ProFormer: Learning Data-efficient Representations of Body Movement with Prototype-based Feature Augmentation and Visual Transformers

Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen
Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology
{firstname.lastname}@kit.edu

***Abstract*— Automatically understanding human behaviour allows household robots to identify the most critical needs and plan how to assist the human according to the current situation. However, the majority of such methods are developed under the assumption that a large amount of labelled training examples is available for all concepts-of-interest. Robots, on the other hand, operate in constantly changing unstructured environments, and need to adapt to novel action categories from very few samples. Methods for *data-efficient* recognition from body poses increasingly leverage skeleton sequences structured as image-like arrays and then used as input to *convolutional* neural networks. We look at this paradigm from the perspective of *transformer* networks, for the first time exploring visual transformers as data-efficient encoders of skeleton movement. In our pipeline, body pose sequences cast as image-like representations are converted into patch embeddings and then passed to a visual transformer backbone optimized with deep metric learning. Inspired by recent success of feature enhancement methods in semi-supervised learning, we further introduce PROFORMER – an improved training strategy which uses soft-attention applied on iteratively estimated action category PROtotypes used to augment the embeddings and compute an auxiliary consistency loss. Extensive experiments consistently demonstrate the effectiveness of our approach for one-shot recognition from body poses, achieving state-of-the-art results on multiple datasets and surpassing the best published approach on the challenging NTU-120 one-shot benchmark by 1.84%. Our code will be made publicly available at https://github.com/KPeng9510/ProFormer.**

I. INTRODUCTION

There have been impressive advances activity recognition frameworks tailored for robotics applications [1], [2], [3], [4], [5], [6], [7], [8]. However, this task remains very challenging in practice, as agents mostly operate in an open constantly changing environment and we will never be able to capture and annotate a high amount of training examples for every possible category [9], which is a requirement in the majority of presented approaches. Especially in the light of growing elderly population, *data-efficient* recognition of Activities of Daily Living (ADL) is a vital ingredient for household robot perception and situation-aware assistance [10].

Problems of data-efficient ADL recognition are often posed in the form of one- or few-shot learning [11], [12], [13], [14] and addressed with metric learning [15], [16], [17] or meta learning [18]. For example, the state-of-the-art approaches for one-shot action recognition from body pose data [15], [15] leverages a Convolutional Neural Network (CNN)-based encoder for signal-level skeleton representation and the deep metric learning paradigm. While few recent works considered *transformer* networks in conventional *video*-based human activity classification [19], their potential as signal encoders of *body movement* and transfer capabilities to *new data-scarce activities classes* has been overlooked.

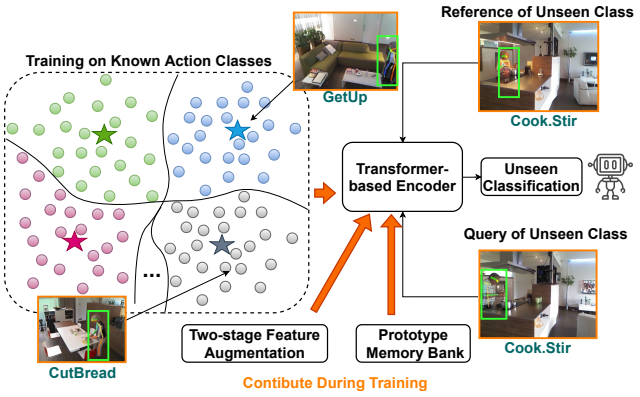

Fig. 1. An overview of the proposed PROFORMER approach for data-efficient representations of body movement using visual transFORMERS and an enhanced training strategy with feature-level augmentations through action category PROtotypes.

Multiple approaches cast skeleton representations as image-like structures [15], [20], [21], oftentimes encoding the time dimension in the rows, skeleton joints in the columns and the 3D coordinates as the channels ($N_{joints} \times N_{frames} \times 3$). Such encodings preserve the communication among different joints under a certain time range and the resulting tensors are passed to a conventional CNN, therefore allowing us to reuse CNN-based architectures developed with image input in mind for the body pose data. Could we also leverage such image-like modelling of skeleton dynamics with the rapidly emerging visual transformers? Different segments of a body movement sequence are not equally transferable. We therefore believe that the transformer networks, which learn to amplify or blend potions of input through self-attention and are also known to suit especially well for sequential data [22], are an excellent tool for building well-generalizable ADL recognition models.

Motivated by this, we seek to investigate the reuse of visual transformers initially developed for images as signal-level body movement encoders in ADL and, additionaly, introduce a new training strategy leading to more robust models. We specifically target data-scarce recognition and

The research leading to these results was supported by the SmartAge project sponsored by the Carl Zeiss Stiftung (P2019-01-003; 2021-2026).

conduct a systematic study featuring off-the-shelf transformer architectures benchmarked in the tasks of skeleton-based one-shot activity recognition. Our framework comprises image-like representation of skeleton movement, *i.e.*, signal-level representation, different variants of the visual transformer backbone and deep metric learning optimization and proves to be more effective in data-scarce human activity recognition than its CNN-based counterparts.

However, problems of data-scarce recognition are not restricted to the architecture and encoding choice. An important question is *how to train your network* in order to obtain well-adaptable representations. Inspired by recent advances in semi-supervised learning [23], [24], [25], we introduce a new optimization strategy, which expands the deep metric learning paradigm with an auxiliary loss for encouraging invariance to transformations through consistency constraints and augmentations with previously learnt action category prototypes. We refer to our final PROtotype based transFORMER architecture as PROFORMER, since it is optimized with consistency- and PROtotype-based feature enhancement.

We show that our method achieves state-of-the-art results across three different benchmarks for skeleton-based one-shot ADL recognition, outperforming the best published results on the challenging NTU-120 benchmark [26] by 1.84%. However, the benefit of the PROFORMER optimization goes beyond well-adaptable movement embeddings. Presumably due to the enhanced intermediate feature augmentations used in PROFORMER, we observe remarkable advantages in case of noise corruptions ($> 35\%$ on NTU-120 with $\sigma = 0.1$ Gaussian noise).

