STMT: A Spatial-Temporal Mesh Transformer for MoCap-Based Action Recognition

Xiaoyu Zhu[1], Po-Yao Huang[2†], Junwei Liang[3†], Celso M. de Melo[4], Alexander Hauptmann[1]
[1]Carnegie Mellon University, [2]FAIR, Meta AI, [3]HKUST (Guangzhou), [4]DEVCOM Army Research Laboratory

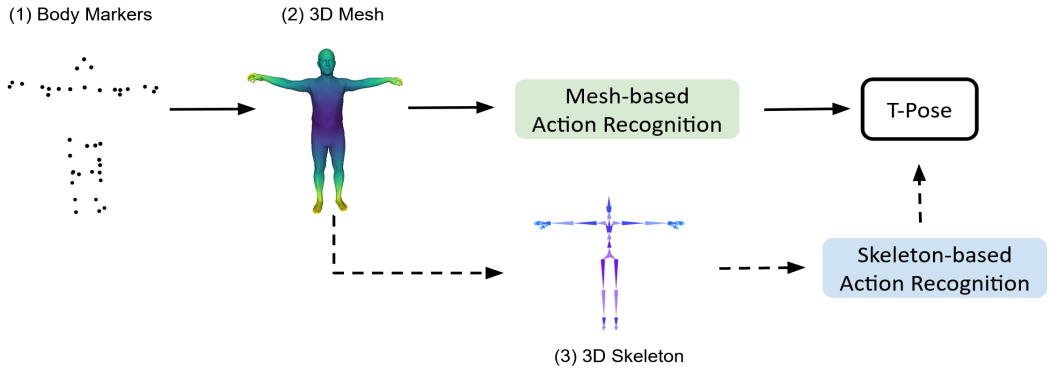

Figure 1. Current state-of-the-art MoCap-based action recognition methods first convert body markers into a human body mesh, which is used to predict a standardized 3D skeleton. The 3D skeleton is used as input for action recognition models (dashed line). We propose a method that directly models the dynamics of raw mesh sequences (solid line). Our method saves the manual effort to derive skeleton representation, and achieves superior recognition performance by leveraging surface motion and body shape knowledge from meshes.

Abstract

We study the problem of human action recognition using motion capture (MoCap) sequences. Unlike existing techniques that take multiple manual steps to derive standardized skeleton representations as model input, we propose a novel Spatial-Temporal Mesh Transformer (STMT) to directly model the mesh sequences. The model uses a hierarchical transformer with intra-frame off-set attention and inter-frame self-attention. The attention mechanism allows the model to freely attend between any two vertex patches to learn non-local relationships in the spatial-temporal domain. Masked vertex modeling and future frame prediction are used as two self-supervised tasks to fully activate the bi-directional and auto-regressive attention in our hierarchical transformer. The proposed method achieves state-of-the-art performance compared to skeleton-based and point-cloud-based models on common MoCap benchmarks. Code is available at https://github.com/zgzxy001/STMT.

1. Introduction

Motion Capture (MoCap) is the process of digitally recording the human movement, which enables the fine-grained capture and analysis of human motions in 3D space [40, 50]. MoCap-based human perception serves as key elements for various research fields, such as action recognition [15, 46–48, 50, 57], tracking [47], pose estimation [1, 27], imitation learning [76], and motion synthesis [47]. Besides, MoCap is one of the fundamental technologies to enhance human-robot interactions in various practical scenarios including hospitals and manufacturing environment [22, 28, 41, 43, 45, 77]. For example, Hayes [22] classified automotive assembly activities using MoCap data of humans and objects. Understanding human behaviors from MoCap data is fundamentally important for robotics perception, planning, and control.

Skeleton representations are commonly used to model MoCap sequences. Some early works [3, 29] directly used body markers and their connectivity relations to form a skeleton graph. However, the marker positions depend on each subject (person), which brings sample variances within each dataset. Moreover, different MoCap datasets usually have different numbers of body markers. For example, ACCAD [48], BioMotion [64], Eyes Japan [15], and KIT [42] have 82, 41, 37, and 50 body markers respectively. This prevents the model to be trained and tested on a unified framework. To use standard skeleton representa-

†Equal Contribution.

tions such as NTU RGB+D [58], Punnakkal *et al.* [50] first used Mosh++ to fit body markers into SMPL-H meshes, and then predicted a 25-joint skeleton [33] from the mesh vertices [54]. Finally, a skeleton-based model [60] was used to perform action recognition. Although those methods achieved advanced performance, they have the following disadvantages. First, they require several manual steps to map the vertices from mesh to skeleton. Second, skeleton representations lose the information provided by original MoCap data (*i.e.*, surface motion and body shape knowledge). To overcome those disadvantages, we propose a mesh-based action recognition method to directly model dynamic changes in raw mesh sequences, as illustrated in Figure 1.

