohn-Bar and Trivedi, 2013; Oreifej and Liu, 2013; Evangelidis et al., 2014; Vemulapalli et al., 2014; Yang and Tian, 2014; Vemulapalli and Chellapa, 2016). As for deep neural networks, recent studies are based on recurrent neural networks (RNNs) (Du et al., 2015b; Shahroudy et al., 2016; Zhang et al., 2017, 2019; Song et al., 2018), convolutional neural networks (CNNs) (Du et al., 2015a; Ke et al., 2017; Li et al., 2017; Liang et al., 2019), and graph convolutional networks (GCNs) (Yan et al., 2018; Lu et al., 2019; Shi et al., 2019a,b; Si et al., 2019; Wang et al., 2020; Xu et al., 2020b; Zhang et al., 2020a,b), demonstrating the benefits of learning intrinsic properties of skeletal actions performed over time. As for *unsupervised* skeleton-based HAR (Zanfir et al., 2013; Martinez et al., 2017; Ben Tanfous et al., 2018; Gui et al., 2018; Li et al.,

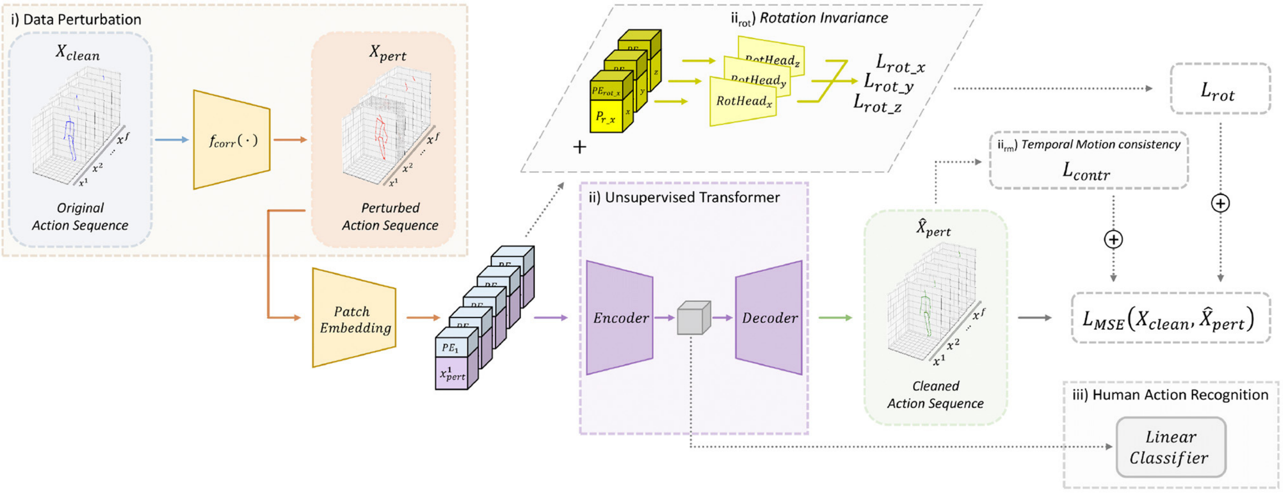

FIGURE 1
Overall methodology (best viewed in color). *(i) Data perturbation:* given a clean skeletal action sequence X_{clean} (blue skeleton), a plausible real-world data perturbation is simulated and applied to the data sample to obtain the input sequence X_{pert} (red skeleton). *(ii) Unsupervised transformer:* our proposed approach, SKELTER, is a transformer-based Encoder and Decoder architecture, able to learn how to denoise the X_{pert} data and reconstruct the animated pose as $\hat{X}_{pert}$ (green skeleton), using the reconstruction loss $\mathcal{L}_{\text{MSE}}$. *(ii$_{rot}$) Rotation invariance*: in cases of rotated data, *RotHead* are plugged into SKELTER (one for each 3D axis). The rotation loss L_{rot} ensures a correct prediction of the rotation angles, granting invariant properties toward 3D rotations. *(iirm) Temporal motion consistency*: the triplet margin loss $\mathcal{L}_{\text{contr}}$ ensures temporal motion consistency in altered data. *(iii) Human action recognition (inference stage):* to perform HAR, a linear classifier is set on the top of the learned feature representations.

2018), several representation learning methods first encode the skeleton sequences into a latent space, followed by a reconstruction stage, then a linear classifier is trained using the frozen latent representations. Methods such as PCRP (Xu et al., 2023) and Predict & Cluster (P&C) (Su et al., 2020a) are built on an encoder–decoder structure. The former reconstructs skeletal data using EM with learnable class prototypes; the latter uses RNNs as an encoder. LongTGAN (Zheng et al., 2018) is a generative adversarial network (GAN)-based method, which uses GRUs with a mixture of adversarial loss and additional inpainting tasks. In contrast, the GAN-based encoder of EnGAN (Kundu et al., 2019) learns representations of skeletal body poses in time. SeBiReNet (Nie et al., 2020) is a Siamese denoising autoencoder tested on feature disentanglement, pose denoising, and unsupervised cross-view HAR. According to Paoletti et al. (2021b), unsupervised feature learning is performed with a convolutional residual autoencoder, demonstrating the benefits of performing residual convolutions to learn representations with spatiotemporal convolutions jointly. In the same study, the authors also used Laplacian regularization to inject the intrinsic knowledge of the connectivity patterns of the body into the network. Li et al. (2021) process the joint, motion, and bone (see study for their definition) information altogether instead of using only skeleton data. U-HAR results are improved using these three modalities within a contrastive learning schema. ISC (Thoker et al., 2021) leverages inter-skeleton contrastive learning and spatiotemporal augmentations to learn invariances w.r.t. skeleton representations. AimCLR (Guo et al., 2022) builds upon contrastive methods, and it can obtain robust representation from extreme augmentations and novel movement patterns.

This study evaluates the aforementioned U-HAR methods when trained and/or tested on corrupted data. Hendrycks and Dietterich (2018) and Yi et al. (2021) tackled similar approaches but for different tasks (i.e., not U-HAR), and the type of data were images and videos (i.e., not skeletons). Different from Hendrycks and Dietterich (2018) and Yi et al. (2021), herein, such data represent real-world challenges, such as noise, clutter, and occlusions. Moreover, we propose a novel method based on an encoder–decoder transformer architecture, showing better performance for the realistic scenarios where perturbed and altered data exist, w.r.t. the already existing U-HAR methods.

2.2. Transformers

Since their inception in NLP research (Vaswani et al., 2017; Brown et al., 2020), transformers have gained popularity in different tasks such as for machine translation (Wang et al., 2019b; Yang et al., 2019), visual question answering (Lu et al., 2019; Su et al., 2020b), action recognition (Girdhar et al., 2019; Bertasius et al., 2021), and human pose estimation (Zheng et al., 2021), to name a few. Vision Transformer (ViT) (Dosovitskiy et al., 2021) is the first pure-transformer model deployed for image classification that was trained on large-scale datasets such as Imagenet-21K (Ridnik et al., 2021) and JFT-300M, achieving remarkable results. On the other hand, ACTOR (Petrovich et al., 2021) is a transformer-based conditional VAE, which can generate action-conditioned human motions by sampling from a sequence-level latent vector. The success of transformers mainly relies on their property to establish long-range connections among time-series data, w.r.t. shorter connections could occur in RNNs or LSTMs. In this study, we inherit ViT (Dosovitskiy et al., 2021) and ACTOR (Petrovich et al., 2021) in our encoder and decoder architecture such that we learn the skeleton-data representations by processing them in spatial and temporal dimensions. We define a spatial transformer, which considers the temporal dimension of the data in terms of temporal position embeddings. This encoder–decoder structure is trained with MSE loss between the input representations and the reconstructed skeletons. The hierarchical transformer by Cheng et al. (2021) fuses part-based skeletal features to higher-level representations using self-attention mechanisms from transformers.

Although the common final task of U-HAR, this model formulates the unsupervised representation learning as a classification problem, predicting the motion direction of masked poses. However, it does not aim to perform data denoising. At the same time, to the best of our knowledge, our approach is the first transformer-based solution, *specifically designed to tackle data denoising for the U-HAR.*

3. Methodology

First, we introduce a description of applicability scenarios for HAR in Section 3.1, followed by our proposed perturbations and alterations in Section 3.2, and then our method, SKELTER, in Section 3.3. The two additional losses devised to obtain robustness toward skeletal rotations and temporal consistencies w.r.t. time-frames alterations, injected into the SKELTER pipeline, are given at the end. For action inference, we use the common protocol of unsupervised feature learning, i.e., linear evaluation (Zheng et al., 2018), such that the latent features learned without supervision are given to a linear classifier to perform HAR. Notably, the inference stage is the same with all SOTA we compare with SKELTER. All these stages are presented in Figure 1. To prove their coherence w.r.t. a practical application, Section 5.1 reports a comparison between the perturbed datasets and a test case of the aforementioned *real-world scenario*.