To summarize, our contributions are:

1) We for the first time explore visual transformers as data-efficient encoders of skeleton movement sequences cast as image-like representations.
2) We introduce PROFORMER – an optimization strategy, where the intermediate transformer representations are enhanced through iteratively estimated category-specific prototypes and an auxiliary consistency loss.
3) We conduct in-depth experiments in one-shot ADL recognition tasks, demonstrating clear benefits of visual transformers as data-efficient body movement encoders. Our PROFORMER model yields state-of-the-art on three datasets [15], [16], [26], surpassing the best published approach on the challenging NTU-120 benchmark by 1.84% for one-shot action recognition.
4) As a side-observation, we discover that PROFORMER optimization strategy is much more resistant to noise corruptions, outperforming the same backbone trained with conventional deep metric learning strategy. We believe this to be of vital importance for robotics applications, since agents often face unstructured environments wth high levels of noise.

II. RELATED WORK

Data-efficient ADL recognition. Despite the impressive improvements in general activity recognition [27], [28], [29] and a variety of approaches developed specifically for robotics [4], [5], [6], [8], [10], deploying such models in practice is very hard, since robots often operate in dynamic environments where changes of potential activities may occur at any time. Still, the majority of previously published research strives for high accuracy on conventionl ADL recognition datasets [5], [26], [30] assuming that a large amount of labelled training examples is available for every activity-of-interest. This is impractical in real robotics applications, where data-efficient learning of new concepts on-the-fly remains a key challenge [9].

Problems of learning activity representations which adapt well to new data-scarce categories are often posed in the form of few-shot recognition, where the methods usually fall into one of two categories: (1) meta-learning-based methods [18], [31], [32], [33], which reinitialize a new set of tasks every epoch following the "learning to learn" paradigm and (2) metric-learning-based methods [15], [16], [17], which aim to project the input to a lower-dimensional space, where same-category samples are close to each other and the inter-category ones are far apart. Our approach falls in the latter category. Few-shot learning has been well-studied in object detection [34], [35] and various recognition tasks [36], [37]. The research of one-shot activity recognition from 3D body poses has been much more sparse and is largely studied in the context of one-shot recognition on the NTU-120 dataset [15], [26], [38], [39], [40]. State-of-the-art recognition results are currently reached by the approach of Memmesheimer *et al.* [16], which uses a CNN-based encoding of 3D skeletons represented as images and optimizes the framework with deep metric learning using a mixture of cross entropy and triplet margin losses. While we specifically focus on learning data-efficient representations of *3D body poses* [15], we need to acknowledge an array of work on *video*-based activity recognition from few training examples [11], [13]. In this work, we for the first time explore the potential of encoding body pose sequences directly using visual transformers, which has been addressed with CNN-based [15] encoders in the past. Our optimization procedure also builds on the deep metric learning paradigm of [15] further extending it with an auxiliary branch providing consistency loss using learned augmentations at feature-level augmented by cluster prototypes.

Visual transformers. Transformer networks [41] are rapidly gaining popularity in computer vision since their operationalization on image patched within the ViT [42] and DeiT [43] architectures. Following this trend, a high number of models has been proposed, which are designed to achieve high accuracy [44], resource-efficiency [45], [46] or are tailored for specific tasks, such as object detection [47] or semantic segmentation [22]. While multiple works leverage *visual* transformer models in conventional *video*-based human activity classification [19], [48], [49], or standard sequence transformers for skeleton encodings [28], [50], [51], their potential of *visual* transformers as data-efficient encoders of *body movement* cast as images has not been considered yet and is the main motivation of our work.

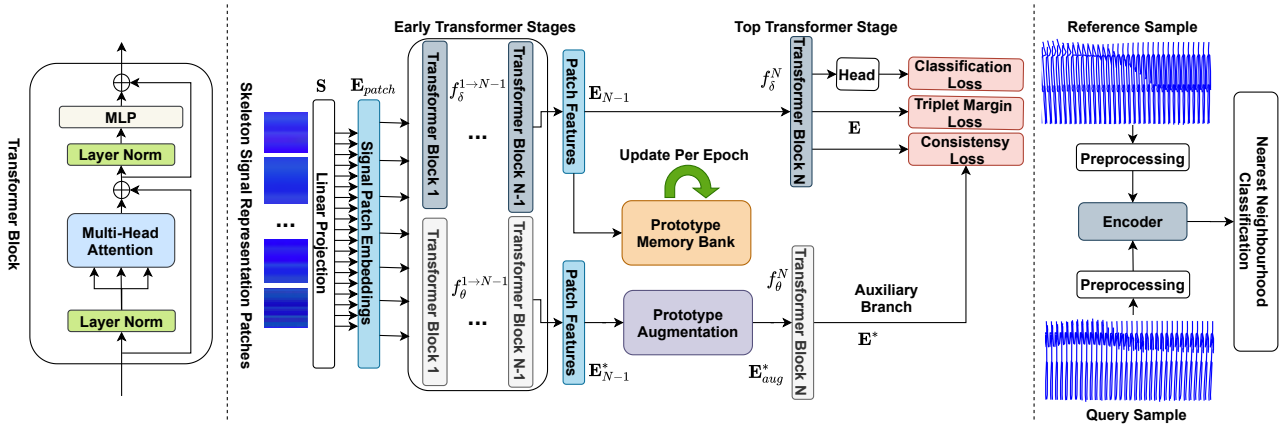

Fig. 2. Overview of the PROFORMER architecture. First, sequences of 3D body poses are encoded as image-like arrays (rows: time dimension, columns: joint IDs, channels: 3D coordinates, see example visualizations in blue on the right). Then, the representations are divided into 2D patches and passed together with the positional embeddings through a linear projection layer to compute 1D representation $\mathbf{E}_{patch}$. Next, our model splits into the *main* and the *auxiliary branches* with multiple transformer layers each, resulting in two intermediate embeddings $\mathbf{E}_{N-1}$ and $\mathbf{E}^*_{N-1}$ (auxiliary branch is marked with *). During training, $\mathbf{E}_{N-1}$ is used to iteratively estimate action-category specific representations ("*prototypes*") stored in the *prototype memory bank*. These prototypes are then used to augment the embedding of the auxiliary branch $\mathbf{E}^\star_{N-1}$ in a two-stage manner resulting in the augmented feature $\mathbf{E}^*_{aug}$. The embeddings $\mathbf{E}_{N-1}$ and $\mathbf{E}^*_{aug}$ are passed through the final transformer block to obtain the final *main* and feature-augmented *auxiliary* embeddings $\mathbf{E}$ and $\mathbf{E}^*$. The mismatch of $\mathbf{E}$ and $\mathbf{E}^*$ is used to estimate the *consistency cost* used together with the classification and triplet margin losses for optimization.