Though mesh representations provide fine-grained body information, it is challenging to classify high-dimensional mesh sequences into different actions. First, unlike structured 3D skeletons which have joint correspondence across frames, there is no vertex-level correspondence in meshes (*i.e.*, the vertices are unordered). Therefore, the local connectivity of every single mesh can not be directly aggregated in the temporal dimension. Second, mesh representations encode local connectivity information, while action recognition requires global understanding in the whole spatial-temporal domain.

To overcome the aforementioned challenges, we propose a novel Spatial-Temporal Mesh Transformer (*STMT*). *STMT* leverages mesh connectivity information to build patches at the frame level, and uses a hierarchical transformer which can freely attend to any intra- and inter-frame patches to learn spatial-temporal associations. The hierarchical attention mechanism allows the model to learn patch correlation across the entire sequence, and alleviate the requirement of explicit vertex correspondence. We further define two self-supervised learning tasks, namely masked vertex modeling and future frame prediction, to enhance the global interactions among vertex patches. To reconstruct masked vertices of different body parts, the model needs to learn prior knowledge about the human body in the spatial dimension. To predict future frames, the model needs to understand meaningful surface movement in the temporal dimension. To this end, our hierarchical transformer pretrained with those two objectives can further learn spatial-temporal context across entire frames, which is beneficial for the downstream action recognition task.

We evaluate our model on common MoCap benchmark datasets. Our method achieves state-of-the-art performance compared to skeleton-based and point-cloud-based models. The contributions of this paper are three-fold:

- We introduce a new hierarchical transformer architecture, which jointly encodes intrinsic and extrinsic representations, along with intra- and inter-frame attention, for spatial-temporal mesh modeling.
- We design effective and efficient pretext tasks, namely masked vertex modeling and future frame prediction, to enable the model to learn from the spatial-temporal global context.
- Our model achieves superior performance compared to state-of-the-art point-cloud and skeleton models on common MoCap benchmarks.

2. Related Work

Action Recognition from Depth and Point Cloud. 3D action recognition models have achieved promising performance with depth [34, 55, 56, 68, 71] and point clouds [18,36,52,70]. Depth provides reliable 3D structural and geometric information which characterizes informative human actions. In MVDI [71], dynamic images [4] were extracted through multi-view projections from depth videos for 3D action recognition. 3D-FCNN [55] directly exploited a 3D-CNN to model depth videos. Another popular category of 3D human action recognition is based on 3D point clouds. PointNet [51] and PointNet++ [52] are the pioneering works contributing towards permutation invariance of 3D point sets for representing 3D geometric structures. Along this avenue, MeteorNet [36] stacked multi-frame point clouds and aggregates local features for action recognition. 3DV [70] transferred point cloud sequences into regular voxel sets to characterize 3D motion compactly via temporal rank pooling. PSTNet [18] disentangled space and time to alleviate point-wise spatial variance across time. Action recognition has shown promising results with 3D skeletons and point clouds. Meshes, which are commonly used in representing human bodies and creating action sequences, have not been explored for the action recognition task. In this work, we propose the first mesh-based action recognition model.

MoCap-Based Action Recognition. Motion-capture (MoCap) datasets [15, 44, 46–48, 50, 57] serve as key elements for various research fields, such as action recognition [15, 44, 46–48, 50, 57], tracking [47], pose estimation [1, 27], imitation learning [76], and motion synthesis [47]. MoCap-based action recognition was formulated as a skeleton-based action recognition problem [50]. Various architectures have been investigated to incorporate skeleton sequences. In [14, 35, 75], skeleton sequences were treated as time-series inputs to RNNs. [24,69] respectively transformed skeleton sequences into spectral images and trajectory maps and then adopted CNNs for feature learning. In [72], Yan *et al*. leveraged GCN to model joint dependencies that can be naturally represented with a graph. In this paper, we propose a novel method to directly model the dynamics of raw mesh sequences which can benefit from surface motion and body shape knowledge.