3.1. Application scenarios for skeleton-based HAR

Concerning skeleton-based HAR experimental pipelines, several steps are involved, which can be summarized into two main components: (1) obtaining 3D-keypoints from RGB videos, usually as sequences of image frames using specific equipment or using pose estimator algorithms and (2) deploying state-of-the-art model capable of correctly classify the correspondent action. The predominant choice bends on benchmark datasets obtained within *staged* scenarios. For example, NTU-60 (Shahroudy et al., 2016) and NTU-120 (Liu et al., 2019) are recorded using depth sensors (i.e., Microsoft Kinect v2) inside a constrained and well-controlled setup to achieve the best quality of data.

On the other hand, these conditions could not always be guaranteed in realistic scenarios. For this less common setup, e.g., a surveillance online video feed, a continuous stream of RGB frames represents the input where pose estimator software infers the initial 2D-keypoints from the RGB frames (Cao et al., 2017) and lifts them into the 3D space (Pavllo et al., 2019). Starting from the nature of the scenario itself (online frame-wise 3D pose estimation), depending on the conditions of the scene itself (e.g., overcrowded frames, bad camera recording quality, errors in camera calibration, and missing frames from recording), and accounting abrupt and unforeseen events (such as noisy estimation, severe occlusions, mis-detected keypoints), the quality of keypoints estimation could be severely affected in this type of scenario.

Due to its unpredictable nature, the quantity, and variability of these unexpected events, action classification from severely affected 3D key points could represent a challenging task for U-HAR SOTA models, which often overlook the particular conditions of this real-world scenario. Therefore, we propose SKELTER as a needed and robust model capable of correctly classifying those actions, regardless of the conditions of given skeletal samples.

3.2. Data perturbation and alteration for HAR

Existing skeleton-based U-HAR methods were evaluated on commonly-used datasets, e.g., NTU-60 (Shahroudy et al., 2016), and NTU-120 (Liu et al., 2019), by applying pre-processing steps (normalization and camera pre-registration). Although such pre-processing represent undoubtedly a common practice to obtain robust features from the skeletal action sequences, the ingredients to apply it might not always be available in real-world processing. In addition, the methods trained on optimum conditions (such as without considering the noise, missing joints) might result in poor performance in their unconstrained real-world processing.

Since the main scope of this study is to evaluate the SOTA and SKELTER in the presence of perturbed and altered data, the first step is, therefore, to define a wide range of *perturbations* (i.e., Gaussian Noise, Joint Outlier, Joint Removal, Limbs Removal, Axis Removal, Shear, and Subtract) and *alterations* (i.e., Rotation and Reverse Motion). Figure 2 illustrates that the blue skeletal poses represent the original data, whereas the red poses represent the transformation applied.

3.2.1. Data perturbation

Gaussian noise (GN): Additive Gaussian noise is applied over the joints (with a mean equal to zero and standard deviation equal to 0.05) to simulate noisy positions caused by the pose estimator model.

Joint outlier (JO): For each skeletal sample, a random joint is selected and its 3D coordinates altered by adding, for each axis, a fixed value within a range of $[-1, 1]$ to simulate an outlier joint that severe incorrect estimations in the camera feed can cause.

Joint removal (JR): For each sample action sequence, a subsection of temporal frames is selected, i.e., a random amount of frames, up to 25% of the entire length, and within these selected frames, a subsection of joints is chosen and set to zero. This random-conditioned selection ensures the simulation of a plausible real-world scenario in which some joints could not be detected.

Limbs removal (LR): For each sample action sequence, the occlusion of an entire limb is simulated by randomly selecting one of the four groups of joints (i.e., left and right arms, left and right legs) and setting their coordinates to zero to simulate e.g., common

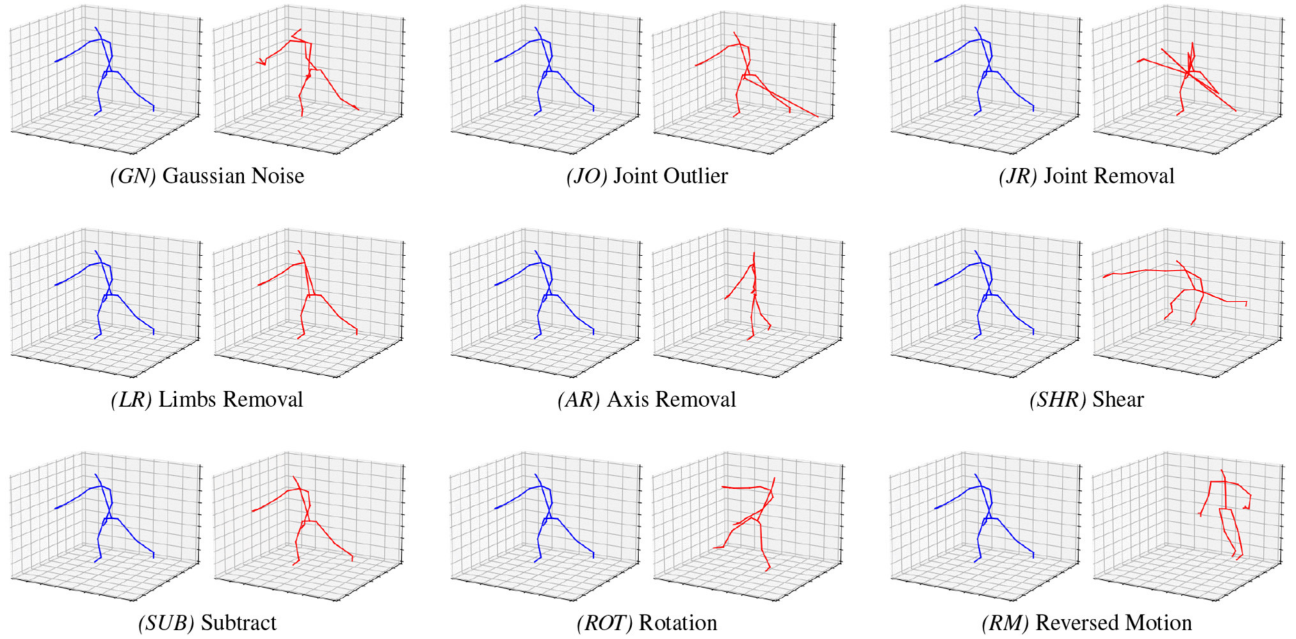

FIGURE 2
The proposed data perturbation and alteration strategies. Starting from the clean "Throw" skeleton action sequences picked from the NTU-60 (Shahroudy et al., 2016) Cross-View split (blue poses), data perturbations or alterations are applied (red poses). Each blue-and-red couple is a sample of a different perturbation (GN, JO, JR, LR, AR, SHR, and SUB, see Section 3.2.1 for definitions) or alteration (ROT and RM, see Section 3.2.2 for definitions).

severe occlusions such as "legs occluded due to the subject being sat at a desk."

Axis removal (AR): This refers to setting an entire axis that is selected randomly to zero. This simulates a failure of a pose estimator to infer 3D poses and as a general-purpose 2D-to-3D hallucination capability of models that are not natively designed for this kind of task.

Shear (SHR): Shear simulates the variations in the camera orientation. Each skeletal joint is displaced in a fixed direction (e.g., slant joints with a random angle $S \in [-1, 1]$), using a linear mapping matrix:

$$\boldsymbol{\Omega}_s = \begin{bmatrix} 1 & S_X^Y & S_X^Z \\ S_Y^X & 1 & S_Y^Z \\ S_Z^X & S_Z^Y & 1 \end{bmatrix} \tag{1}$$

Subtract (SUB): The entire skeleton in shifts 3D space by selecting a random joint and setting it as the new root joint. This is a simulation of the situations arising when, e.g., a pose estimator fails to correctly detect a skeletal pose, resulting in an abrupt shift of spatial coordinates.

3.2.2. Data alteration

Rotation (ROT): 3D-skeletal data are rotated along XYZ axes, using the respective rotation matrices given in Equation (2). Rotation is involved in testing the strength of a method under view-point variations, e.g., in scenarios such as camera surveillance where a skeleton pose of a person is captured through multi-camera settings. To simulate plausible contexts, a randomly-sampled Z-axis rotation along all 360 degrees is applied, whereas on X and Y axes, the rotation angles' range spans in-between $[-30, 30]$ degrees.

$$\boldsymbol{\Omega}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \boldsymbol{\Omega}_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},$$

$$\boldsymbol{\Omega}_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Reversed motion (RM): The order of the time frames of a given sample is randomly reversed (with a 50% chance), to ensure a model learns human motion when a reversed perspective is shown. This is useful especially when SKELTER is trained on datasets that contain ambiguous or subtle actions, e.g., actions such as *"wear a shoe"* or *"take off a shoe,"* which are theoretically similar but different w.r.t. motion execution and action label.