Furthermore, inspired by the recent success of feature augmentation method in semi-supervised learning [24], we for the first time propose training a visual transformer with an additional auxiliary branch for augmenting the embeddings using category-specific prototypes and self-attention.

III. VISUAL TRANSFORMER NETWORKS FOR DATA-EFFICIENT REPRESENTATIONS OF BODY POSES

Our proposed approach is a transformer-specific training strategy for robust one-shot action recognition. We refer to the model leveraging both ideas as PROFORMER, since the enhancements at feature-level are achieved with the help of iteratively estimated action category PROtotypes. An overview of the PROFORMER architecture is in Figure 2 and Algorithm 1. Next, we give a formal definition of the addressed task (Sec. III-A); describe the general skeleton embedding pipeline and fundamentals of the visual transformer backbone (Sec. III-B); and, finally, we present our complete PROFORMER method leveraging augmentations through iteratively estimated prototypes of activity categories (Sec. III-C).

A. Problem formulation

Our idea is to leverage visual transformers to learn well-adaptable skeleton movement embeddings which generalize to new activity types with very little training data. The task we address is one- and few-shot activity recognition from body poses [15] where a priori knowledge acquired from data-rich action classes is transferred to categorize new data-scarce classes. Formally, C_{base} denotes the set of $|C_{base}|$ data-rich categories available during training through large amount of labelled data $D_{base} = \{(\mathbf{S}_i, l_i)\}_{i=1}^{U}$, $l_i \in C_{base}$ while U indicates the number of samples in D_{base}. Our goal is to distinguish the $|C_{novel}|$ new activity classes C_{novel}, for which only κ reference training examples are available for each class (where κ can be as few as 1). These data-scarce examples are referred to as support set $D_{supp} = \{\mathbf{S}_i\}_{i=1}^{O}$ while O indicates the number of samples in D_{supp} and $C_{base} \cap C_{novel} = \emptyset$. The final task is then to assign a category $l_n \in C_{novel}$ to each sample from the test set D_{test} containing samples from the data-scarce categories C_{novel}.

B. Rethinking visual transformers for body poses

1) From 3D skeleton joints to image-like representations and patch embeddings: Let $\mathbf{S}$ be a given sequence of T body poses comprising the 3D coordinates (x,y,z) of J skeleton joints. As mentioned before, multiple published frameworks encode the joint positions or their displacements in 2D arrays, which are then passed to convolutional networks [15], [20], [21], [52]. In a similar fashion, we form the image-like representation by first concatenating the coordinates of different joints $(x_i^t, y_i^t, z_i^t), i \in [0,J], t \in [0,T]$ along the joint axis (indexed with j) resulting in a single time-step vector with dimensionality $\mathbb{R}^{J\times 3}$ and then chaining these vectors along the temporal axis t, which leads to an image-like representation of shape $\mathbb{R}^{T\times J\times 3}$.

While data in this form can be directly passed to a 2D convolution layer, a conventional transformer layer [41] operates on 1D sequences of token embeddings and our body pose representation should meet this requirement. In visual transformers, this issue is often solved through *patch embeddings* [42], dividing the 2D space patches, which are then represented through learnt 1D embeddings. To this intent, we downsample a 2D body pose representation $\mathbf{S} \in \mathbb{R}^{T\times J\times 3}$ into $\frac{TJ}{M^2}$ patches of size $M \times M$. The flattened content of each patch is passed to a linear projection layer $h_\gamma : \mathbb{R}^{(M^2\cdot 3)} \to \mathbb{R}^H$, resulting in a 1D vector $\mathbf{e_i} \in \mathbb{R}^H$ representing the ith patch embedding. Additionally, positional embedding $\mathbf{s}_i$ are linearly added to inject spatial information of the patch. Then final sequence used as input to the transformer layer then becomes $\mathbf{E}_{patch} = \{\mathbf{e}_1 + \mathbf{s}_1, \mathbf{e}_2 + \mathbf{s}_2, ..., \mathbf{e}_L + \mathbf{s}_L\}$, where

the overall sequence length L corresponds to the number of patches $L = \frac{TJ}{M^2}$.

2) Visual transformer architecture: The key components of a vanilla transformer block (see Figure 2 on the left) are Multi-head Self-Attention (MSA) and Multi-Layer Perceptron (MLP), coupled with layer normalization and residual connections at the end of each block [41], [42]. MSA chains multiple Self-Attention (SA) layers, which are mostly based on three concepts, *Query*, *Key* and *Value* denoted as $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$: $SA(\mathbf{Q},\mathbf{K},\mathbf{V}) = Softmax(\mathbf{Q}\mathbf{K}^T/\sqrt{s_k})\mathbf{V}$, where s_k is the scaling factor to avoid the influence brought by dot product on the variance. The self-attention mechanism uses linear projections $\mathbf{P}$ to get the three main components $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ based on the sequence input $\mathbf{E}_{patch}$, so that $\mathbf{Q} = \mathbf{P}_Q\mathbf{E}_{patch}, \mathbf{K} = \mathbf{P}_K\mathbf{E}_{patch}, \mathbf{V} = \mathbf{P}_V\mathbf{E}_{patch}$. Multiple SA results are then linked in MSA through another linear projection $\mathbf{P}_{MSA}$ applied on concatenated SA outputs: $MSA(\mathbf{E_{patch}}) = Concat(SA_1, SA_2, ..., SA_N)\mathbf{P}_{MSA}$. The ***Swin*** [44] model is built exclusively upon transformer layers but features MSA blocks with reduced complexity with shifted windows. The ***LeViT*** [45] uses attentional bias instead of the positional embeddings in order to reduce its influence while generating *Key*. Furthermore, LeViT leverages a CNN before the patch embedding step.