Masked Autoencoder. Masked autoencoder has gained attention in Natural Language Processing and Computer Vision to learn effective representations using auto-encoding. Stacked denoising autoencoders [66] treated masks as a noise type and used denoising autoencoders to denoise corrupted inputs. ViT [13] proposed a self-supervised pre-training task to reconstruct masked tokens. More recently, BEiT [2] proposed to learn visual representations by reconstructing the discrete tokens [53]. MAE [23] proposed a simple yet effective asymmetric framework for masked image modeling. In 3D point cloud analysis, Wang *et al.* [67] chose to first generate partial point clouds by calculating occlusion from random camera viewpoints, and then completed occluded point clouds using autoencoding. Point-BERT [73] followed the success of BERT [12] to predict the masked tokens learned from points. However, applying self-supervised learning to temporal 3D sequences (*i.e.* point cloud, 3D skeleton) has not been fully explored. One probable reason is that self-supervised learning on high-dimensional 3D temporal sequences is computationally-expensive. In this work, we propose an effective and efficient self-supervised learning method based on masked vertex modeling and future frame prediction.

3. Method

3.1. Overview

In this section, we describe our model for mesh-based action recognition, which we call *STMT*. The inputs of our model are temporal mesh sequences: $\mathbf{M} = ((\mathbf{P}_1, \mathbf{A}_1), (\mathbf{P}_2, \mathbf{A}_2), \cdots, (\mathbf{P}_t, \mathbf{A}_t))$, where t is the frame number. $\mathbf{P}_i \in \mathbb{R}^{N\times 3}$ represents the vertex positions in Cartesian coordinates, where N is the number of vertices. $\mathbf{A}_i \in \mathbb{R}^{N\times N}$ represents the adjacency matrix of the mesh. Element $\mathbf{A}_i^{mn} \in \mathbf{A}_i$ is one when there is an edge from vertex V_m to vertex V_n, and zero when there is no edge. The mesh representation with vertices and their adjacent matrix is a unified format for various body models such as SMPL [39], SMPL-H [54], and SMPL-X [49]. In this work, we use SMPL-H body models from AMASS [40] to obtain the mesh sequences, but our method can be easily adapted to other body models.

Mesh's local connectivity provides fine-grained information. Previous methods [21, 59] proved that explicitly using surface (*e.g.*, mesh) connectivity information can achieve higher accuracy in shape classification and segmentation tasks. However, classifying temporal mesh sequences is a more challenging problem, as there is no vertex-level correspondence across frames. This prevents graph-based models from directly aggregating vertices in the temporal dimension. Therefore, we propose to first leverage mesh connectivity information to build patches at the frame level, then use a hierarchical transformer which can freely attend to any intra- and inter-frame patches to learn spatial-temporal associations. In summary, it has the following key components:

- **Surface Field Convolution** to form local vertex patches by considering both intrinsic and extrinsic mesh representations.
- **Hierarchical Spatial-Temporal Transformer** to learn spatial-temporal correlations of vertex patches.
- **Self-Supervised Pre-Training** to learn the global context in terms of appearance and motion.

See Figure 2 for a high-level summary of the model, and the sections below for more details.

3.2. Surface Field Convolution

Because displacements in grid data are regular, traditional convolutions can directly learn a kernel for elements within a region. However, mesh vertices are unordered and irregular. Considering the special mesh representations, we represent each vertex by encoding features from its neighbor vertices inspired by [51,52]. To fully utilize meshes' local connectivity information, we consider the mesh properties of extrinsic curvature of submanifolds and intrinsic curvature of the manifold itself. Extrinsic curvature between two vertices is approximated using Euclidean distance. Intrinsic curvature is approximated using Geodesic distance, which is defined as the shortest path between two vertices on mesh surfaces. We propose a light-weighted surface field convolution to build local patches, which can be denoted as:

$$\boldsymbol{F}_{VG}^{\prime(x,y,z)} = \sum_{(\delta_x,\delta_y,\delta_z)\in G(x,y,z)} \boldsymbol{W}^{(\delta_x,\delta_y,\delta_z)} \cdot \boldsymbol{F}^{(x+\delta_x,y+\delta_y,z+\delta_z)} \tag{1}$$

$$\boldsymbol{F}_{VE}^{\prime(x,y,z)} = \sum_{(\zeta_x,\zeta_y,\zeta_z)\in E(x,y,z)} \boldsymbol{W}^{(\zeta_x,\zeta_y,\zeta_z)} \cdot \boldsymbol{F}^{(x+\zeta_x,y+\zeta_y,z+\zeta_z)} \tag{2}$$

G and E is the local region around vertex (x, y, z). In this paper, we use k-nearest-neighbor to sample local vertices. $(\delta_x, \delta_y, \delta_z)$ and $(\zeta_x, \zeta_y, \zeta_z)$ represent the spatial displacement in geodesic and euclidean space, respectively. $\boldsymbol{F}^{(x,y,z)}$ denotes the feature of the vertex at position (x, y, z).