3.3. Proposed method

Our method, SKELTER, was designed by following the general direction endowed by ViT (Dosovitskiy et al., 2021) for embedding the input data and ACTOR (Petrovich et al., 2021) for the overall encoder/decoder structure. The training paradigm fosters the model to learn robust features for HAR, describing below its components in detail. Following that, additional modules and losses of the method were defined to pursue robustness toward skeletal rotations (Section 3.3.5) and temporal consistencies (Section 3.3.6), to disambiguate between specular actions w.r.t. time-frames alterations.

3.3.1. Transformer-based encoder and decoder

On our frame-wise skeleton encoder, the temporal frames of the given sample represent the input tokens for the transformer module to capture their global dependencies.

The input sequence is defined as $X \in \mathbb{R}^{f\times(3J)}$, where f is the number of time-frames of the action sequence, and J represents the number of joints for each 3D pose.

Skeleton data, which can be (in general) clean or (in this case) perturbed denoted as $\{X_{clean}, X_{pert}\}$ respectively, are fed into the transformer-based encoder and decoder sharing the same architecture (described below). Each 3D-skeletal pose is defined as $X^i_{pert} \in \mathbb{R}^{1\times(3J)}$, $i = 1, 2, \ldots, f$ of each time-frame f as a *patch token*.

Subsequently, the *patch embedding* $P \in \mathbb{R}^{f\times d}$ is the linear projection of joints into a high-dimensional feature, where d is the embedding dimension, using a trainable linear layer $E \in \mathbb{R}^{(3J)\times d}$:

$$P = [x^1E,\ x^2E,\ \ldots,\ x^fE] + PE \tag{3}$$

The *positional embedding* $PE \in \mathbb{R}^{f\times d}$, inherited from Vaswani et al. (2017), come in aid to the transformer module to maintain positional information about the skeletal sequence (i.e., the temporal frame order) as:

$$PE_{(f,2d)} = \sin(f/10000^{2i/d}), \tag{4}$$
$$PE_{(f,2d+1)} = \cos(f/10000^{2i/d}) \tag{5}$$

where i is the dimension of the embedding.

3.3.2. Attention in transformers

The core principle of transformers is the *scaled dot-product attention*, where information coming from different data representations and positions are encoded in a parallel way given by:

$$Attention(Q, K, V) = Softmax(QK^T/\sqrt{d})V, \tag{6}$$

where *Attention* is a mapping function using $Q, K, V \in \mathbb{R}^{N\times d}$ (a query, key, and value matrix, respectively). N is the number of sequence vectors, and d represents its dimension scaled for normalization.

These matrices are computed from P, by the linear transformations W_Q, W_K, and $W_V \in \mathbb{R}^{d\times d}$ as:

$$Q = PW_Q, K = PW_K, V = PW_V. \tag{7}$$

3.3.3. Transformer multiple self-attention heads

To encode attention, multiple h self-attention heads are concatenated together as:

$$MSA(Q, K, V) = Concat(H_1,\ H_2,\ \ldots,\ H_h)W_{out}, \tag{8}$$
$$H_i = Attention(Q_i, K_i, V_i),\ i \in [1,\ \ldots,\ h]. \tag{9}$$

The general structure of a transformer stack L identical layers given the embedded space $P \in \mathbb{R}^{f\times d}$. Each layer contains a multi-head attention block in conjunction with an MLP layer.

These blocks are placed in-between a Layer Norm $LN(\cdot)$ and a residual connection such that:

$$Y'_l = MSA(LN(Y_{l-1})) + Y_{l-1}, \tag{10}$$
$$Y_l = MLP(LN(Y_l)) + Y'_l, \tag{11}$$
$$Z = LN(Y_l), \tag{12}$$

where the transformer output $Z \in \mathbb{R}^{f\times d}$ has the same size of its input $P \in \mathbb{R}^{f\times d}$, and it is averaged in frame dimension to get a vector $\mathbf{z} \in \mathbb{R}^{1\times d}$.

3.3.4. Denoising property

The transformer-based decoder reconstructs each skeletal action sequence, starting from the unsupervised latent features Z, into $\hat{X}_{pert} \in \mathbb{R}^{f\times(3J)}$. The MSE reconstruction loss ensures the model correctly encodes and rebuilds each data sample free of any noise or corruptions injected during training:

$$\mathcal{L}_{\text{MSE}} = \tfrac{1}{2}\mathbb{E}_{X\sim\mathcal{B}}\big[\|X_{clean} - \hat{X}_{pert}\|^2_F\big], \tag{13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e., the Euclidean norm of the vector obtained after flattening the tensor. The MSE loss is minimized over mini-batches $\mathcal{B}$.

3.3.5. Rotation invariance

Granting the flexibility of the transformer-based approach to combine reconstruction loss with other complementary losses, this section introduces an additional loss to ensure learning consistencies w.r.t. rotation invariance. This is visualized in Figure 1.

First, each skeletal action sequence was altered by applying ROT (3D rotations, see Section 3.2) to obtain X_{rot}. Rotation labels were defined as y_x, y_y, and y_z, corresponding to the rotation angles applied to the *rotated* action sequence X_{rot}. These rotation labels are only used for the skeletal rotation prediction task, *but not for U-HAR* i.e., not used for the classification task.

During training, for each 3D axis, an additional patch token P_r and relative positional embeddings PE_r were stacked (concatenated) on top of the existing ones, thus obtaining:

$$P_{rot} = concat(P_r, P) + concat(PE_r, PE) \tag{14}$$

After the encoding stage, the first three vectors were selected from Z (the latent features extracted from P_r) and fed into three different linear layers, representing the axes' rotation heads. The overall goal is to classify the correct rotation angles (as predicted rotation labels $\hat{y}_x$, $\hat{y}_y$, and $\hat{y}_z$) using cross entropy losses, defined as:

$$\hat{y}_x = softmax(RotHead_x(Rot(x_{i\_clean}, \alpha))) \quad (15)$$

$$\mathcal{L}_{\text{rot\_x}}(\theta) = -\frac{1}{N}\sum_i^N log\ \hat{y}_x^\alpha \quad (16)$$

$$\hat{y}_y = softmax(RotHead_y(Rot(x_{i\_clean}, \beta))) \quad (17)$$

$$\mathcal{L}_{\text{rot\_y}}(\theta) = -\frac{1}{N}\sum_i^N log\ \hat{y}_y^\beta \quad (18)$$

$$\hat{y}_z = softmax(RotHead_z(Rot(x_{i\_clean}, \gamma))) \quad (19)$$

$$\mathcal{L}_{\text{rot\_z}}(\theta) = -\frac{1}{N}\sum_i^N log\ \hat{y}_z^\gamma, \quad (20)$$

where $Rot(\cdot,\cdot)$ is the rotation function (as shown in Equation 2), $RotHead(\cdot)$ is the output of each rotation heads, and θ denotes the encoder parameters. The final loss for this task is:

$$\mathcal{L} = \tfrac{1}{2}\mathbb{E}_{X\sim\mathcal{B}}\left[\|X_{rot} - \hat{X}_{rot}\|_F^2\right] + \mathcal{L}_{\text{rot\_x}} + \mathcal{L}_{\text{rot\_y}} + \mathcal{L}_{\text{rot\_z}}. \quad (21)$$

3.3.6. Temporal motion consistency with triplet loss

The motion information of a skeletal action sequence can be easily obtained from joints data as it can be represented as the temporal displacement of each joint (Li et al., 2021), i.e., $x_{t+1} - x_t$. Herein, the goal of this paper is to better regularize the model by checking consistencies between the reconstructed skeleton and its data byproduct (i.e., the motion data) using a Triplet Margin Loss (Balntas et al., 2016):

$$\mathcal{L}_{\text{contr}}(a, p, n) = max\{\|a_i - p_i\|_2 - \|a_i - n_i\|_2 + margin,\ 0\}, \quad (22)$$

where a is the anchor samples, joints data coming from X_{pert}, p represents the positive samples obtained from *forward* motion data (unaltered motion data), n is the negative samples consisting of *reversed* motion data and left the default value of 1 for *margin*. This ensures that the latent features learn to reconstruct action samples into the correct temporal motion despite the presence of altered data (RM, see Section 3.2) by attracting the positive samples of the correct motion and pushing afar the inverted motion data which can perturb the model performance. The final loss for this task is given by:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{contr}}. \quad (23)$$

4. Experimental analysis

Each skeletal action sequence is normalized in terms of bone length in the range of [−1, 1]. As for their temporal sequence length, every missing time-frames were discarded (applying methods introduced in Su et al., 2020a) and regularized the frame numbers to match a fixed size (fixing each sequence length up to 100 time-frames by applying a regularization in which frames of longer samples were cut or replicate frames for shorter samples). Both encoder and decoder modules are made of two transformer layers with four attention heads each. Patch embedding and latent space sizes are set to 256. The positional embedding length is set to 100, matching the temporal length of the given action sequences. The model is trained for 100 epochs using AdamW optimizer with a batch size of 64 and a learning rate of 0.001 (with a decay scheduling at epochs 20 and 70).