Deep metric learning optimization and inference. For optimization, we follow the Deep Metric Learning (DML) of Memmesheimer *et al.* [15]. We use Multi-Similarity Miner [53] to select the most informative positive and negative pairs of embeddings which are then leveraged to create triplets $\{(\mathbf{E}_a, \mathbf{E}_n, \mathbf{E}_p)\}_{i=1}^{N_t}$ for the triplet loss with margin λ: $L_{tpl} = \sum_{i=1}^{N_{tpl}}[\|\mathbf{E}_{a_i} - \mathbf{E}_{n_i}\| - \|\mathbf{E}_{a_i} - \mathbf{E}_{p_i}\| + \lambda]_+$, where N_{tpl} denotes the number of triplets. As in [15], we use an additional classification loss L_{class} (cross entropy), so that the final loss is a weighted sum of the aforementioned losses: $L_{all} = \alpha L_{tpl} + \beta L_{class}$. When only few reference sample are available at test-time, we pass the test example as well as the reference samples through our model to obtain the embeddings and then use nearest-neighbour to assign the final category.

C. PROFORMER: *Improving transformer training with prototype feature augmentation*

In this section, we introduce PROFORMER – an optimization strategy for learning data-efficient movement embeddings inspired by the recent success of feature augmentation methods in semi-supervised learning [23], [24], [25]. In addition to the transformer-based architecture and conventional deep metric learning losses, PROFORMER leverages a consistency cost encouraging the model to output similar results if the body movement embedding is passed through transformations. We achieve these transformations on feature-level by augmenting the intermediate representations through iteratively estimated action category prototypes and learnt attention.

1) Feature-augmenting auxiliary branch: Conceptually, the PROFORMER framework divides the architecture in two branches after the patch embedding: the *main* branch

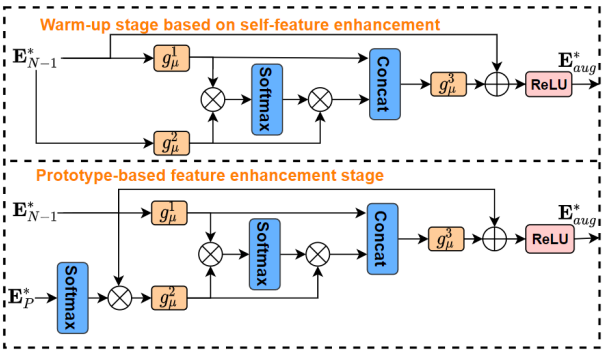

Fig. 3. Overview of the self-augmentation at feature-level leveraged in the auxiliary branch of the proposed method. During the warm-up phase (top), the feature itself is used as the basis to compute attention masks used to self-augment the feature. At the later stage, we use action-specific prototypes Softmax-normalized along the channel dimension in order to augment the embedding (bottom).

used for the conventional DML training and inferring the final movement embedding $\mathbf{E}$ used at test-time, and the *auxiliary* branch aimed to produce *augmented* embedding $\mathbf{E}^\star$, after which the consistency cost of the main- and auxiliary embeddings is estimated as *consistency loss* based on cosine similarity: $L_{cons} = 1 - cos(\mathbf{E}, \mathbf{E}^*)$. Additionally, the embeddings of the *main* branch are used to compute activity-specific prototypes, which, in return, are used to augment the features in the *auxiliary* branch explained in the next section. The auxiliary branch is active only during training supervised only via the consistency loss, while the main branch is trained through a combination all three losses: $L_{all} = \alpha * L_{tpl} + \beta * L_{class} + \gamma * L_{cons}$. Intuitively, we incentivise the main branch encoder to push the initial and the auxiliary-branch-augmented embeddings closer.

2) Feature Enhancements via Action Category Prototypes and Self-augmentation Warm-Up: We build on the top of a recent feature augmentation approach for semi-supervised learning, but additionally propose a warm-up self-augmentation phase and certain architecture changes, which have proven to be effective in augmenting body movement.

Estimating action category prototypes. For the auxiliary branch augmentations at feature-level, we draw inspiration from FeatMatch [24], a recent method for self-supervised image classification, where a learnt weighted combined category-specific prototypes is used to enhance the intermediate features when referring to feature-level augmentations. Specifically, for each data-rich action category $l_i \in C_{base}$, we iteratively estimate its prototype in the latent space as the center of all training set embeddings of the specific action (we use the embeddings after the $N-1$ layer if N is our number of transformer layers). Note, that unlike FeatMatch, we use the centers of the data-rich base categories available during training (while clustering is used in self-supervised learning due to absence of labels). Every epoch, these action category prototypes are iteratively updated and stored into a fixed-sized vector, which we refer to as the *Prototype Memory Bank* (PMB). These action category prototypes are then used for feature augmentations in order to estimate the

Algorithm 1 Training Procedure of PROFORMER

Input: $\mathbf{S}$ – a batch in D_{train}; $\mathbf{S}_p$ and $\mathbf{S}_n$ – positive and negative anchor; $f_\delta^{1\to N-1}$ and $f_\theta^{1\to N-1}$ – first N-1 transformer layers of main and auxiliary branches; f_δ^N and f_θ^N – the N-th (last) transformer layer for main and auxiliary branches; EMB – Embedding layer; N_e – maximum training epochs; N_t – epoch threshold for the stage changing; $\mathbf{E}$ and $\mathbf{E}^*$ – embedding for main and auxiliary branches; PMB – Prototypes memory bank; WarmUpAug and PrototypeAug – Warm-up stage and prototype-based feature augmentation stage

