Unsupervised 3D Skeleton-Based Action Recognition using Cross-Attention with Conditioned Generation Capabilities

David J. Lerch [1*], Zeyun Zhong [1,2*], Manuel Martin [1], Michael Voit [1], Jürgen Beyerer [1,2]
[1] Fraunhofer IOSB firstname.lastname@iosb.fraunhofer.de
[2] Karlsruhe Institute of Technology firstname.lastname@kit.edu

Abstract

Human action recognition plays a pivotal role in various real-world applications, including surveillance systems, robotics, and occupant monitoring in the car interior. With such a diverse range of domains, the demand for generalization becomes increasingly crucial. In this work, we propose a cross-attention-based encoder-decoder approach for unsupervised 3D skeleton-based action recognition. Specifically, our model takes a skeleton sequence as input for the encoder and further applies masking and noise to the original sequence for the decoder. By training the model to reconstruct the original skeleton sequence, it simultaneously learns to capture the underlying patterns of actions. Extensive experiments on NTU and NW-UCLA datasets demonstrate the state-of-the-art performance as well as the impressive generalizability of our proposed approach. Moreover, our experiments reveal that our approach is capable of generating conditioned skeleton sequences, offering the potential to enhance small datasets or generate samples of under-represented classes in imbalanced datasets. Our code will be published on GitHub.

1. Introduction

Human action recognition has emerged as a pivotal technology in various fields such as intelligent transportation systems or human-robot interaction, enabling advanced capabilities in safety enhancement, traffic management and human-machine interaction. Given the wide range of domains in human action recognition, including surveillance systems, robotics, occupant monitoring in the car interior, and pedestrian action recognition, the demand for generalization becomes crucial. Unsupervised action recognition methods provide a viable approach by utilizing unlabeled data to learn representations that generalize well across diverse scenarios, capturing the underlying structures and patterns of actions.

Skeleton-based action recognition techniques have gained popularity in recent years, as image-based approaches face challenges with varying imaging conditions, such as different camera viewpoints, lighting conditions and clothing [2, 26, 29]. Skeleton-based action recognition requires significantly less computational resources compared to image-based approaches [23], and generalizes well across complex environments [14]. Additionally, image-based methods may compromise privacy as they rely on storing and analyzing RGB images of individuals.

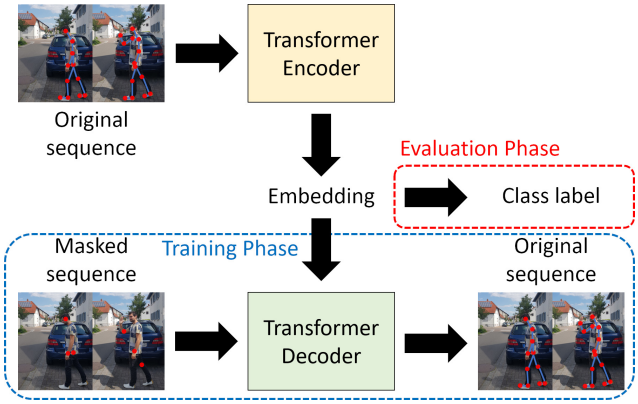

Figure 1. Principle of the transformer autoencoder for human action recognition. The original sequence is encoded by the transformer encoder. During training, the embedding is used for cross-attention in the transformer decoder. During evaluation, we use the embedding generated by the frozen encoder to generate the class label for the action recognition downstream task.

Recently, masking-based approaches have shown great success in image domain. Notably, the Masked AutoEncoder (MAE) [10] has effectively employed an encoder-decoder transformer architecture to reconstruct original images, even when extensive image patches are masked. This architecture is successfully employed in masked autoencoders for 3D mesh data aswell [12]. SkeletonMAE [32] pioneered the adaptation of the MAE concept to skeleton data, incorporating both frame-level and joint-level masks. However, unlike images, skeletons provide a more concise

representation of humans. Consequently, the process of masking out a larger proportion of frames and joints may introduce an overly challenging task, potentially hindering the model's learning process.

In this work, we propose a novel cross-attention-based autoencoder (see Figure 1) to address the aforementioned challenges in unsupervised skeleton-based action recognition. Our encoder takes a skeleton sequence as input and employs a learnable `[CLS]` token to summarize the input. Our decoder takes a masked (and noisy) version of the original skeleton sequence as input. By extracting knowledge from the `[CLS]` token output, the decoder becomes better equipped to comprehend and reconstruct the original skeleton sequence. This reconstruction, in turn, contributes to enhancing the encoder's learning process. We expect our model to perform well on actions which can be distinguished only by the skeleton sequence.

Our evaluation on the three popular action recognition datasets NTU 60 [19], NTU 120 [13], and NW-UCLA [28] demonstrates state-of-the-art and comparable performance as well as impressive generalizability. Furthermore, we investigate the capability of the proposed method for conditioned data generation, showing great potential which can be used to create conditioned synthetic data to diversify small-sized or rebalance imbalanced skeleton-based datasets.