3.3. Hierarchical Spatial-Temporal Transformer

We propose a hierarchical transformer that consists of intra-frame and inter-frame attention. The basic idea behind our transformer is three-fold: (1) Intra-frame attention can encode connectivity information from the adjacency matrix, while such information can not be directly

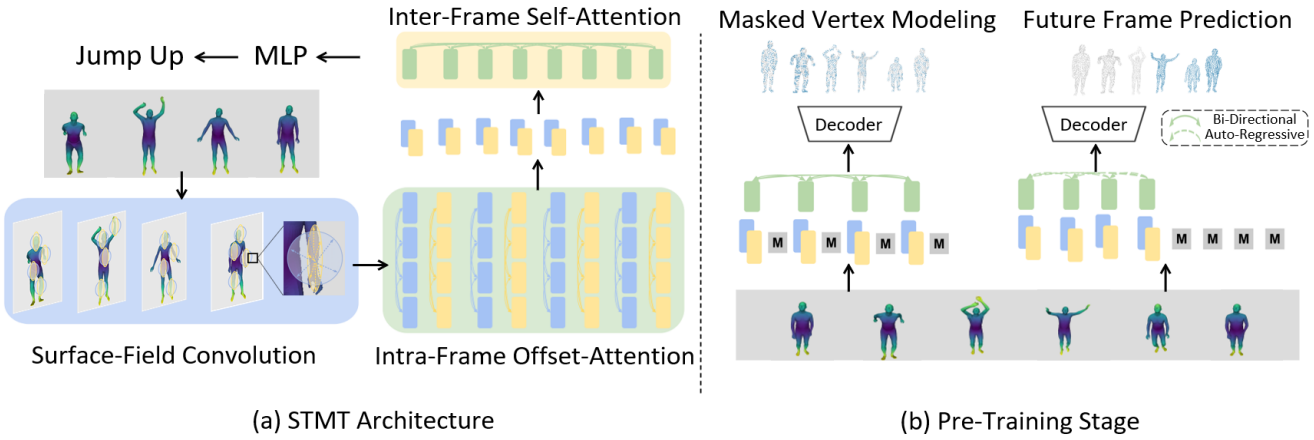

Figure 2. Overview of the proposed framework. **(a) Overview of *STMT*.** Given a mesh sequence, we first develop vertex patches by extracting both intrinsic (geodesic) and extrinsic (euclidean) features using surface field convolution. The intrinsic and extrinsic features are denoted by yellow and blue blocks respectively. Those patches are used as input to the intra-frame offset-attention network to learn appearance features. Then we concatenate intrinsic patches and extrinsic patches of the same position. The concatenated vertex patches (green blocks) are fed into the inter-frame self-attention network to learn spatial-temporal correlations. Finally, the local and global features are mapped into action predictions by MLP layers. **(b) Overview of Pre-Training Stage.** We design two pretext tasks: masked vertex modeling and future frame prediction for global context learning. Bidirectional attention is used for the reconstruction of masked vertices. Auto-regressive attention is used for the future frame prediction task.

aggregated in the temporal domain because vertices are unordered. (2) Frame-level offset-attention can be used to mimic the Laplacian operator to learn effective spatial representations. (3) Inter-frame self-attention can learn feature correlations in the spatial-temporal domain.

3.3.1 Intra-Frame Offset-Attention

Graph convolution networks [6] show the benefits of using a Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{E}$ to replace the adjacency matrix $\mathbf{E}$, where $\mathbf{D}$ is the diagonal degree matrix. Inspired by this, offset-attention has been proposed and achieved superior performance in point-cloud classification and segmentation tasks [20]. We adapt offset-attention to attend to vertex patches. Specifically, the offset-attention layer calculates the offset (difference) between the self-attention (SA) features and the input features by element-wise subtraction. Offset-attention is denoted as:

$$\boldsymbol{F}_{out} = OA(\boldsymbol{F}_{in}) = \phi(\boldsymbol{F}_{in} - \boldsymbol{F}_{sa}) + \boldsymbol{F}_{in}. \quad (3)$$

where ϕ denotes a non-linear operator. $\boldsymbol{F}_{in} - \boldsymbol{F}_{sa}$ is proved to be analogous to discrete Laplacian operator [20], *i.e.* $\boldsymbol{F}_{in} - \boldsymbol{F}_{sa} \approx \boldsymbol{L}\boldsymbol{F}_{in}$. As Laplacian operators in geodesic and euclidean space are expected to be different, we propose to use separate transformers to model intrinsic patches and extrinsic patches. Specifically, the aggregated feature for vertex V is denoted as:

$$\boldsymbol{F}_V'^{(x,y,z)} = OA_G(\boldsymbol{F}_{VG}'^{(x,y,z)}) \oplus OA_E(\boldsymbol{F}_{VE}'^{(x,y,z)}) \quad (4)$$