The experimental analysis was performed on two large-scale skeletal action datasets: NTU-60 (Shahroudy et al., 2016) and NTU-120 (Liu et al., 2019) using all available data splits, i.e., Cross-Subject, Cross-View, and Cross-Setup. For action inference, the common protocol of unsupervised feature learning was used, i.e., linear evaluation (Zheng et al., 2018), such that the latent features (learned without supervision) are given to a linear classifier to perform HAR. Notice that the inference stage is the same with all SOTA competitors. The performance of our method (SKELTER) was compared against 9 SOTA skeleton-based U-HAR methods: LongTGAN (Zheng et al., 2018), MS2L (Lin et al., 2020), P&C (Su et al., 2020a), PCRP (Xu et al., 2023), AS_CAL (Rao et al., 2021), AE-L (Paoletti et al., 2021b), CrosSCLR (Li et al., 2021), ISC (Thoker et al., 2021), and AimCLR (Guo et al., 2022). In Supplementary material, a comprehensive analysis of the inference time and space complexity for our model, as well as other methods, is presented.

4.1. Comparison with SOTA for data perturbation

By first verifying if the initial claim of this paper is valid (i.e., U-HAR methods are not resilient to data perturbations), all SOTA were evaluated by supplying their code publicly in two distinct evaluation phases:

- Investigate SOTA U-HAR and SKELTER's accuracy results and performance drop when data perturbation is applied *only* on the test set, where the SOTA models are pre-trained using the original and unaltered data. Table 1 reports quantitative results, whereas Figure 3 represents the graphical counterpart in terms of bar plots (lower the bars, better the results).

We provide in Supplementary material the extended tables for all experiments and complete bar plots for better readability.

- Investigate the accuracy results and performance drop of SOTA U-HAR and SKELTER when data perturbation is applied on *both* the train and test set, *de-facto* re-training from scratch all SOTA models providing perturbed data.

We provide in Supplementary material (due to space constraints) the extended tables for all experiments, along with complete bar plots for better readability.

Overall, the extensive quantitative and qualitative results confirm and demonstrate the sensible weakness in performance (i.e., classification accuracy) of these approaches w.r.t. such perturbations, showing that all the methods' performance decrease when the testing data is corrupted, in some cases up to 70%. However, it is important to notice that even for the cases in which

TABLE 1 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when *only* the *testing* splits of NTU-60 (Shahroudy et al., 2016) (top) and NTU-120 (Liu et al., 2019) (bottom) are perturbed by: SUB, AR, JR, SHR, GN, LR, and JO (see Section 3.2 for definitions).

	NTU-60 C-subject										NTU-60 C-view									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
Performance Accuracy (ACC %) – Perturbations on *Test set only*																				
LongTGAN	52.1	4.9	12.9	32.3	10.7	32.7	30.9	29.1	23.0	29.1	56.4	8.7	14.6	38.3	11.2	39.2	19.6	12.0	21.8	34.6
MS2L	52.6	16.6	15.2	21.2	15.7	19.8	20.7	34.0	23.9	28.7	46.4	10.1	15.9	11.0	14.4	22.2	12.2	30.7	20.5	25.9
P&C FS	50.6	5.9	18.1	37.9	14.4	39.2	35.9	34.9	27.9	22.7	76.3	8.5	23.5	60.8	19.7	63.1	50.7	29.5	38.9	37.4
P&C FW	50.7	18.3	14.2	41.7	13.2	42.4	35.2	35.2	29.9	20.8	76.1	5.9	15.8	59.1	12.5	61.2	39.2	29.1	34.4	41.7
PCRP	53.9	6.2	12.2	39.6	15.8	40.2	32.7	42.8	28.7	25.2	63.5	15.3	14.1	45.5	17.4	46.8	40.7	32.2	32.2	31.3
AS_CAL	58.5	39.7	36.8	46.1	37.9	46.5	41.3	38.9	40.5	18.0	64.6	37.7	33.9	46.7	34.7	46.1	35.4	40.2	39.1	25.5
AE-L	69.9	30.3	31.6	65.4	23.0	66.7	59.7	50.1	48.7	21.2	**85.4**	11.4	35.4	76.4	24.7	75.5	66.0	58.2	51.6	33.8
CrosSCLR	**77.8**	51.2	40.1	50.5	40.4	22.4	49.4	57.4	47.8	30.0	83.4	58.0	44.6	56.5	53.0	28.6	52.7	57.4	54.1	29.3
ISC	76.3	54.2	50.1	63.8	50.8	63.0	56.9	62.1	58.2	18.1	85.2	60.1	49.2	74.0	62.4	72.1	68.8	70.1	66.0	19.2
AimCLR	74.3	55.7	50.0	66.3	58.2	65.0	60.1	63.3	60.5	13.8	79.7	60.9	54.9	76.5	65.8	73.8	70.4	74.1	68.8	10.2
SKELTER	69.2	**57.2**	**60.0**	**69.0**	**63.7**	**67.9**	**63.9**	**68.9**	**64.4**	**4.8**	78.5	**62.1**	**66.4**	**77.5**	**70.5**	**76.8**	**71.9**	**77.5**	**71.8**	**6.7**
	NTU-60 C-subject										NTU-60 C-view									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
Performance Accuracy (ACC %) – Perturbations on *Test set only*																				
LongTGAN	35.6	4.8	7.3	26.7	5.2	26.7	20.8	18.8	17.7	17.9	39.7	3.9	8.7	27.2	6.4	28.1	17.4	12.6	16.5	23.2
MS2L	24.3	8.5	10.2	7.7	8.4	9.4	9.1	12.7	10.5	13.8	23.8	8.3	10.0	8.4	9.5	8.4	9.9	8.3	10.3	13.5
P&C FS	40.5	2.1	6.0	29.9	6.5	30.2	28.8	24.5	19.8	20.7	42.4	12.7	9.7	25.1	7.3	25.1	20.8	22.0	18.4	24.0
P&C FW	40.3	14.3	9.4	30.4	7.6	31.5	28.3	24.2	21,5	18.8	42.9	2.1	4.7	32.6	6.9	33.0	23.0	22.1	20.1	22.8
PCRP	41.7	3.7	6.5	27.8	8.5	28.2	22.4	22.8	19.5	22.2	45.1	13.5	8.7	30.7	9.7	31.2	27.1	20.7	21.7	23.4
AS_CAL	48.6	27.6	23.9	34.1	25.0	34.7	26.1	30.1	28.3	20.3	49.2	26.5	22.6	35.7	23.5	36.2	32.9	35.8	29.9	19.3
AE_L	59.1	7.6	19.3	47.3	11.7	51.3	40.9	47.2	34.9	24.2	62.4	7.7	22.6	48.1	12.9	42.8	40.3	32.6	32.9	29.5
CrosSCLR	67.9	40.7	26.4	40.8	35.9	13.5	39.2	47.0	37.7	30.2	66.7	41.7	29.0	42.1	36.1	18.0	43.0	50.1	40.8	25.9
ISC	67.1	44.0	37.2	50.8	44.4	52.8	49.3	50.3	47.5	19.6	67.9	40.5	34.8	50.8	38.4	43.9	48.1	53.2	46.4	21.5
AimCLR	**68.2**	44.9	42.0	53.2	46.9	54.9	50.1	55.9	50.2	18.0	**68.8**	41.1	37.0	57.1	41.1	44.2	50.9	54.4	48.4	20.4
SKELTER	52.9	**46.5**	**48.7**	**58.2**	**51.7**	**59.1**	**53.9**	**58.9**	**53.9**	**0.0**	56.0	**42.9**	**40.6**	**60.9**	**44.1**	**45.7**	**56.5**	**60.5**	**50.2**	**5.8**

The average (AVG) accuracy and the Drop, ↓ w.r.t. clean data (CLN), are given (the lower, the better). CLN stands for clean data, i.e., usage of original data as supplied by the datasets. The best results of each column are given in bold.