for all $epoch \in Range(N_e)$ **do**
  for all $\mathbf{S} \in D_{train}$ **do**
    if $epoch > N_t$ **then**
      for all l *in* $label_S$ **do** $Append(PMB[l]) \to List_p$
      end for
      $\mathbf{E}_P^* = Concat(List_p)$
    end if
    if *BaseModel is not None* **then** $\mathbf{S} = BaseModel(\mathbf{S})$
    end if
    $\mathbf{E}_{patch} = PatchEmbeddingAndEncoding(\mathbf{S})$
    $\mathbf{E}_{N-1} = f_\delta^{1\to N-1}(\mathbf{E}_{patch})$, $\mathbf{E}_{N-1}^* = f_\theta^{1\to N-1}(\mathbf{E}_{patch})$
    if $epoch < N_t$ **then** $\mathbf{E}_{aug}^* = WarmUpAug(\mathbf{E}_{N-1}^*)$
    else $\mathbf{E}_{aug}^* = PrototypeAug(\mathbf{E}_{N-1}^*, \mathbf{E}_P^*)$
    end if
    $\mathbf{E} = EMB(f_\delta^N(\mathbf{E}_{N-1}))$, $\mathbf{E}^* = EMB(f_\theta^N(\mathbf{E}_{aug}^*))$
    $L_{tpl} = TripletMarginLoss(\mathbf{E}, \mathbf{E}_n, \mathbf{E}_p)$
    $L_{cons} = ConsistencyLoss(\mathbf{E}, \mathbf{E}^*)$
    $L_{class} = ClassificationLoss(Head(\mathbf{E}), label_S)$
    BackPropagation$(L_{tpl} + L_{class} + L_{cons})$
  end for
  if $epoch < N_t - 1$ **then**
    $CalculatePrototypes(D_{train}) \to Set(\mathbf{E}_{N-1}) \to PMB$
  end if
end for

consistency cost.

Prototype-based feature enhancement with self-augmentation warm-up. Leveraging prototype-based augmentation in context of one-shot learning requires further conceptual changes. First, since the prototypes indeed correspond to actual action categories from C_{base} (*i.e.* only one of the current training categories is correct), we first apply Softmax normalization across the channel dimension for prototypes vector $\mathbf{E}_P^*$ and then refine it with the features $\mathbf{E}_{N-1}^*$ and project it into an embedding space as $\mathbf{E}_{r,N-1}^* = g_\mu^2(Softmax(\mathbf{E}_P^*) \cdot \mathbf{E}_{N-1}^*)$. At the same time, the features $\mathbf{E}_{N-1}^*$ is also projected as $\mathbf{E}_{l,N-1}^* = g_\mu^1(\mathbf{E}_{N-1}^*)$. Then the attention weight $\mathbf{W}$ is calculated as $\mathbf{W} = Softmax(\mathbf{E}_{l,N-1}^{*\,T}\mathbf{E}_{r,N-1}^*)$. After aggregating the information coming from prototypes vector $\mathbf{E}_P^*$ to original feature $\mathbf{E}_{N-1}^*$ as,

$$\mathbf{E}_{agg,N-1}^* = g_\mu^3([\mathbf{W}\mathbf{E}_{r,N-1}^*, \mathbf{E}_{l,N-1}^*]), \tag{1}$$

the final augmented feature $\mathbf{E}_{aug}^*$ is then obtained through a residual connection with original feature $\mathbf{E}_{N-1}^*$ by $\mathbf{E}_{aug}^* = ReLU(\mathbf{E}_{N-1}^* + \mathbf{E}_{agg,N-1}^*)$, where g_μ^1 and g_μ^2 indicate two fully-connected (fc) layers (no weight sharing), and g_μ^3 indicates a stack of two fc layers with ReLU in the middle. $[\cdot]$ denotes concatenation. As in our case the prototypes are linked to true action categories from C_{base} (in contrast to unsupervised clustering necessary in self-supervised tasks), using centers of the assigned categories in the early training epochs would be unreliable. To alleviate this issue, we introduce an additional *warm-up phase*. The key idea is to leverage self-augmentation instead of prototype-based augmentation until certain level of convergence is reached. At earlier stages, we therefore replace the attended prototype representation with the embedding $\mathbf{E}_{N-1}^*$. Figure 3 illustrates the difference between the self-augmentation warm-up phase (top) and the prototype-based augmentation (bottom).

IV. EXPERIMENTS

A. Datasets and implementation details

We perform comprehensive studies for one-shot human activity recognition from 3D body poses on three challenging datasets: NTU-60 [54], NTU-120 [26] and Toyota Smarthome [30]. To compare with competitive state-of-the art methods for data-efficient recognition, we select NTU-120 as our primary test bed, as it is a well-established benchmark for one-shot recognition from 3D body poses [15], [26], [38], [39], [55]. Additionally, we adapt the evaluation protocols of Toyota Smarthome and NTU-60 to suit our data-scarce representation learning task. The NTU-120/NTU-60/Toyota Smarthome benchmarks feature $100/48/24$ data-rich training categories and $20/12/7$ data-scarce test categories respectively for which $\kappa \in \{1,3,5\}$ reference examples are available (see problem definition in Sec. III-A).

Implementation details. For PROFORMER training we set the warm-up phase threshold $N_t = 20$ and train our model using RMSProp [56] for 100 epochs and batch size of 32. To reproduce the performance of our approach on the NTU-120, we use the learning rate of $3.5e^{-6}$ with the weights of three different losses selected as 0.5. In addition to the aforementioned implementation details of our approach on the NTU-120 dataset [26] for one-shot action recognition in our paper, for the other two datasets, the learning rate is chosen as $3.5e^{-4}$ due to the different difficulty level of different datasets and the weight of the consistency loss for the experiments on the NTU-60 [54] and Toyota smarthome [30] is chosen as 0.1 together with batch size selected as 32. The detailed evaluation protocols will be released in our github repository.

B. Experiment results

Table I illustrates the one-shot recognition results on NTU-120 [26] for different variants of our transformer-based model compared with (1) the previously published methods for one-shot recognition from body poses (first group of approaches), (2) the CNN-based skeleton encoding baselines implemented for the *Transformer vs. CNN* study (second group), and against each other. In the latter case, we first consider (3) two off-the-shelf visual transformers trained with the native DML paradigm as explained in Sec. III-B (4th group in Table I), and, finally, we consider (4) the two selected variants (with the LeViT and Swin backbones) equipped with the proposed PROFORMER optimization described in Sec. III-C (the last group of approaches in Table I). The results further validated through three-shot experiments

TABLE I

COMPARISON TO THE EXISTED METHODS FOR ONE-SHOT ACTIVITY RECOGNITION FROM 3D BODY POSES ON NTU-120,† INDICATES THE APPROACHES ARE IMPLEMENTED BY [26]