Here $F_{VG}'^{(x,y,z)} \in \mathbb{R}^{N\times d_g}$ and $F_{VE}'^{(x,y,z)} \in \mathbb{R}^{N\times d_e}$ are local patches learned using Equ. 1 and Equ. 2. $F_V'^{(x,y,z)} \in \mathbb{R}^{N\times d}$ denotes the local patch for position (x, y, z), where $d = d_g + d_e$. The weights of OA_G and OA_E are not shared.

3.3.2 Inter-Frame Self-Attention

Given F_V' which encodes local connectivity information, we use self-attention (SA) [65] to learn semantic affinities between different vertex patches across frames. Specifically, let $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ be the *query, key* and *value*, which are generated by applying linear transformations to the input features $F_V' \in \mathbb{R}^{N\times d}$ as follows:

$$\begin{aligned} (\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= F_V' \cdot (\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v) \\ \boldsymbol{Q}, \boldsymbol{K} &\in \mathbb{R}^{N\times d_a}, \quad \boldsymbol{V} \in \mathbb{R}^{N\times d} \\ \boldsymbol{W}_q, \boldsymbol{W}_k &\in \mathbb{R}^{d\times d_a}, \quad \boldsymbol{W}_v \in \mathbb{R}^{d\times d} \end{aligned} \quad (5)$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$ are the shared learnable linear transformation, and d_a is the dimension of the query and key vectors. Then we can use the query and key matrices to calculate the attention weights via the matrix dot-product:

$$\boldsymbol{A} = (\tilde{\alpha})_{i,j} = \text{softmax}(\frac{\boldsymbol{Q} \cdot \boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_a}}). \quad (6)$$

$$\boldsymbol{F}_{sa} = \boldsymbol{A} \cdot \boldsymbol{V} \quad (7)$$

The self-attention output features $\boldsymbol{F}_{sa}$ are the weighted sums of the value vector using the corresponding attention weights. Specifically, for a vertex patch in position (x, y, z),

its aggregated feature after inter-frame self-attention can be computed as: $\boldsymbol{F}_{sa}^{(x,y,z)} = \sum \boldsymbol{A}^{(x,y,z),(x',y',z')} \times \boldsymbol{V}^{(x',y',z')}$, where (x', y', z') belongs to the Cartesian coordinates of $\boldsymbol{F}'_V$.

3.4. Self-Supervised Pre-Training

Self-supervised learning has achieved remarkable results on large-scale image datasets [23]. However, self-supervised learning for temporal 3D sequences (*i.e.* point cloud, 3D skeleton) remains to be challenging and has not been fully explored. There are two possible reasons: (1) self-supervised learning methods rely on large-scale datasets to learn meaningful patterns [10]. However, existing MoCap benchmarks are relatively small compared to 2D datasets like ImageNet [11]. (2) Self-supervised learning for 3D data sequences is computationally expensive in terms of memory and speed. In this work, we first propose a simple and effective method to augment existing MoCap sequences, and then define two effective and efficient self-supervised learning tasks, namely masked vertex modeling and future frame prediction, which enable the model to learn global context. The work that is close to us is OcCO [67], which proposed to use occluded point cloud reconstruction as the pretext task. OcCO has a computationally-expensive process to generate occlusions, including point cloud projection, occluded point calculation, and a mapping step to convert camera frames back to world frames. Different from OcCO, we randomly mask vertex patches or future frames on the fly, which saves the pre-processing step. Moreover, our pre-training method is designed for temporal mesh sequences and considers both bi-directional and auto-regressive attention.