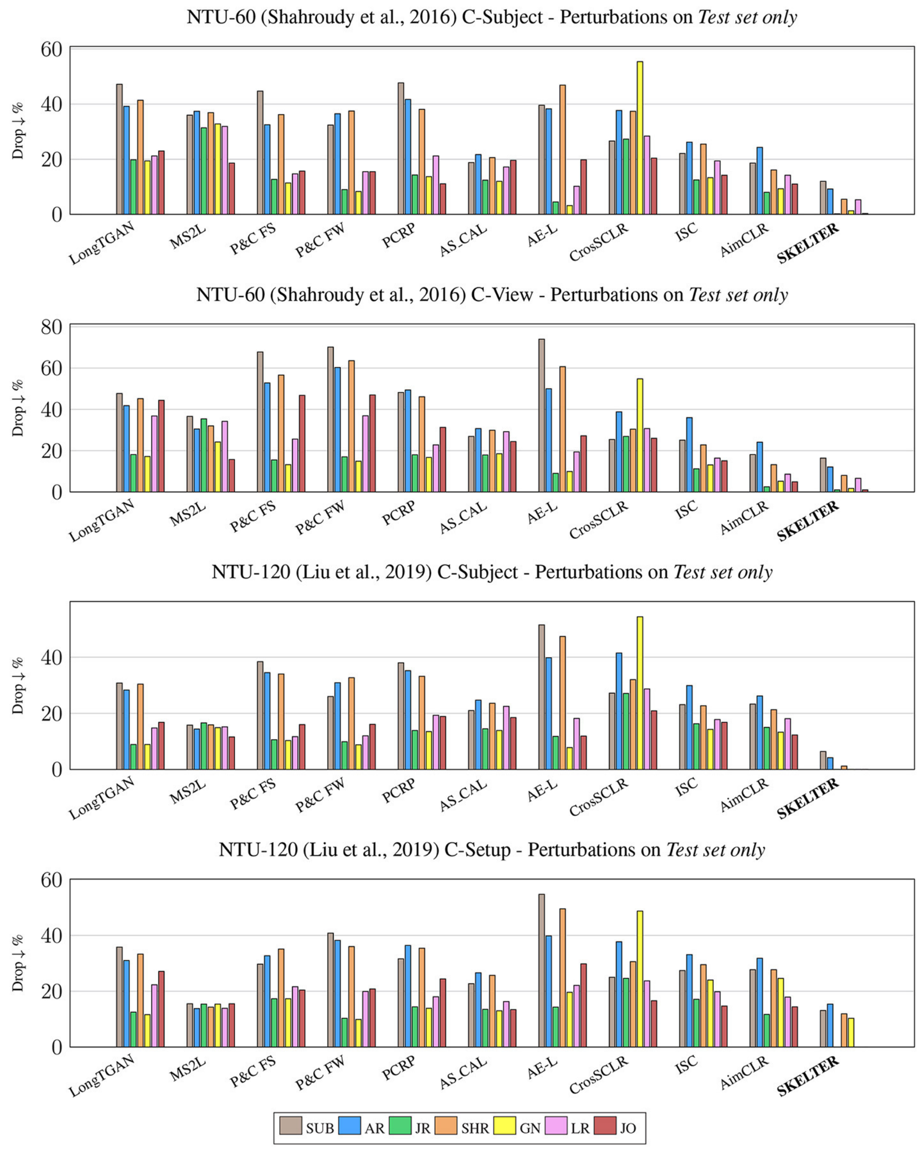

FIGURE 3
Performance drop ↓ % (related to the decrease of accuracy points) of SOTA U-HAR and SKELTER when *only* the *testing* splits of NTU-60 (Shahroudy et al., 2016) (top half) and NTU-120 (Liu et al., 2019) (bottom half) are perturbed by: SUB, AR, JR, SHR, GN, LR, and JO (see Section 3.2 for definitions). The lowest bars represent the best results.

the set of data perturbations are introduced to the models in their training, remarkable drops in the performance still exist, up to 45%.

The reader can observe that for the perturbed data, the accuracy of SKELTER is better than the others in all datasets: such strong drops are not observed for SKELTER, proving its better denoising capabilities compared to SOTA. In other words, the performance drop of SKELTER is lower than others, and its performance is more accurate than others with the perturbed data.

It is also important to highlight that some methods, such as AS-CAL (Rao et al., 2021), CrosSCLR, ISC, and AimCLR, all perform contrastive learning while they augment the data in terms of e.g., Shear, Gaussian noise, and Rotation. Therefore, they would be more resistant to the corresponding perturbations. However, compared to SKELTER, their performance decrease is relevant.

4.2. Comparison with SOTA for data alteration

This section reports the performance of SOTA U-HAR when rotation (ROT) and reversed motion (RM) are applied to the datasets, with the same experimental pipeline described in the previous section (Table 2 and Figure 4). These results also include SKELTER's performance in four settings to examine the importance of using our rotation-invariance and triplet losses:

- Pure SKELTER: using only the $\mathcal{L}_{\text{MSE}}$ loss (Equation 13).
- SKELTER with the rotation invariance loss (Equation 21).
- SKELTER with the triplet loss $\mathcal{L}_{\text{contr}}$ (Equation 22).

TABLE 2 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when only the *testing* splits of NTU-60 (Shahroudy et al., 2016) (top) and NTU-120 (Liu et al., 2019) (bottom) are altered by: ROT and RM (see Section 3.2 for definitions) in terms of the average (AVG) accuracy and the Drop ↓ w.r.t. clean data (CLN) (the lower, the better).

	NTU-60 C-subject					NTU-60 C-view					NTU-120 C-subject					NTU-120 C-setup				
	CLN	ROT	RM	AVG	Drop ↓	CLN	ROT	RM	AVG	Drop ↓	CLN	ROT	RM	AVG	Drop ↓	CLN	ROT	RM	AVG	Drop ↓
				ACC (%)	(%)				ACC (%)	(%)				ACC (%)	(%)				ACC (%)	(%)
Performance Accuracy (ACC %) – Alterations on *Test set* only																				
LongTGAN	52.1	23.4	30.1	26.7	25.4	56.4	32.0	20.4	26.2	30.2	35.6	18.2	30.9	24.5	11.1	39.7	24.2	20.0	22.1	17.6
MS2L	52.6	38.2	33.4	35.8	16.8	46.4	33.8	34.2	34.0	12.4	24.3	13.3	14.9	14.1	10.2	23.8	12.8	17.5	15.1	8.8
P&C FS	50.6	31.0	33.8	32.4	18.2	76.3	46.2	47.8	47.0	29.3	40.5	22.6	27.8	25.2	15.3	42.4	20.8	22.4	21.6	20.8
P&C FW	50.7	32.6	36.2	34.4	16.3	76.1	37.7	48.7	43.2	32.9	40.3	20.4	27.4	23.9	16.4	42.9	21.4	34.9	28.1	14.8
PCRP	53.9	29.9	38.8	34.3	19.6	63.5	37.5	40.0	38.7	24.8	41.7	24.9	30.5	27.7	14.0	45.1	23.3	30.5	26.9	18.2
AS_CAL	58.5	33.3	43.7	38.5	20.0	64.6	33.0	43.9	38.4	26.2	48.6	20.0	32.8	26.4	22.2	49.2	21.9	34.2	28.1	21.1
AE-L	69.9	57.0	54.1	55.5	14.4	**85.4**	58.8	58.2	58.5	26.6	59.1	42.4	46.2	44.3	14.8	62.4	40.7	48.8	44.7	17.7
CrosSCLR	**77.8**	60.1	58.8	59.4	18.4	83.4	66.9	68.8	67.8	15.6	67.9	45.9	50.1	48.0	19.9	66.7	52.8	54.2	53.5	13.2
ISC	76.3	60.3	62.9	61.6	14.7	85.2	67.2	70.1	68.6	16.6	67.1	50.7	48.1	49.4	17.7	67.9	53.0	54.7	53.8	14.1
AimCLR	74.3	62.7	63.4	63.1	11.2	79.7	69.4	73.0	71.2	8.5	**68.2**	51.2	52.9	52.1	16.1	**68.8**	54.1	56.0	55.1	13.7
SKELTER (Pure)	69.2	63.8	65.2	64.5	4.7	78.5	70.1	76.1	73.1	5.4	52.9	54.1	54.6	54.3	0.0	56.0	55.3	57.7	56.5	0.0
SKELTER (*w/ RotHeads*)	69.2	**66.2**	61.1	63.6	5.6	78.5	**75.2**	73.4	74.3	4.2	52.9	**56.6**	53.8	55.2	0.0	56.0	**58.8**	56.1	57.4	0.0
SKELTER (*w/* $\mathcal{L}_{\text{contr}}$)	69.2	62.0	**68.7**	65.3	3.9	78.5	69.8	**78.0**	73.9	2.8	52.9	53.0	**59.0**	56.0	0.0	56.0	54.9	**61.0**	57.9	0.0
SKELTER (*w/ RotHeads +* $\mathcal{L}_{\text{contr}}$)	69.2	62.7	64.1	63.4	5.8	78.5	68.2	75.9	72.1	6.4	52.9	53.8	54.0	53.9	0.0	56.0	55.2	56.9	56.1	0.0

CLN stands for clean data, i.e., usage of original data as supplied by the datasets. The results of SKELTER are given in three settings: (a) pure SKELTER, (b) SKELTER with the rotation head (RotHeads), (c) SKELTER with $\mathcal{L}_{\text{contr}}$ and (d) SKELTER with the rotation head (RotHeads) and $\mathcal{L}_{\text{contr}}$. The best results of each column are given in bold while the second best result is underlined.