Encoder	Accuracy	F1	Recall	Prec
Previously published Approaches				
AN† [39]	41.0	-	-	-
FC† [39]	42.1	-	-	-
AP† [38]	42.9	-	-	-
APSR [26]	45.3	-	-	-
TCN-OneShot [55]	46.3	-	-	-
SL-DML [15]	50.9	-	-	-
Skeleton-DML [40]	54.2	-	-	-
Transformer-based Encoder of 3D Body Poses optimized with DML (ours)				
SL-DML (Swin) [44]	53.13	52.09	53.48	53.16
SL-DML (LeViT) [45]	53.19	52.22	53.85	53.29
CNN-based Encoder w. Prototype Augmentation (ours)				
ProCNN (SL-DML [15])	46.93	45.40	47.17	46.96
Transformer-based Encoder w. Prototype Augmentation (ours)				
ProFormer (Swin)	54.70	53.39	54.81	54.63
ProFormer (LeViT)	**55.94**	**54.29**	**55.80**	**56.04**

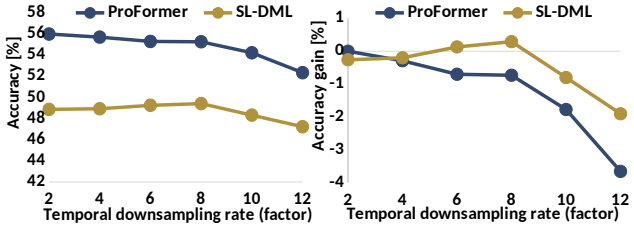

Fig. 4. Impact of temporal resolution on a transformer- and a CNN-based model (SL-DML refers to the approach of [15] which encodes body pose sequences with a ResNet). We drop portions of the input sequence at test-time (a downsampling rate of 12 means that $\frac{1}{12}$th of the input was used).

on NTU-120, NTU-60 and Toyota Smarthome (Table II), where we compare the best performing CNN-based (leveraging the ResNet18 backbone, *i.e.*, the approaches of [15], [15]) and transformer-based approaches (using Swin and LeViT backbones). In all test beds, we consider the transformer-based body pose embedding framework with and without the PROFORMER optimization.

1) Evaluation of the PROFORMER *approach:* Next, we empirically evaluate the PROFORMER training strategy. Specifically, we equip the selected best transformer-based architectures, (*i.e.*, variants with LeViT [45] and Swin [44] backbones) with the auxiliary branch for attention-based augmentations via feature-level prototypes introduced in Sec. III-C. This branch is used to compute an *additional* auxiliary consistency loss and the most meaningful comparison of PROFORMER is therefore against its same-backbone-counterpart optimized with deep metric learning only. Although our method does not influence the architecture at test-time, it performs surprisingly well. For example, for one-shot recognition, PROFORMER optimization leads to accuracy gains between 1.57% (NTU-120, Table I) and 6.21% (Toyota Smarthome, Table II) for the Swin backbone. We observe the benefits of PROFORMER across all datasets, surpassing the two best previously published approach SL-DML [15] by $>5\%$ and Skeleton-DML [40] by 1.84% (Table I). Interestingly, the LeViT-based PROFORMER achieves the best results on all settings of NTU-120 and NTU-60, while the Swin-based PROFORMER is considerably better on Toyota Smarthome (Table II). One potential explanation is the less controlled environment in Toyota Smarthome resulting in noisier body pose sequences. Swin simplifies the multi-scale attention using shifted window, which might lead to less overfitting to noisy data. The visual transformer-based encoder structure shows better performance while comparing with GCN-, CNN- and skeleton-transformer based encoder as illustrated in Table V for one shot action recognition on NTU 120 dataset, where CTR-GCN [29] and ST-TR [28] are separately leveraged for GCN-based and skeleton-transformer based encoders. As shown in Table I, we also evaluate the performance of the proposed two-stage training strategy with a CNN-based encoder (specifically SL-DML), marked as PROCNN. Interestingly, the performance of PROCNN decays compared with SL-DML approach, indicating that the proposed training strategy is effective specifically for transformer-based architectures.

The effect of the warm-up phase. As explained in Sec. III-C, our method is inspired by the recent FeatMatch [24] technique proposed for semi-supervised image classification, but has multiple conceptual changes, mostly motivated by the fact, that our approach can leverage prototypes specific for the data-rich movement categories from C_{base}, while this is not the case in semi-supervised learning. The most important difference is presumably the additional warm-up stage, with self-augmentation, as category-specific prototypes are not well-formed in the first stages of training. Our ablation experiment in Table IV validates this assumption, showing that the accuracy degrades by almost 3% when no warm-up was employed and action category prototypes were leveraged from the very beginning.

Tolerance to noisy inputs. The quality of the skeleton data itself is influenced by a variety of factors, such as sensor noise or occlusions. A larger gap between our PROFORMER model and standard DML trained on Toyota Smarthome (which is noisier than the more controlled NTU-datasets) hints towards its advantages specifically for imperfect input.

To validate if this is the case, we evaluate the model for inputs corrupted by different magnitudes of Gaussian noise and discover a remarkable tolerance of PROFORMER in this regard (Table III). While the prediction quality diminishes for DML-based models, the PROFORMER approach is much more robust when confronted with unreliable data. In particular, the performance for PROFORMER-LeViT falls from 55.94% on clean data to 51.91% for Gaussian noise with $\sigma = 0.05$, while this decline is much higher ($53.19\% \rightarrow 21.97\%$) for LeViT trained without the proposed auxiliary branch. We attribute this to the extensive learnt augmentations at the feature-level taking place in the PROFORMER auxiliary branch. The additional consistency loss encourages the model to output similar results if the embedding has been altered, which suits naturally to the use-case of noise disturbances. We compare our PROFORMER only with SL-DML for ro-

TABLE II

RESULTS FOR ONE- AND THREE-SHOT RECOGNITION FROM 3D BODY POSES ON NTU-120 [26], NTU-60 [54], AND TOYOTA SMARTHOME [30].