In summary, the main contributions of our work are:

1) A novel cross-attention-based autoencoder for unsupervised 3D skeleton-based human action recognition, which demonstrates state-of-the-art results on NTU 60, NTU 120 and NW-UCLA datasets.

2) A cross-dataset evaluation showing the impressive generalizability of the proposed approach as well as a comparison to MAE-based approaches with our implementation.

3) Capability of conditioned data generation which can be used to create data conditioned on action labels to diversify small-sized datasets or to generate samples of underrepresented classes in imbalanced skeleton-based datasets.

2. Related Work

2.1. Supervised Skeleton-based Human Action Recognition

Skeleton-based action recognition techniques have gained popularity in recent years due to the limitations of image-based approaches, i.e., vulnerability to varying imaging conditions and privacy issues. Deep learning architectures have been widely utilized for skeleton-based action recognition, including Convolutional Neural Networks (CNN) [7, 36], RNN [21, 27], GCN [4, 20], and Transformers [5, 11, 17, 32]. CNNs showed effectiveness in extracting spatial features from skeleton data, while RNNs have been utilized to process the temporal aspects of the sequences. GCNs are used to model the topological graph features inherent in the skeleton data, capturing the spatial relationships between joints [35]. Transformers, which have shown promise in various sequential data analysis tasks, were also employed for skeleton-based action recognition, allowing for global representation learning.

2.2. Unsupervised Skeleton-based Action Recognition

Traditional unsupervised skeleton-based action recognition methods [3, 18, 22, 38] rely on Recurrent Neural Networks (RNN) architectures, including Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). While these methods showed promising results, the recurrent models are known to struggle with modeling long-range temporal dependencies due to their sequential (non-parallel) nature [6, 39]. In the realm of unsupervised skeleton-based action recognition, several notable approaches have emerged, each contributing unique methodologies to this challenging task.

Zheng et al. [38] introduced Long-Term Generative Adverserial Network (LongT-GAN), an encoder-decoder architecture based on GRUs. LongT-GAN focused on learning how to represent skeletal body poses over time. In addition to the main task of action recognition they introcuded an auxiliary inpainting task supported by an adversarial loss. This auxiliary task aided the learning process, improving the overall performance of the model. Xu et al. [33] published Prototypical Contrast and Reverse Prediction (PCRP) which went beyond the vanilla autoencoder for skeleton-based action recognition by incorporating an ad-hoc training mechanism based on expectation-maximization with learnable class prototypes. This mechanism enhanced the performance and robustness of the autoencoder by leveraging class-specific information during training. Su et al. [22] presented the Predict & Cluster (P&C) method, which utilized a Bi-directional-GRU (Bi-GRU) encoder and an Uni-directional-GRU decoder architecture. In addition to the unsupervised learning objective, action classification was performed using a 1-Nearest Neighbor (1-NN) predictor. Rao et al. [18] combined contrastive learning with a momentum LSTM. This Augmented Skeleton-based Contrastive Action Learning (AS-CAL) approach captured meaningful representations of skeletal data to encode long-term actions, enabling the model to capture temporal dependencies effectively. Paoletti et al. [15] presented a convolutional residual autoencoder (CR-AE) for unsupervised feature learning from 3D skeletal data. The Laplacian regularizer utilized graph Laplacian to enforce alignment between scalar com-

ponents in the feature space and improve feature representation learning by incorporating the knowledge of skeletal geometry. Chen et al. [3] introduced a Bi-GRU with attention mechanism in the encoder stage to better capture the long-term dependence of the input skeleton sequence. The decoder is trained using fixed weights and fixed states strategies to improve the clustering performance of the encoder, and the k-nearest neighbors algorithm (KNN) is utilized for action classification without the need for extra weight learning.

All aforementioned approaches use RNNs and different regularization strategies to support modelling long-range temporal dependencies. Our autoencoder is capable of modelling long-range temporal dependencies due to the transformer-based architecture.

In recent years non-generative transformer-based approaches for skeleton-based action recognition [11, 30] outperformed RNN based approaches. These approaches mainly focus on fine-tuning the model on the data after pretraining. In our work we evaluate the pretrained model without any fine-tuning. In contrast to the aforementioned approaches, our generative model is able to generate conditioned data.

In 2017, Vaswani et al. [25] published the standard transformer architecture consisting of an encoder and a decoder. The encoder of their model maps input symbols to continuous representations using self-attention and feed-forward layers, while the decoder generates an output sequence autoregressively. Attention mechanisms, such as scaled dot-product attention, are used to compute weighted sums of values based on queries and keys. Positional encodings are added to the input embeddings to provide information about token positions in the sequence. He et al. [10] proposed MAE for image classification tasks. The encoder receives unmasked patches and encodes them into hidden representations. By taking these representations, the decoder reconstructs the original image. This architecture is optimized by calculating a reconstruction loss on masked patches only. SkeletonMAE [32] pioneered the adaptation of the MAE concept to skeleton data, incorporating both frame-level and joint-level masks. Unlike MAE-based approaches, our approach reconstructs the original skeleton sequence by taking the knowledge of the complete sequence, which is encoded in the `[CLS]` embedding by the encoder. Different from SkeletonMAE [32], which incorporates STTFormer [17] architecture with MAE [10], our approach uses vanilla transformer architecture [25] with cross-attention mechanism adopted for 3D-skeletal sequences. We reconstruct the skeleton sequence not only based on the masked sequence but also on the knowledge obtained from the `[CLS]` embedding.