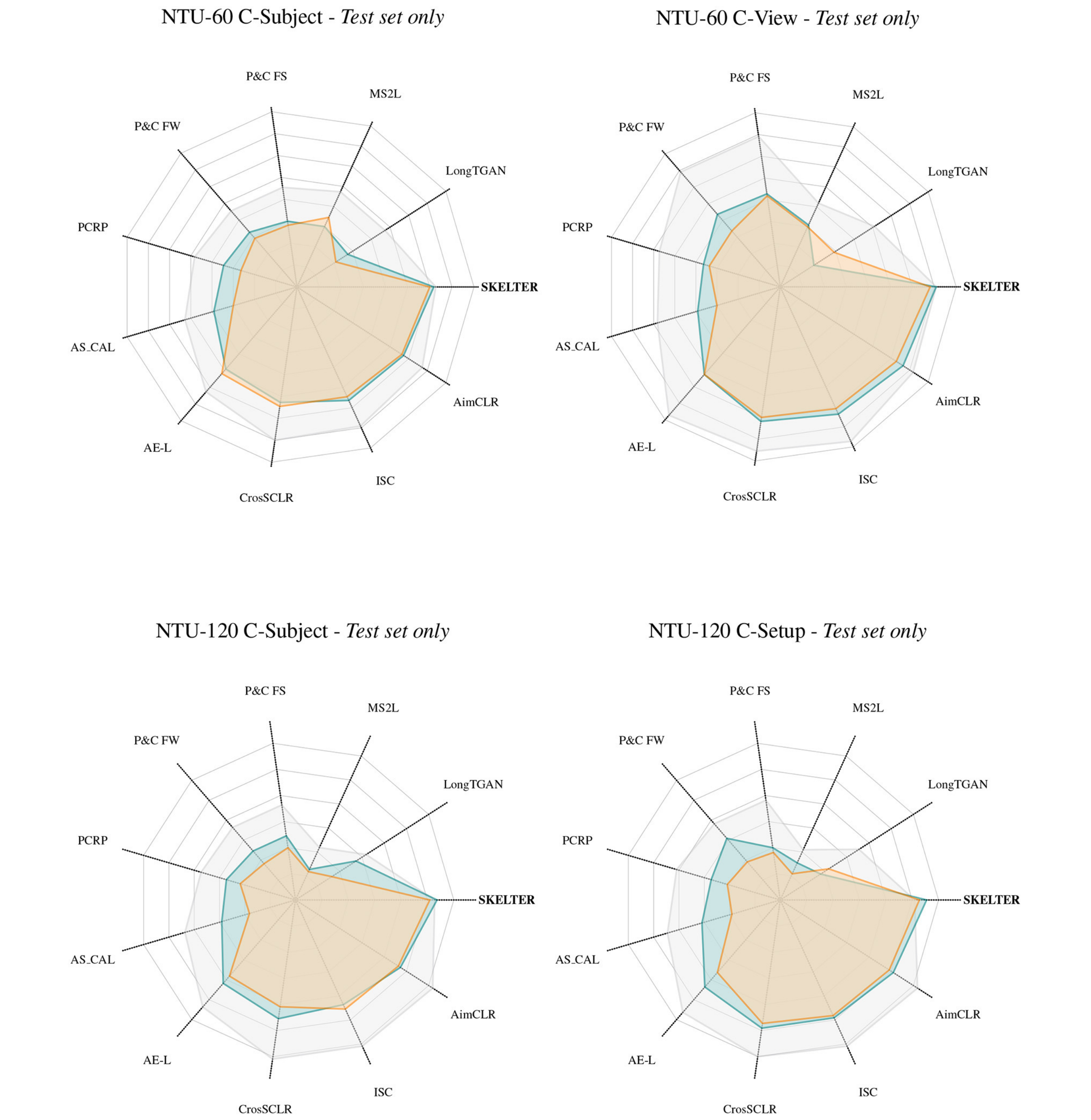

FIGURE 4
Kiviat plots in terms of Accuracy (%) between the SOTA U-HAR and SKELTER when only the *testing* splits of NTU-60 (Shahroudy et al., 2016) and NTU-120 (Liu et al., 2019) are altered by: ROT and RM (see Section 3.2 for definitions). Each ray line represents the accuracy results of each method (where the center is the zero), and colored lines and areas represent the Accuracy values w.r.t. the CLN (gray), ROT (blue), and RM (orange) applied. CLN stands for clean data, i.e., usage of original data as supplied by the datasets.

- SKELTER with the rotation invariance loss (Equation 21) and the triplet loss $\mathcal{L}_{\text{contr}}$ (Equation 22).

The reader can observe the same trends in the previous section, such that when Rotation and Reversed Motion are applied, the performance of SKELTER drops less than SOTA methods while performing better than all SOTA in terms of accuracy. Additionally, the proposed *rotation invariance* head and the inclusion of *triplet loss for temporal motion consistency* always improve the performance, achieving the best out of all.

5. Qualitative results

Figure 5 shows the visualizations of a skeletal action sequence “Throw” picked from the NTU-60 (Shahroudy et al., 2016) Cross-View split, demonstrating how SKELTER reconstructs and denoise

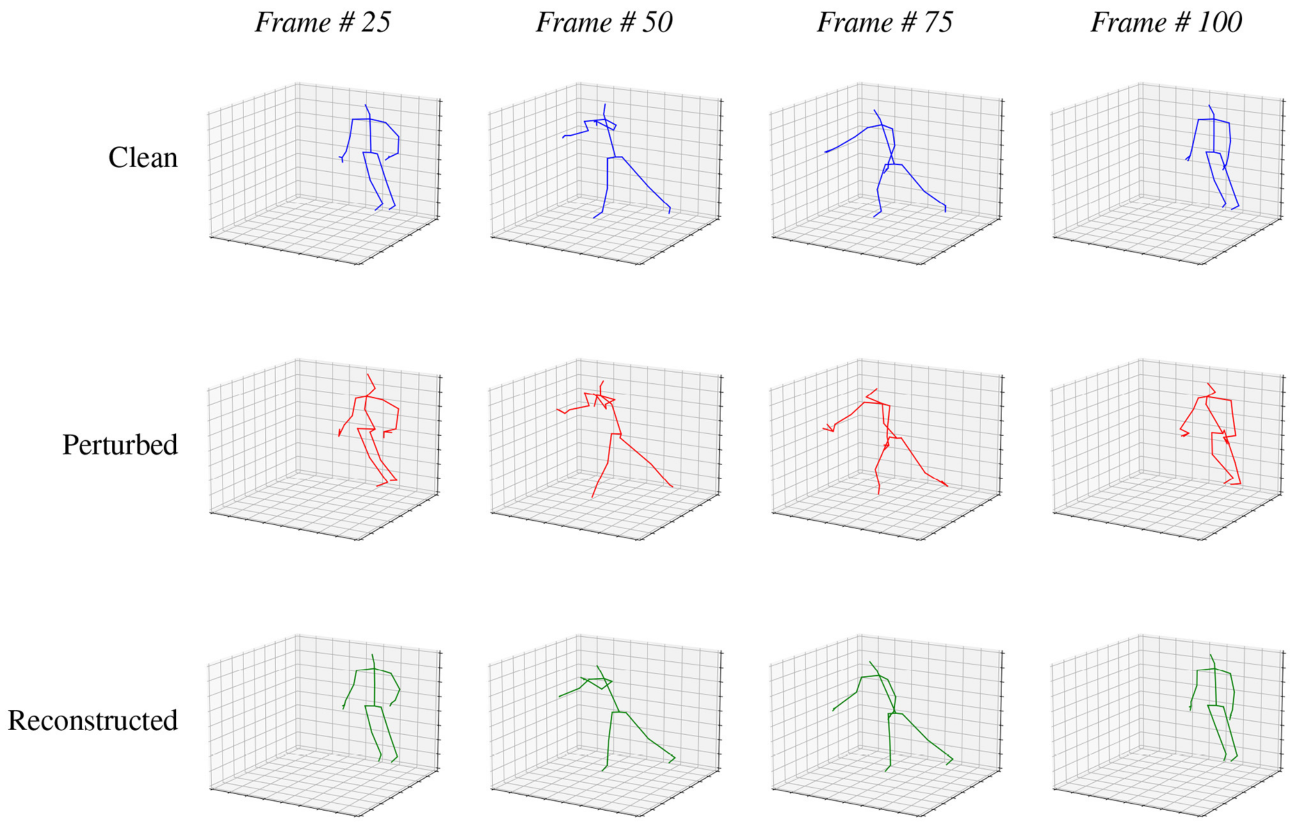

FIGURE 5
Original (blue), perturbed (red), and SKELTER-reconstructed (green) skeletal pose. As the data perturbation *Gaussian additive noise* is applied, each column represents one particular frame of the overall sequence. **Left** to **right:** frame #25, frame #50, frame #75, frame #100.

each sample accordingly as the sequence unfolds w.r.t. its temporal dimension. As a reference, the original unaltered skeletons are represented in *blue* color, their perturbed counterpart (by applying one of the perturbations from Section 3.2) in *red* color, and the denoised skeletons obtained from our SKELTER model in *green* color. In addition, we report in Supplementary material some of the proposed perturbations, which potentially could negatively affect our method's performances and the state-of-the-art: starting on a variety of perturbed data, the effectiveness of SKELTER can be seen through *the smoothed denoised skeleton reconstruction* of it even in case of heavy data perturbation.