Encoder	NumShots	NTU-120				NTU-60				Toyota Smart Home			
		Accuracy	F1	Recall	Precision	Accuracy	F1	Recall	Precision	Accuracy	F1	Recall	Precision
CNN-based Encoder of 3D Body Poses optimized with DML (approach of [15])													
SL-DML [15]	1	49.19	47.54	49.80	49.23	54.82	54.31	56.72	54.65	58.98	27.15	27.64	35.00
SL-DML [15]	3	59.95	58.97	59.94	60.03	64.84	83.56	64.80	64.84	60.54	28.39	31.01	34.39
Skeleton-DML [40]	1	50.07	48.77	51.21	50.67	55.54	50.88	53.13	51.24	47.31	18.45	18.58	23.80
Skeleton-DML [40]	3	61.85	60.84	62.45	61.83	66.36	66.29	67.40	66.35	54.91	28.67	34.76	35.91
Visual Transformer-based Encoder of 3D Body Poses optimized with DML (ours)													
Swin [44]	1	53.13	52.09	53.48	53.16	56.99	56.24	58.67	56.99	58.76	28.83	29.17	32.34
Swin [44]	3	60.65	60.41	61.30	60.71	64.73	64.83	65.53	64.72	60.71	28.40	28.48	33.43
LeViT [45]	1	53.19	52.22	53.85	53.29	64.45	64.17	66.35	64.47	62.22	31.98	**37.56**	**35.16**
LeViT [45]	3	62.04	61.59	62.81	62.14	68.89	68.66	70.02	68.89	53.81	28.49	31.31	36.50
Visual Transformer-based Encoder with Prototype Feature Augmentation (ours)													
ProFormer (Swin)	1	54.70	53.39	54.81	54.63	58.61	57.70	59.60	58.60	**64.97**	29.42	30.96	32.27
ProFormer (Swin)	3	60.97	60.51	61.95	61.04	64.91	64.21	64.90	64.90	**77.51**	**38.48**	**38.15**	**41.07**
ProFormer (LeViT)	1	**55.94**	**54.29**	**55.80**	**56.04**	**67.67**	**67.87**	**68.74**	**67.67**	64.46	**31.91**	34.07	33.58
ProFormer (LeViT)	3	**62.62**	**62.08**	**63.01**	**62.71**	**72.47**	**72.20**	**73.25**	**72.47**	64.18	29.98	31.60	31.58

TABLE III

EFFECT OF INPUT CORRUPTION ON THE NTU-120 BENCHMARK FOR ONE-SHOT ACTIVITY RECOGNITION.

Control Condition Gaussian Noise $\sigma = 0.1, \mu = 0$				
Encoder	**Accuracy**	**F1**	**Recall**	**Prec.**
ResNet18 [57]	21.42	11.83	8.50	21.71
LeViT [45]	22.31	12.32	8.79	22.40
ProFormer (LeViT)	**52.54**	**51.16**	**51.61**	**52.65**
Control Condition Gaussian Noise $\sigma = 0.05, \mu = 0$				
Encoder	**Accuracy**	**F1**	**Recall**	**Prec.**
ResNet18 [57]	21.76	12.23	8.70	21.86
LeViT [45]	21.97	12.82	9.69	22.07
ProFormer (LeViT)	**51.91**	**50.08**	**51.67**	**52.01**

TABLE IV

EFFECT OF THE SELF-AUGMENTATION WARM-UP USED IN THE PROFORMER (ONE-SHOT RECOGNITION ON NTU-120).

ProFormer Variant	Accuracy	F1	Recall	Precision
No self-augment. warm-up	53.25	51.82	53.48	53.36
With self-augment. warm-up	**55.94**	**54.29**	**55.80**	**56.04**

TABLE V

A COMPARISON TO OTHER ENCODER ARCHITECTURE.

Methods	Accuracy	F1	Recall	Precision
SL-DML (CTR-GCN[29])	43.92	41.38	45.21	43.89
SL-DML (STTR[28])	39.56	39.45	41.92	39.58
ProFormer (LeViT)	**55.94**	**54.29**	**55.80**	**56.04**

bustness verification since the input form is the same to ours to make fair comparison, while the only difference between SL-DML and Skeleton-DML is the input form. Although resistance to noise is not the primary goal of this paper, we see it as an interesting finding and aim to explore it in depth in the future.

Impact of the temporal resolution. Lastly, we look at the role of temporal dimension by explicitly blending out portions of skeleton sequences at test-time. Our intuition is that if the model learns *movements*, local temporal changes of the skeleton positions should be important for the decision and the performance should monotonically decrease as the sequences get shorter. Figure 4 illustrates the changes in accuracy if the input stream was downsampled by different rates. For example, the downsampling rate of 2 means that we pick up the frames using temporal step as 2. In Figure 4, while the overall accuracy of PROFORMER is higher than for the reference approach [15], it consistently declines for the transformer-based network as more sequence parts are removed. This is not the case for a CNN-based approach, where the performance starts to decline only if the sequence is 10 times shorter. Surprisingly, up until the downsampling rate of 8, the CNN-based method yields a slight performance increase, indicating that the reference framework puts more weight on classification of static skeleton poses while the movement plays a less role, rather constituting additional noise. In contrast, local skeleton movements seem to be captured by PROFORMER, as the recognition quality declines with larger temporal downsampling rate. We view the learning of temporal cues as a positive property of transformer-based approaches and believe that their initial objective of dealing with sequential data [41] makes them excellent encoders of body movement.

V. CONCLUSION

To effectively assist people, robots need to accurately understand the current state of the human [10]. In this work, we tackle the problem of data-scarce recogition of daily activities, which is vital for robust and interactive assistive robots operating in dynamic and unstructured environments, where novel avtivities-of-interest may occur at any time.We operationalize and study off-the shelf well kown visual transformer architectures in the context of one-shot recognition from 3D skeletons by casting these streams of 3D coordinates of joints as image-like representations, yielding excellent performance on three datasets. Inspired by recent success of augmentation-based methods in semi-supervised learning, we further introduce the PROFORMER leveraging an additional auxiliary branch encouraging the embedder to produce similar results despite extensive augmentations at the feature-level. A key ingredient of our approach is a two-phase augmentation technique leveraging learned self-attention to alter the embedding based on either the input itself (warm-up phase) or, at the later stages, based iteratively updated embedding prototypes of activity classes. With no change of the architecture at test-time, the PROFORMER approach consistently improves performance, surpassing the best published method for skeleton-based one shot recognition by 1.84% and shows more robust performance.