Existing works on 3D skeleton sequence generation use variational transformer autoencoder [16]. However, since they add a class label for the conditioned generation the approach is not well suited for action recognition. Our approach closes the gap between action recognition downstream tasks and conditioned data generation.

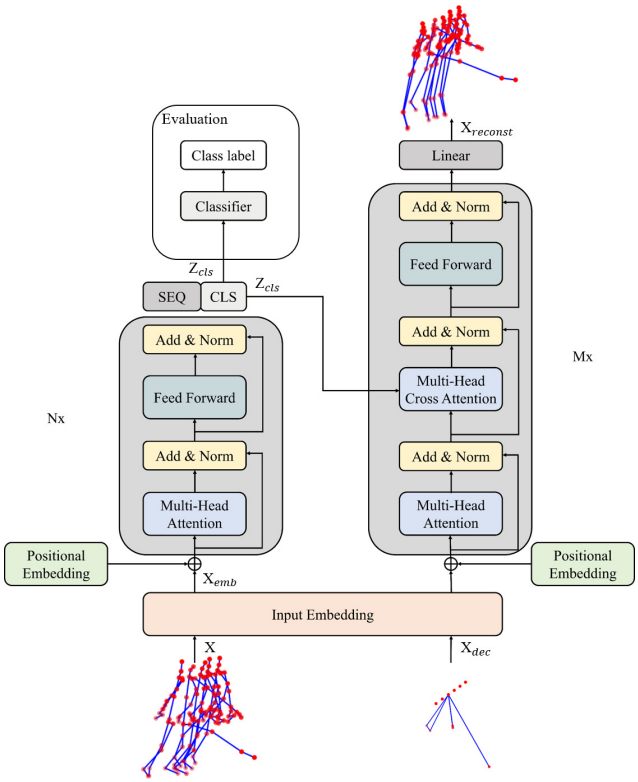

Figure 2. Architecture of the proposed transformer autoencoder. It consists of N stacked transformer encoders (left) and M stacked transformer decoders (right). The `[CLS]` embedding from the last encoder layer is used for the cross-attention in the decoder. The input to the encoder is the original sequence. The input to the decoder is the masked and noised sequence. The decoder reconstructs the original input as a denoising autoencoder.

3. Method

In this section, we present our novel transformer autoencoder architecture for 3D skeleton-based human action recognition (TAHAR). We describe our transformer autoencoder pipeline and how it supports learning meaningful representations with the transformer's cross-attention principle. With our novel autoencoder architecture we are able to contribute to classification tasks as well as denoising skeleton sequences and conditioned data generation.

3.1. Overall Architecture

Unlike other transformer autoencoders for skeleton-based human action recognition [32, 34], our proposed transformer architecture has two input streams of skeleton

sequences. As illustrated in Figure 2, both the encoder and the decoder take a 3D skeleton sequence as input. Each skeleton sequence is comprised of V joints and T frames, with each joint consisting of C distinct features. The encoder takes the original 3D skeleton sequence, represented as flattened joints X, as input and embeds it into an embedding space using an embedding layer. A learnable `[CLS]` token is prepended to the embedded sequence with the aim of extracting both sequence-level information (i.e., spatio-temporal motion) and class-wise information, which can be used to classify the skeleton sequence during evaluation. The decoder takes the masked version of the original sequence, X_{dec}, as input and learns to reconstruct the original input X by conditioning on the sequence-level and class-wise knowledge contained in the `[CLS]` embedding Z_{cls}.

With the proposed novel masking-based learning scheme, the `[CLS]` token can be trained in a fully unsupervised manner without introducing any class label. Furthermore, in contrast to the original transformer architecture [25], we choose our transformer to be non-autoregressive, allowing incorporating bi-directional information and accelerating inference.

3.2. Transformer Encoder Architecture

Following RNN-based approaches, we treat the skeleton sequence as a temporal sequence with flattened joints $\mathrm{X} \in \mathbb{R}^{T \times VC}$. The encoder input sequence is first encoded by an embedding layer, which is implemented with a linear layer and shared for both encoder and decoder respectively. Following He et al. [10], we prepend a learnable `[CLS]` token to the embedded skeleton sequence, forming the input embedding $\mathrm{X}_{emb} \in \mathbb{R}^{(T+1) \times D}$, with D being the hidden dimension both for the encoder and decoder blocks.