5.1. Real-life scenario—A case study

Section 3.1 sets the foundations of the overall claim of this paper: devise an unsupervised model, U-HAR oriented, capable of handling data corruption of skeleton poses in any conditions which can be found in more practical scenarios. Subsequent sections proved the usefulness of SKELTER for this particular task. Still, an important question remained unanswered: if the proposed skeleton poses perturbations or alterations (described in Section 3.2) plausibly reflect data corruption that could be found in real-world scenarios. This section describes a case study about a simulated scenario, comparing a perturbed dataset (i.e., perturbed NTU-60 Shahroudy et al., 2016) and real-world 2D-skeleton poses. The goal is to demonstrate that both data distributions can overlap, confirming the plausibility of proposed perturbation w.r.t. real data.

A set of 2D skeleton poses were captured from a CCTV video stream using OpenPose (Cao et al., 2017) to achieve this. Recordings were made in an office scenario, where the original video stream was deleted later to maintain the privacy of people detected. This can be seen in Figure 6, where a clean office background is left only for visualization purposes: the *left* pose represents a sample frame from the real-world poses captured, and the *right* pose represents a sample frame from the perturbed dataset. In addition, camera parameters and a reference origin point were recorded and estimated to ensure an equal comparison for both data distributions. As for the *perturbed dataset*, a world-to-camera projection had to be performed to convert its 3D poses into 2D poses, compatible in terms of the number of joints (keeping only a subset of 17 skeleton joints common to each other), their order and their pixel position w.r.t. camera parameters estimated beforehand. The reference origin point was necessary to align all poses from both datasets. In addition, for the perturbed dataset, to add variety and add realism, each pose was rotated along its Z-axis before performing the camera projection to ensure a similar behavior naturally occurring in real-life scenarios (i.e., rotations of

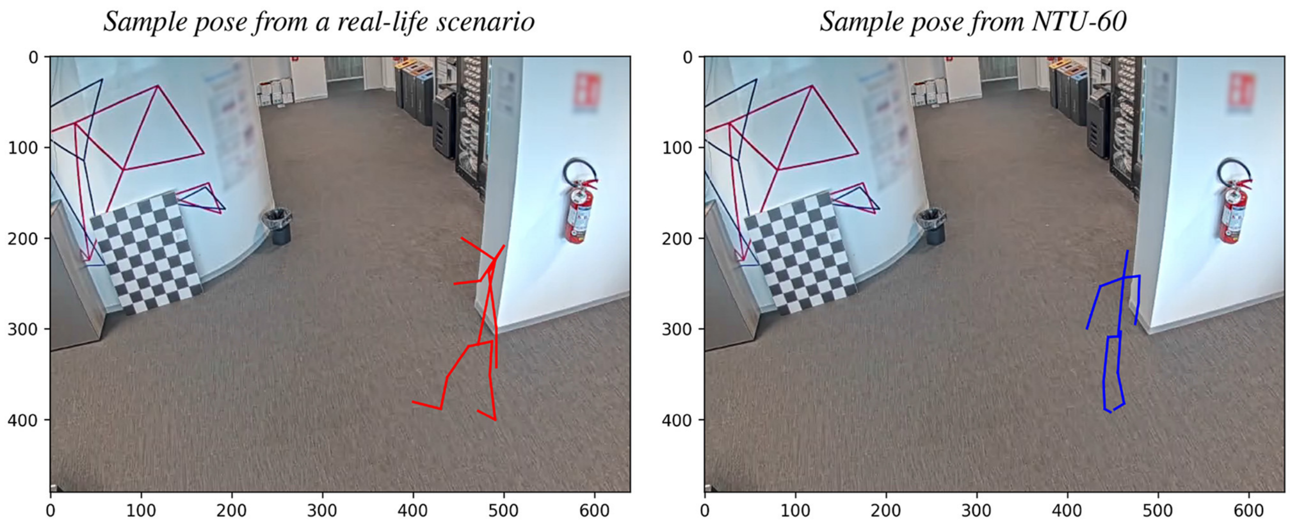

FIGURE 6
Graphical comparison between perturbed and real-life sample poses. **Left:** 2D skeleton pose estimated using OpenPose (Cao et al., 2017), from a sample captured from a CCTV video stream (red skeleton). Camera calibration and reference origin point estimated beforehand for the 3D-to-2D conversion of the perturbed dataset. All 2D poses were normalized and centered w.r.t. the reference point, which is set identically to the perturbed poses. **Right:** a sample from NTU-60 (Shahroudy et al., 2016) (blue skeleton) after applying the world-to-camera projection, using camera parameters obtained earlier, making sure that both distributions of poses are compatible with each other. Axis values correspond to the pixel values of the recorded frame (i.e., 640 × 480). In both cases, the RGB background is left for illustration purposes.

TABLE 3 Statistics between perturbed NTU-60 (Shahroudy et al., 2016) and real-world 2D poses.

		Missing joints (AVG %)	Missing limbs (AVG %)	MMD ($p < 0.05$)
Perturbed NTU-60	CLN	0.32	0.49	0.0095
	SUB	5.89	0.01	0.0078
	AR	0.21	0.30	0.0313
	JR	2.11	0.57	0.0157
	SHR	0.89	1.32	0.0191
	GN	0.24	0.45	0.0294
	LR	20.87	25.38	0.0009
	JO	0.64	0.49	0.0103
	Real-world	13.45	22.76	-

Values of missing joints and limbs are reported as the average percentage w.r.t. all joints of 2D poses. MMD refers to the Maximum Mean Discrepancy (Tolstikhin et al., 2016) between the real-world 2D poses and each distinct proposed perturbation of NTU-60 (Shahroudy et al., 2016).

people detected). As the last step, pose normalization in unit-norm was applied for both datasets.

Table 3 reports some statistics related to the number of missing joints, missing limbs (i.e., a group of joints), and the *Maximum Mean Discrepancy* (MMD). Missing joints and limbs refer to the averaged percentage value of each distinct joint that is missing (i.e., zero-valued) for the former and the missing values of groups of joints that form one of the four limbs (i.e., arms and legs). Results show that the Limbs Removal perturbation is the closest w.r.t. real-world 2D poses, simulating the high occurrence of missing entire body parts due to heavy occlusions instead of milder occlusions like single Joints Removal. Maximum mean discrepancy (MMD) (Tolstikhin et al., 2016) is a kernel-based statistical test used to determine whether two given data distributions are identical. In addition to being used as a statistical test (as an integral probability metric), MMD can also be used as a loss or cost function in various machine learning algorithms (as a distance, or difference, between feature means). It is often used as a simpler discriminator because of its easy implementation and the rich kernel-based theory that underlies its principles. The kernel trick was used to estimate this measure, and a lower value denotes a statistically-significative overlap between the two data distributions. It was performed by comparing the real-world 2D poses with each and distinct NTU-60 (Shahroudy et al., 2016) dataset perturbation proposed in Section 3.2. In all cases, its value was below the null hypothesis $p < 0.05$, denoting the plausibility of such proposed data perturbation strategies, despite the semantic differences and type of motion involved.

6. Conclusions

Robust human action recognition is a fundamental capability in artificial intelligence systems, and it becomes rather crucial to assess the resilience of a human action recognition system before its implementation in practical contexts, especially in biomedical applications where robustness and denoising capabilities are imperative. In this paper, we have shown that data perturbations and alterations can severely reduce the performance of existing approaches. We first introduced several perturbations and alterations commonly found when extracting skeletal data in realistic environments (e.g., occlusions, geometrical distortions, noise, etc.). Then, we presented a novel framework based on a transformer encoder-decoder, accepting 3D-skeletal data as the input. We also presented additional losses to have robust representations against rotation variances and provide consistent temporal motion. Indeed, we showed that the current methods have a relevant drop in performance while our approach is less affected by such data perturbations and alterations. This confirms that our approach might be prone to better resistance to challenging realistic operational scenarios.

Data availability statement

The publicly available datasets presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

GP, CB, and AD contributed to the conception and design of the study. GP performed the experimental analysis and wrote the first draft of the manuscript. CB and AD edited the sections of the manuscript. All authors contributed to the manuscript discussions and revisions as well as approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1203901/full#supplementary-material

References

Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. (2016). "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *BMVC*, 3.

Ben Tanfous, A., Drira, H., and Ben Amor, B. (2018). "Coding kendall's shape trajectories for 3D action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2840–2849.

Bertasius, G., Wang, H., and Torresani, L. (2021). "Is space-time attention all you need for video understanding?" in *ICML*, 4.

Beyan, C., Karumuri, S., and Volpe, G. (2021). Modeling multiple temporal scales of full-body movements for emotion classification. *IEEE Trans. Affect. Comput.* 14, 1070–1081. doi: 10.1109/TAFFC.2021.3095425

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33*, 1877–1901.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). "Realtime multi-person 2D pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299.