REFERENCES

[1] R. Mojarad, F. Attal, A. Chibani, and Y. Amirat, "Automatic classification error detection and correction for robust human activity recognition," *RA-L*, 2020.

[2] J. D. Jones, C. Cortesa, A. Shelton, B. Landau, S. Khudanpur, and G. D. Hager, "Fine-grained activity recognition for assembly videos," *RA-L*, 2021.

[3] E. E. Aksoy, E. Ovchinnikova, A. Orhan, Y. Yang, and T. Asfour, "Unsupervised linking of visual features to textual descriptions in long manipulation activities," *RA-L*, 2017.

[4] M. M. Islam and T. Iqbal, "HAMLET: A hierarchical multimodal attention-based human activity recognition algorithm," in *IROS*, 2020.

[5] J. Jang, D. Kim, C. Park, M. Jang, J. Lee, and J. Kim, "ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly," in *IROS*, 2020.

[6] A. Roitberg, N. Somani, A. Perzylo, M. Rickert, and A. Knoll, "Multimodal human activity recognition for industrial manufacturing processes in robotic workcells," in *ICMI*, 2015.

[7] M. M. Islam and T. Iqbal, "Multi-GAT: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition," *RA-L*, 2021.

[8] C. R. Dreher, M. Wächter, and T. Asfour, "Learning object-action relations from bimanual human demonstration using graph networks," *RA-L*, 2020.

[9] N. Sünderhauf *et al.*, "The limits and potentials of deep learning for robotics," *IJRR*, 2018.

[10] E. G. Christoforou, A. S. Panayides, S. Avgousti, P. Masouras, and C. S. Pattichis, "An overview of assistive robotics and technologies for elderly care," in *MEDICON*, 2019.

[11] C. Cao, Y. Li, Q. Lv, P. Wang, and Y. Zhang, "Few-shot action recognition with implicit temporal alignment and pair similarity optimization," *CVIU*, 2021.

[12] Z. Gao, L. Guo, W. Guan, A.-A. Liu, T. Ren, and S. Chen, "A pairwise attentive adversarial spatiotemporal network for cross-domain few-shot action recognition-R2," *TIP*, 2021.

[13] J. Patravali, G. Mittal, Y. Yu, F. Li, and M. Chen, "Unsupervised few-shot action recognition via action-appearance aligned meta-adaptation," in *ICCV*, 2021.

[14] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, "Temporal-relational CrossTransformers for few-shot action recognition," in *CVPR*, 2021.

[15] R. Memmesheimer, S. Häring, N. Theisen, and D. Paulus, "Skeleton-DML: Deep metric learning for skeleton-based one-shot action recognition," *WACV*, 2022.

[16] R. Memmesheimer, N. Theisen, and D. Paulus, "SL-DML: Signal level deep metric learning for multimodal one-shot action recognition," in *ICPR*, 2021.

[17] Y. Zou, Y. Shi, Y. Wang, Y. Shu, Q. Yuan, and Y. Tian, "Hierarchical temporal memory enhanced one-shot distance learning for action recognition," in *ICME*, 2018.

[18] Y. Zou, Y. Shi, D. Shi, Y. Wang, Y. Liang, and Y. Tian, "Adaptation-oriented feature projection for one-shot action recognition," *TMM*, 2020.

[19] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViVit: A video vision transformer," in *ICCV*, 2021.

[20] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *ACPR*, 2015.

[21] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *ICMEW*, 2017.

[22] S. Zheng *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021.

[23] D. Berthelot *et al.*, "ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[24] C.-W. Kuo, C.-Y. Ma, J.-B. Huang, and Z. Kira, "FeatMatch: Feature-based augmentation for semi-supervised learning," in *ECCV*, 2020.

[25] K. Sohn *et al.*, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[26] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *TPAMI*, 2020.

[27] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *CVPR*, 2020.

[28] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *CVIU*, 2021.

[29] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *ICCV*, 2021.

[30] S. Das *et al.*, "Toyota smarthome: Real-world activities of daily living," in *ICCV*, 2019.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[32] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *ICML*, 2016.

[33] Y.-X. Wang and M. Hebert, "Learning to learn: Model regression networks for easy small sample learning," in *ECCV*, 2016.

[34] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *CVPR*, 2020.

[35] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *ICCV*, 2019.

[36] S. Reiß, A. Roitberg, M. Haurilet, and R. Stiefelhagen, "Activity-aware attributes for zero-shot driver behavior recognition," in *CVPRW*, 2020.

[37] A. Roitberg, M. Martinez, M. Haurilet, and R. Stiefelhagen, "Towards a fair evaluation of zero-shot action recognition using external data," in *ECCVW*, 2018.

[38] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *TPAMI*, 2018.

[39] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *CVPR*, 2017.

[40] R. Memmesheimer, S. Häring, N. Theisen, and D. Paulus, "Skeleton-DML: Deep metric learning for skeleton-based one-shot action recognition," in *WACV*, 2022.

[41] A. Vaswani *et al.*, "Attention is all you need," in *NeurIPS*, 2017.

[42] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[43] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021.

[44] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[45] B. Graham *et al.*, "LeViT: a vision transformer in ConvNet's clothing for faster inference," in *ICCV*, 2021.

[46] Q. Zhang and Y. Yang, "ResT: An efficient transformer for visual recognition," *arXiv preprint arXiv:2105.13677*, 2021.

[47] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[48] Z. Liu *et al.*, "Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021.

[49] Y. Zhang *et al.*, "VidTr: Video transformer without convolutions," in *ICCV*, 2021.

[50] R. Bai *et al.*, "GCsT: Graph convolutional skeleton transformer for action recognition," *arXiv preprint arXiv:2109.02860*, 2021.

[51] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *ICPRW*, 2021.

[52] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "SkeleMotion: A new representation of skeleton joint sequences based on motion information for 3D action recognition," in *AVSS*, 2019.

[53] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *CVPR*, 2019.

[54] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *CVPR*, 2016.

[55] A. Sabater, L. Santos, J. Santos-Victor, A. Bernardino, L. Montesano, and A. C. Murillo, "One-shot action recognition in challenging therapy scenarios," in *CVPRW*, 2021.

[56] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, 2012.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.