To preserve the temporal information, we make use of learnable positional embeddings. After N consecutive encoder blocks consisting of self-attention and feed-forward layers, the encoder transforms the encoder input sequence to a sequence of continuous hidden representations $\mathrm{Z} \in \mathbb{R}^{(T+1) \times D}$, consisting of a `[CLS]` embedding $\mathrm{Z}_{cls} \in \mathbb{R}^{1 \times D}$ and a sequence of frame representations $\mathrm{Z}_{seq} \in \mathbb{R}^{T \times D}$.

3.3. Transformer Decoder Architecture

For the decoder input, we apply multiple augmentations on the encoder input sequence, including random scale, rotation, and additive random noise, as detailed in Section 4.3. After the normalization and augmentation we apply temporal frame masking and spatial joint masking on the augmented skeleton sequence and set the masked frames and joints to zeros. For frame masking, we choose to set all joints in a frame to zero, and for joint masking, we choose to set one joint in all frames to zero. However, in contrast to other MAE approaches, where masked joints are not forwarded to the model, our decoder input with the masked joints has the same shape as the original sequence.

The decoder takes the normalized, augmented and masked sequences $\mathrm{X}_{dec} \in \mathbb{R}^{T \times VC}$ as input and embeds them with the same embedding layer as in the encoder. We choose the encoder and decoder embedding layer to be the same, since the two inputs are two different augmentations of the same sequence. The decoder consists of M consecutive blocks, each containing self-attention, cross-attention, and feed-forward layers. Instead of using the complete encoder output sequences, as done in the standard transformer architecture [25], we only use the encoder output corresponding to the `[CLS]` token $\mathrm{Z}_{cls} \in \mathbb{R}^{1 \times D}$. Finally, the generated sequence representations are transformed by a linear layer back to the encoder input space $\mathrm{X}_{\text{reconst}} \in \mathbb{R}^{T \times VC}$.

3.4. Generative Capabilities

Denoising Autoencoder The decoder of the proposed architecture takes the masked and noisy skeleton sequence as input, and maps it to continuous hidden space. By training it to reconstruct the encoder input sequence, the proposed architecture acts as a denoising autoencoder [1, 10], and learns to capture the essential features and patterns of the input while filtering out the noise and recreate masked information. As reconstructing masked skeleton sequence is particularly challenging, especially when the salient frames or joints are missing, we inject the knowledge of the original input via the `[CLS]` embedding Z_{cls} to the decoder, to facilitate the learning process of the decoder. By taking the `[CLS]` embedding into account via cross-attention, the decoder is able to reconstruct the skeleton sequence even with a strongly masked input sequence. This, in turn, benefits the learning of the encoder, i.e., encoding the original inputs to hidden representations, making the proposed architecture capable of both classification and reconstruction or generation of the encoder input sequence.

Conditioned skeleton sequence generation Our model does not require the encoder's and decoder's input to be of the same sequence. When using the same `[CLS]` embedding Z_{cls} with random noise as decoder inputs, we can vary the decoder outputs while keeping the characteristics of the sequence-level and class-wise knowledge contained in the `[CLS]` embedding. With this feature of our novel transformer autoencoder, we contribute to conditioned data generation.

3.5. Training Objective

For training, we calculate the loss between the reconstructed skeleton sequence $\mathrm{X}_{\text{reconst}} \in \mathbb{R}^{T \times VC}$ by the decoder and the original sequence $\mathrm{X} \in \mathbb{R}^{T \times VC}$. Similar to [10], we only calculate the reconstruction loss on the

masked skeleton frames and joints. In this work, we use smooth L1 loss [9] as the reconstruction loss to train the proposed approach, which combines the advantages of L1-loss (steady gradients for large reconstruction error x) and L2-loss (less oscillation during updates when x is small):

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| \leq 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \quad (1)$$

4. Experiments

We now evaluate our TAHAR model against other baselines on action recognition tasks. We perform ablation studies to evaluate cross-dataset performance and qualitatively evaluate the generative capabilities of our model. With our TAHAR model we aim to contribute to generalization tasks and therefore focus on frozen unsupervised pretrained models for the downstream tasks.

4.1. Datasets

We use three commonly used datasets for our evaluation: NTURGB+D 60 [19], NTURGB+D 120 [13], and North-Western UCLA (NW-UCLA) [28]. NTURGB+D 60 [19] is a large-scale human action recognition dataset, which contains 56,880 samples of 3D skeleton sequences. It contains 60 action classes, 40 distinct subjects and up to two subjects per sequence. Each skeleton has 25 joints. The dataset is captured from three different views. For evaluation, cross-subject (xsub) and cross-view (xview) protocols are provided.

NTURGB+D 120 [13] is an extension of the NTURGB+D 60 dataset. It contains 113,945 3D skeleton sequences categorized into 120 action classes. The NTURGB+D 120 provides a cross-setup (xset) and cross-subject (xsub) evaluation protocol respectively. For cross-dataset validation we split the NTURGB+D 120 dataset into the complementary NTU 60 (actions 1-60) and NTU 61-120 (actions 61-120).