Cebanov, E., Dobre, C., Gradinaru, A., Ciobanu, R.-I., and Stanciu, V.-D. (2019). "Activity recognitions for ambient assisted living using off-the-shelf motion sensing input devices," in *2019 Global IoT Summit (GIoTS)* (IEEE), 1–6.

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

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). "An image is worth 16x16 words: transformers for image recognition at scale," in *ICLR*.

Du, Y., Fu, Y., and Wang, L. (2015a). "Skeleton based action recognition with convolutional neural network," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)* (IEEE), 579–583.

Du, Y., Wang, W., and Wang, L. (2015b). "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of IEEE CVPR*, 1110–1118.

Evangelidis, G., Singh, G., and Horaud, R. (2014). "Skeletal quads: human action recognition using joint quadruples," in *2014 22nd International Conference on Pattern Recognition* (IEEE), 4513–4518.

Girdhar, R., Carreira, J., Doersch, C., and Zisserman, A. (2019). "Video action transformer network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 244–253.

Gui, L.-Y., Wang, Y.-X., Liang, X., and Moura, J. M. (2018). "Adversarial geometry-aware human motion prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 786–803.

Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T., and Ding, R. (2022). "Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 762–770.

Hendrycks, D., and Dietterich, T. (2018). "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations*.

Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. (2017). "A new representation of skeleton sequences for 3D action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3288–3297.

Kundu, J. N., Gor, M., Uppala, P. K., and Radhakrishnan, V. B. (2019). "Unsupervised feature learning of human actions as trajectories in pose embedding manifold," in *2019 IEEE Winter Conference on Applications of omputer Vision (WACV)* (IEEE), 1459–1467.

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

Li, J., Zhao, Q., Wong, Y., and Kankanhalli, M. S. (2018). "Unsupervised learning of view-invariant action representations," in *32nd Conference on Neural Information Processing Systems, Vol. 2018* (Montreal, QC), 1254–1264.

Li, L., Wang, M., Ni, B., Wang, H., Yang, J., and Zhang, W. (2021). "3D human action representation learning via cross-view consistency pursuit," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4741–4750.

Liang, D., Fan, G., Lin, G., Chen, W., Pan, X., and Zhu, H. (2019). "Three-stream convolutional neural network with multi-task and ensemble learning for 3D action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.*

Lin, L., Song, S., Yang, W., and Liu, J. (2020). "MS2L: multi-task self-supervised learning for skeleton based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2490–2498.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. (2019). NTU RGB+ D 120: a large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2684–2701. doi: 10.1109/TPAMI.2019.2916873

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). "ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems.* p. 13–23.

Martinez, J., Black, M. J., and Romero, J. (2017). "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2891–2900.

Ni, B., Wang, G., and Moulin, P. (2011). "RGBD-HuDaAct: a color-depth video database for human daily activity recognition," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (IEEE), 1147–1153.

Nie, Q., Liu, Z., and Liu, Y. (2020). "Unsupervised 3D human pose representation with viewpoint and pose disentanglement," in *Computer Vision–ECCV 2020: 16th European Conference* (Glasgow: Springer), 102–118.

Ohn-Bar, E., and Trivedi, M. (2013). "Joint angles similarities and hog2 for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 465–470.

Oreifej, O., and Liu, Z. (2013). "HON4D: histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of IEEE CVPR*, 716–723.

Paoletti, G., Beyan, C., and Del Bue, A. (2022). Graph laplacian-improved convolutional residual autoencoder for unsupervised human action and emotion recognition. *IEEE Access* 10, 131128–131143. doi: 10.1109/ACCESS.2022.3229478

Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2021a). "Subspace clustering for action recognition with covariance representations and temporal pruning," in *2020 25th International Conference on Pattern Recognition (ICPR)* (IEEE), 6035–6042.

Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2021b). "Unsupervised human action recognition with skeletal graph laplacian and self-supervised viewpoints invariance," in *32nd British Machine Vision Conference 2021, BMVC 2021.*

Parvin, P., Chessa, S., Manca, M., and Paterno', F. (2018). "Real-time anomaly detection in elderly behavior with the support of task models," in *Proceedings of the ACM on Human-Computer Interaction*, 1–18.

Pavllo, D., Feichtenhofer, C., Grangier, D., and Auli, M. (2019). "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7753–7762.

Petrovich, M., Black, M. J., and Varol, G. (2021). "Action-conditioned 3D human motion synthesis with transformer VAE," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10985–10995.

Petrushin, A., Freddolini, M., Barresi, G., Bustreo, M., Laffranchi, M., Del Bue, A., et al. (2022). "IoT-powered monitoring systems for geriatric healthcare: overview," in *Internet of Things for Human-Centered Design: Application to Elderly Healthcare*, 99–122.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vis. Comput.* 28, 976–990. doi: 10.1016/j.imavis.2009.11.014

Presti, L. L., and La Cascia, M. (2016). 3D skeleton-based human action classification: a survey. *Pattern Recogn.* 53, 130–147. doi: 10.1016/j.patcog.2015.11.019

Rao, H., Xu, S., Hu, X., Cheng, J., and Hu, B. (2021). Augmented skeleton based contrastive action learning with momentum LSTM for unsupervised action recognition. *Inform. Sci.* 569, 90–109. doi: 10.1016/j.ins.2021.04.023

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. (2021). "ImageNet-21k pretraining for the masses," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1).*

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). "NTU RGB+ D: a large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1010–1019.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019a). "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7912–7921.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019b). "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of IEEE CVPR*, 12026–12035.

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

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2018). Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. *IEEE Trans. Image Process.* 27, 3459–3471. doi: 10.1109/TIP.2018.2818328

Su, K., Liu, X., and Shlizerman, E. (2020a). "Predict & cluster: unsupervised skeleton based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9631–9640.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2020b). "Vl-BERT: pre-training of generic visual-linguistic representations," in *International Conference on Learning Representations.*

Thoker, F. M., Doughty, H., and Snoek, C. G. (2021). "Skeleton-contrastive 3D action representation learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 1655–1663.

Tolstikhin, I., Sriperumbudur, B. K., and Schölkopf, B. (2016). "Minimax estimation of maximum mean discrepancy with radial kernels," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*. p. 1938–1946.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems.* p. 6000–6010.

Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). "Human action recognition by representing 3D skeletons as points in a lie group," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 588–595.

Vemulapalli, R., and Chellapa, R. (2016). "Rolling rotations for recognizing human actions from 3D skeletal data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4471–4479.

Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012). "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 1290–1297.

Wang, L., Huynh, D. Q., and Koniusz, P. (2019a). A comparative review of recent kinect-based action recognition algorithms. *IEEE Trans. Image Process.* 29, 15–28. doi: 10.1109/TIP.2019.2925285

Wang, M., Ni, B., and Yang, X. (2020). Learning multi-view interactional skeleton graph for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 6940–6954. doi: 10.1109/TPAMI.2020.3032738

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., et al. (2019b). "Learning deep transformer models for machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1810–1822.

Xing, Y., and Zhu, J. (2021). Deep learning-based action recognition with 3D skeleton: a survey. *CAAI Trans. Intell. Technol.* 6, 80–92. doi: 10.1049/cit2.12014

Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., and Zhang, W. (2020a). "Deep kinematics analysis for monocular 3D human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 899–908.

Xu, S., Rao, H., Hu, X., Cheng, J., and Hu, B. (2023). "Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition," in *IEEE Transactions on Multimedia, Vol. 25* (IEEE), 624–634. doi: 10.1109/TMM.2021.3129616

Xu, S., Rao, H., Peng, H., Jiang, X., Guo, Y., Hu, X., and Hu, B. (2020b). Attention-based multilevel co-occurrence graph convolutional LSTM for 3-D action recognition. *IEEE Internet Things J.* 8, 15990–16001. doi: 10.1109/JIOT.2020.3042986

Yan, S., Xiong, Y., and Lin, D. (2018). "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence.*

Yang, X., and Tian, Y. (2014). "Super normal vector for activity recognition using depth sequences," in *Proceedings of the IEEE Conference on omputer Vision and Pattern Recognition*, 804–811.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). "XLNet: generalized autoregressive pretraining for language understanding," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. p. 5753–5763.

Yi, C., Yang, S., Li, H., Tan, Y.-p., and Kot, A. (2021). "Benchmarking the robustness of spatial-temporal models against corruptions," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).*

Zanfir, M., Leordeanu, M., and Sminchisescu, C. (2013). "The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proceedings of IEEE ICCV*, 2752–2759.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2017). "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proceedings of the IEEE international conference on computer vision*, 2117–2126.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2019). View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 1963–1978.

Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., and Zheng, N. (2020a). "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1112–1121.

Zhang, X., Xu, C., and Tao, D. (2020b). "Context aware graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14333–14342.

Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., and Ding, Z. (2021). "3D human pose estimation with spatial and temporal transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11656–11665.

Zheng, N., Wen, J., Liu, R., Long, L., Dai, J., and Gong, Z. (2018). "Unsupervised representation learning with long-term dynamics for skeleton based action recognition," in *AAAI Conference on Artificial Intelligence*.