North-Western UCLA (NW-UCLA) [28] contains 1494 action sequences performed by 10 subjects. The action sequences are captured from 3 different views. The dataset contains 10 different action classes. We choose the cross-view evaluation protocol with view 1 and 2 as training samples and view 3 as testing samples following Wang et al. [28].

4.2. Evaluation Protocols

In order to show the full potential of our unsupervised approach and to show the generalization capabilities of our model, we evaluate on frozen pretrained models instead of fine-tuning [30] [32]. For the evaluation, we employ the following two evaluation protocols.

1-NN evaluation protocol [22] is a common evaluation protocol for autoencoders in human action recognition. The model is trained unsupervised. After training a knn classifier is fitted to the training data. The evaluation is performed choosing $k = 1$ to assign the action labels to the test data.

Linear evaluation protocol (LEP) [37] is a widely used evaluation protocol for unsupervised learning tasks. The model is first pre-trained unsupervised. Thereafter, the model weights are frozen and a single linear layer is attached to the last layer of the model. The linear layer is then trained supervised on the training data.

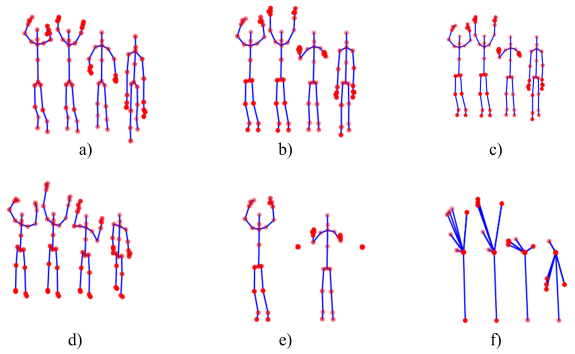

Figure 3. Skeleton sequences of a) raw sequence and b) view-invariant skeleton sequence [8]. For the encoder input we use the view-invariant skeleton sequence. For the decoder input we use augmentations of c) scale, d) rotation, e) frame masking and f) joint masking on the view-invariant skeleton sequence.

Frame	Joint	Noise	Acc.
0	0	0	19.7
0	0	0.1	54.9
0	**0**	**0.3**	**56.0**
0	0	0.5	53.0
0.6	0.6	0	51.0
0.8	**0.8**	**0**	**56.0**
1.0	1.0	0	54.8
0.8	0.8	0.1	53.8
0.8	0.8	0.3	54.1
0.8	0.8	0.5	52.1

Table 1. Evaluation of the classification accuracy in [%] for different frame and joint masking probabilities as well as different noise probabilities on NTU 60 xsub dataset with 1-NN evaluation method

4.3. Implementation Details

Data preprocessing. For data augmentation, we utilize the preprocessing pipelines provided by Duan et al. [8]. Firstly, we center the skeletons to establish a consistent reference point. Additionally, we perform normalization of orientation, ensuring that the orientations of the skeletons are standardized across different samples. Furthermore, we

employ random scale, rotation, and noise to augment the input data. The normalizations and augmentations of our prepreocessing pipeline are illustrated in Figure 3. These techniques introduce variations in the size, orientation, and noise levels of the skeletons, enabling our model to learn more robust and generalized representations [8]. Besides, we implement temporal masking (frame masking) and spatial masking (joint masking). We feed the masked joints and frames with value zero into the decoder. Therefore the size of the masked sequence equals the size of the original sequence.

Training details. We pre-train our network without labels. We use the same optimizer and scheduler for all datasets. We train with AdamW optimizer and batch size of 128. The learning rate is linearly ramped up during the first 10 epochs from 0.0001 to 0.001. Thereafter, we decay the learning rate with a cosine schedule. We pre-train the model for 90 epochs. The weight decay is set to 0.01. We choose joint and frame masking rates to 0.8 respectively. We use a smooth L1 loss function on the reconstructed skeleton. The ground truth is the normalized input sequence. The linear classifier is trained with AdamW optimizer, batch size of 128 and weight decay of 0.01 as well. The initial learning rate is set to 0.0001. The linear layer is trained for 30 epochs. The learning rate is multiplied by 0.1 at epochs 5 and 15 respectively. We select a value of two for both the number of encoder blocks N and the number of decoder blocks M. For the hidden dimension of the transformers we choose 512. We choose the noise probability to 0.1.

5. Results

In Section 5.1, we ablate the proposed transformer autoencoder architecture. Continuing with the best architecture and masking and noise configurations, we compare our proposed novel cross-attention-based autoencoder with the MAE-based models in Section 5.2, and conduct cross-dataset evaluation in Section 5.3. Finally, in Section 5.4, we compare our approach against state-of-the-art autoencoder-based methods.

5.1. Ablation Study

In this section we examine ablation experiments of our proposed framework. We evaluate the performance of our network for different hidden dimensions and depths as well as for different augmentations. Moreover, we conduct experiments on the linear evaluation and on the generative capability of our approach.

Masking rate and noise probability of the decoder inputs. In Table 1, the classification performance of the transformer autoencoder on NTU 60 for different masking and noise levels is shown. The standard deviation of the noise is

Dropout	Noise	Acc.
0	0	71.5
0	**0.1**	**72.4**
0	0.3	71.1
0	1	61.3
0.1	**0**	**72.4**
0.3	0	70.7
0.5	0	70.4

Table 2. Evaluation of the classification accuracy in [%] for different dropout levels of the linear layer and different noise levels on the output of the frozen model on NTU 60 xsub dataset with LEP

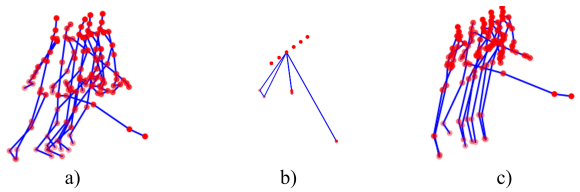

Figure 4. Input and output sequences of the transformer autoencoder. We use our transformer autoencoder to denoise the masked skeleton sequence. In a) the encoder input with activity A024: *kicking something* is shown. In b) the masked decoder input with activity A024: *kicking something* is shown. In c) the denoised decoder output is shown.

1. When there is no frame and joint masking as well as without any noise the classification performance is significantly lower than with masking or noise. Also we find that with both masking and noise the performance drops as well. The results show, that our denoising model works with both additive noise and masking. We find that with our transformer autoencoder architecture we achieve our best results with higher mask rates than SkeletonMAE [32]. Even when the decoder input is set to 0, when joint and frame mask rates are 1.0, the classification performance on NTU 60 xsub with 1-NN evaluation drops by 1.2% (see Table 1).

LEP settings. For the linear evaluation we conduct experiments including dropout and additive Gaussian noise shown in Table 2. The noise is added to the embedding directly instead of the input sequence. We find that low noise probabilities or dropout result in higher accuracy.

Denoising decoder. We now use our model as denoiser and reconstruct the skeleton sequence from a masked sequence. In Figure 4 the encoder input, the decoder input and the decoder output is shown. The encoder generates the `[CLS]` which is incorporated in the decoder using cross-attention. The masked decoder input doesn't contain enough information for the decoder to reconstruct the skeleton sequence using self-attention only. The reconstruction contains the signature movement of the activity "kicking something".

Method	Frame	Joint	# Frames	Acc.
SkeletonMAE [32]	0.8	0.8	20	9.0
	0.5	0.5	20	15.2
	0.5	0	20	26.0
	0.1	0	20	24.4
	0	0	20	26.4
	0	0	56	24.4
	0	0	100	24.6
TAHAR	0.8	0.8	-	**56.0**

Table 3. Comparison of the classification accuracy in [%] on NTU 60 xsub dataset with our implementation of the MAE architecture [10,32] against our TAHAR model. For the MAE architecture we use the configuration of [32] and explore other frame and joint masking rates as well as different numbers of frames per input sequence.

Datasets	NW-UCLA	NTU 60 xsub	NTU 61-120 xsub	NTU 120 xsub
NW-UCLA	**95.2**	93.1	92.4	**95.2**
NTU 60 xsub	66.0	**72.4**	69.3	69.8
NTU 61-120 xsub	59.0	61.6	62.8	**63.1**
NTU 120 xsub	60.4	62.5	60.5	**65.1**

Table 4. Cross-dataset evaluation on NTU 60 xsub, NTU 61-120 xsub, NTU 120 xsub and NW-UCLA datasets with the train datasets in columns and validation datasets in rows. The values are classification accuracies in [%] with the best score bold.

5.2. Masked Autoencoder

In Table 3, we compare our approach with the masked autoencoder [10, 32]. Since the proposed method takes the flattened joints as input, the joint masking method proposed in [32] does not directly apply in our case. We therefore set the masked joints to zeros if the joint masking rate is bigger than 0. The encoder takes the unmasked frames as well as a `[CLS]` token as input and maps them to hidden representations. These representations are then fed to the decoder along with the masked frames in the form of learnable `[mask]` token. To allow a fair comparison, we keep the hidden dimension and number of encoder and decoder blocks the same as in TAHAR.

We first take the default number of frames from [32] and our default frame and joint masking rate, and find that this setting is too challenging for training the MAE-based model with our implementation. We then take the default values from [32], which are 0.5 for frames and 0.5 for joints, and observe an improvement of 6.2% (9.0% $\rightarrow$ 15.2%). When no joint masking is employed, the performance rises to 26%, a result that aligns with our hypothesis.

This outcome underscores the notion that solely reconstructing from a masked skeleton sequence presents significant challenges for the optimization of the model. Additionally, we conduct experiments with no masking at all as well as variable number of frames, and observe no significant performance changes. Compared to the MAE-based model, our proposed approach leverages the cross-attention to extract the knowledge from the original skeleton sequence, which not only simplifies the task of reconstruction but also facilitates the optimization process for the model.

5.3. Cross-dataset Evaluation

For a quantitative evaluation of the generalization capabilities of our model we conduct cross-dataset evaluation on 4 different datasets and subsets (see Table 4). The results of the cross-dataset evaluation show the necessity of large datasets for the unsupervised pre-training. Exept for NTU 60, our model pre-trained on NTU 120 outperforms the specialized models even on the unseen NW-UCLA data. Our model pre-trained on the small NW-UCLA dataset with 10 action classes and evaluated with LEP on the large and complex NTU 120 xsub with 120 action classes (see Table 4) outperforms state-of-the-art models evaluated with LEP by 1.3% (see Table 5). These results show the strengths of our model to generalize from small to big datasets.

5.4. Unsupervised Action Recognition

We conduct experiments on the three datasets NTU 60, NTU 120 and NW-UCLA with the five splits NTU 60 xview, NTU 60 xsub, NTU 120 xset, NTU 120 xsub and NW-UCLA xview. For the evaluation we use both 1-NN and LEP. The 1-NN classifier is fitted to the `[CLS]` token with dimension 512. The linear layer for LEP is trained on top of the `[CLS]` token with input dimension 512.

We compare our transformer autoencoder against state-of-the-art autoencoders for human action recognition. The benchmark autoencoders use GRU architectures [3, 18, 22, 31, 33, 38] or derivatives of GRUs [15]. The results in Table 5 show that our framework achieves state-of-the-art performance on the NTU 60 dataset. Our network outperforms state-of-the-art autoencoders on the NW-UCLA by 8.6% on 1-NN and by 8.2% on LEP. While a performance decrease is noticeable when transitioning from 1-NN to LEP by CRRL [31] on NTU 60 xview and NW-UCLA, our model consistently exhibits a performance increase on all datasets. For the NTU 120 dataset the results in Table 5 show that our network outperforms the current state-of-the-art. For the large and challenging NTU 120 dataset our model outperforms the baselines by 3.9% with xset and 2.6% with xsub split and 1-NN evaluation. For the LEP we observe that our model outperforms the baselines by a margin of 2.4% and 6.0% respectively.

t-SNE visualizations. For a qualitative evaluation of our re-

	Models	NTU 60 xview	NTU 60 xsub	NW-UCLA	NTU 120 xset	NTU 120 xsub
1-NN	P&C [22]	76.3	50.7	84.9	42.7	41.7
	BGAEUN [3]	77.4	51.8	-	-	-
	CRRL [31]	75.2	**60.7**	86.4	-	-
	CR-AE [15]	**83.1**	54.1	-	44.7	42.4
	TAHAR (Ours)	79.6	56.0	**95.0**	**48.6**	**45.0**
LEP	LongT GAN [38]	48.1	39.1	74.3	-	-
	PCRP [33]	63.5	53.9	87.0	45.1	41.7
	AS-CAL [18]	64.6	58.5	-	49.2	48.6
	CRRL [31]	73.8	67.6	83.8	57	56.2
	CR-AE [15]	**85.4**	69.9	-	62.4	59.1
	TAHAR (Ours)	82.1	**72.4**	**95.2**	**64.8**	**65.1**

Table 5. Comparison of unsupervised trained autoencoders evaluated on action recognition datasets. The values are classification accuracies in [%] with 1-NN [22] and LEP evaluation method on NTU 60 and NW-UCLA and NTU 120 datasets. The best score is bold and the second best score is underlined.

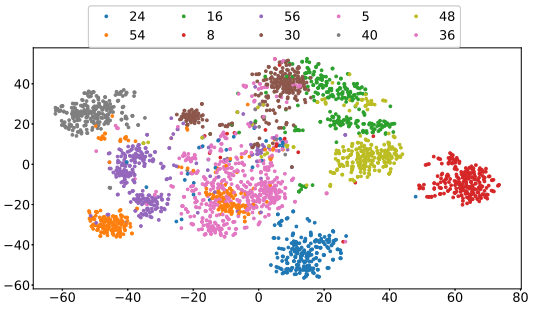

Figure 5. The t-SNE visualization of the `[CLS]` token output for the NTU 60 validation data. 10 random categories are sampled. The activites are as follow: 24: *kicking something*, 16: *wear a shoe*, 56: *giving something to other person*, 5: *drop*, 48: *nausea or vomiting condition*, 54: *point finger at the other person*, 8: *sitting down*, 30: *typing on a keyboard*, 40: *cross hands in front (say stop)*, 36: *shake head*.

sults, we provide t-SNE [24] plot of NTU 60 validation data (see Figure 5). We randomly select 10 action classes of the NTU 60 dataset. In the visualization actions with a unique joint movement like sitting, kicking or cross hands are well separated. Actions which share movements e.g. moving one hand forward such as pointing at another person and giving something to another person are poorly separated in the embedding space. The reason for this could be the limitation to one person for our approach. Extending the input for two skeletons per sequence could solve this problem.

6. Conclusions

Our transformer autoencoder for 3D skeleton-based human action recognition (TAHAR), leveraging transformer's cross-attention mechanism, and denoising autoencoder, holds promise for generalization tasks in human action recognition. Our two-stream transformer autoencoder architecture, improves recognition accuracy and generalization across datasets and even on unseen action classes. We show that our encoder is able to separate actions well when the actions contain unique joints movements, such as kicking or sitting down. Experimental results demonstrate our model's effectiveness, outperforming the state-of-the-art models on the NTU 120 and NW-UCLA datasets and achieving comparable results on NTU 60. Ablation studies on cross-dataset evaluation show great generalization capability of our model, even when trained on a small dataset. The generative capabilities of our approach can be used to generate conditioned data for many skeleton-based applications. In the future, we plan to investigate the effectiveness of the generative capability of the proposed approach for small-sized or imbalanced datasets.

ACKNOWLEDGMENT

This work was supported by the German FEDERAL MINISTRY FOR ECONOMIC AFFAIRS AND CLIMATE ACTION through the INITIATIVE project and by the Carl Zeiss Foundation through the JuBot project.

References

[1] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2013. 4

[2] Nicolo Carissimi, Paolo Rota, Cigdem Beyan, and Vittorio Murino. Filling the gaps: Predicting missing joints of human poses using denoising autoencoders. In *Proceedings*

of the European Conference on Computer Vision (ECCV) Workshops, September 2018. 1

[3] Li Chen, Nan Ma, and Guoping Zhang. Bi-gru-attention enhanced unsupervised network for skeleton-based action recognition. In *International Conference on Autonomous Unmanned Systems*, 2022. 2, 3, 7, 8

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[5] Yi-Bin Cheng, Xipeng Chen, Junhong Chen, Pengxu Wei, Dongyu Zhang, and Liang Lin. Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021. 2

[6] Sangwoo Cho, Muhammad Maqbool, Fei Liu, and Hassan Foroosh. Self-attention network for skeleton-based human action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 2

[7] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *IAPR Asian Conference on Pattern Recognition*, 2015. 2

[8] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition. In *Proceedings of ACM International Conference on Multimedia*, 2022. 5, 6

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2015. 5

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3, 4, 7

[11] Boeun Kim, Hyung Jin Chang, Jungho Kim, and Jin Young Choi. Global-local motion transformer for unsupervised skeleton-based action learning. In *Computer Vision – ECCV 2022*, 2022. 2, 3

[12] Yaqian Liang, Shanshan Zhao, Baosheng Yu, Jing Zhang, and Fazhi He. Meshmae: Masked autoencoders for 3d mesh data analysis. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*. Springer Nature Switzerland, 2022. 1

[13] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10), 2020. 2, 5

[14] Manuel Martin, David Lerch, and Michael Voit. Viewpoint invariant 3d driver body pose-based activity recognition. In *2023 IEEE Intelligent Vehicles Symposium (IV)*. 1

[15] Giancarlo Paoletti, Cigdem Beyan, and Alessio Del Bue. Graph laplacian-improved convolutional residual autoencoder for unsupervised human action and emotion recognition. *IEEE Access*, 10, 2022. 2, 7, 8

[16] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[17] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition, 2022. 2, 3

[18] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 2021. 2, 7, 8

[19] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 5

[20] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[21] Sijie Song, Cuiling Lan, Juynliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 2

[22] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 7, 8

[23] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3), 2023. 1

[24] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9, 2008. 8

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017. 3, 4

[26] Cailing Wang and Jingjing Yan. A comprehensive survey of rgb-based and skeleton-based human action recognition. *IEEE Access*, 11, 2023. 1

[27] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[28] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2, 5

[29] Lei Wang, Du Q. Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29, 2020. 1

[30] Lei Wang and Piotr Koniusz. 3mformer: Multi-order multi-mode transformer for skeletal action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 5

[31] Peng Wang, Jun Wen, Chenyang Si, Yuntao Qian, and Liang Wang. Contrast-reconstruction representation learning for self-supervised skeleton-based action recognition. *IEEE Transactions on Image Processing*, 2022. 7, 8

[32] Wenhan Wu, Yilei Hua, Ce zheng, Shiqian Wu, Chen Chen, and Aidong Lu. SkeletonMAE: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition, 2022. 1, 2, 3, 5, 6, 7

[33] Shihao Xu, Haocong Rao, Xiping Hu, Jun Cheng, and Bin Hu. Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition. *IEEE Transactions on Multimedia*, 2023. 2, 7, 8

[34] Hong Yan, Yang Liu, Yushen Wei, Guanbin Li, and Liang Lin. Skeletonmae: Graph-based masked autoencoder for skeleton sequence pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[35] Han Yao, S-J Zhao, Chi Xie, Kenan Ye, and Shuang Liang. Recurrent graph convolutional autoencoder for unsupervised skeleton-based action recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2

[36] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[37] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016. 5

[38] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2, 7, 8

[39] Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer. Anticipative feature fusion transformer for multi-modal action anticipation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2