3.4.1 Data Augmentation through Joint Shuffle

Considering the flexibility of SMPL-H representations, we propose a simple yet effective approach to augment SMPL-H sequences by shuffling body pose parameters. Specifically, we split SMPL-H pose parameters into five body parts: bone, left/right arm, and left/right leg. We use $I_{bone}, I_{leg}^{left}, I_{leg}^{right}, I_{arm}^{left}, I_{arm}^{right}$ to denote the SMPL-H pose indexes of the five body parts. Then we synthesize new sequences by randomly selecting body parts from five different sequences. We keep the temporal order for each part such that the merged action sequences have meaningful motion trajectories. Pseudocode for the joint shuffle is provided in Algorithm 1. The input to Joint Shuffle are SMPL-H pose parameters $\theta \in \mathbb{R}^{b\times t\times n\times 3}$, where b is the sequence number, t is the frame number, and n is the joint number. We randomly select the shape β and dynamic parameters ϕ from one of the five SMPL-H sequences to compose a new SMPL-H body model. Given b SMPL-H sequences, we can synthesize ${}^{b}C_5 = \frac{b!}{5!(b-5)!}$ number of new sequences. We prove that the model can benefit from large-scale pre-training in Section 4.6.

Algorithm 1: Pseudocode of STMT Joint Shuffle

function STMT_JOINT_SHUFFLE($\theta \in \mathbb{R}^{b\times t\times n\times 3}, I_{bone}, I_{leg}^{left}, I_{leg}^{right}, I_{arm}^{left}, I_{arm}^{right}$)
  $\theta_s \leftarrow random\_sample(\theta, 5)$ ▷ $\theta_s \in \mathbb{R}^{5\times t\times n\times 3}$, randomly sample five SMPL-H sequences
  $t_{max} \leftarrow get\_max\_length(\theta_s)$ ▷ compute the maximum sequence length in θ_s
  $\theta_{new} \leftarrow Initialize(t_{max}, n, 3)$
  $P \leftarrow \{I_{bone}, I_{leg}^{left}, I_{leg}^{right}, I_{arm}^{left}, I_{arm}^{right}\}$
  for i in $0, 1, 2, 3, 4$ **do**
    $\theta_s \leftarrow repeat(\theta_s[i], (t_{max}, n, 3))$ ▷ pad each sequence to the max length using repeating
    $\theta_{new}[P[i]] \leftarrow \theta_s[i][P[i]]$ ▷ assign the body-part sequence
  return θ_{new}

3.4.2 Masked Vertex Modeling with Bi-Directional Attention

To fully activate the inter-frame bi-directional attention in the transformer, we design a self-supervised pretext task named Masked Vertex Modeling (MVM). The model can learn human prior information in the spatial dimension by reconstructing masked vertices of different body parts. We randomly mask r percentages of the input vertex patches, and force the model to reconstruct the full sequences. Moreover, we use bi-directional attention to learn correlations among all remaining local patches. Each patch will attend to all patches in the entire sequence. It models the joint distribution of vertex patches over the whole temporal sequences x as the following product of conditional distributions, where x_i is a single vertex patch:

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, .., x_i, ..., x_N). \tag{8}$$

Where N is the number of patches in the entire sequence x after masking. Every patch will attend to all patches in the entire sequence. In this way, bi-directional attention is fully-activated to learn spatial-temporal features that can accurately reconstruct completed mesh sequences.

3.4.3 Future Frame Prediction with Auto-Regressive Attention

The masked vertex modeling task is to reconstruct masked vertices in different body parts. The model can reconstruct completed mesh sequences if it captures the human body prior or can make a movement inference from nearby frames. As action recognition requires the model to understand the global context, we propose the future frame

prediction (FFP) task. Specifically, we mask out all the future frames and force the transformer to predict the masked frames. Moreover, we propose to use auto-regressive attention for the future frame prediction task, inspired by language generation models like GPT-3 [5]. However, directly using RNN-based models [9] in GPT-3 to predict future frames one by one is inefficient, as 3D mesh sequences are denser compared to language sequences. Therefore, we propose to reconstruct all future frames in a single forward pass. For auto-regressive attention, we model the joint distribution of vertex patches over a mesh sequence x as the following product of conditional distributions, where x_i is a single patch at frame t_i:

$$p(x) = \prod_{i=1}^{N} p(x_i | x_1, x_2, ..., x_M). \tag{9}$$

Where N is the number of patches in the entire sequence x after masking. $M = (t_i - 1) \times n$, where n is the number of patches in a single frame. Each vertex patch depends on all patches that are temporally before it. The auto-regressive attention enables the model to predict movement patterns and trajectories, which is beneficial for the downstream action recognition task.

3.5. Training

In the pre-training stage, we use PCN [74] as the decoder to reconstruct masked vertices and predict future frames. The decoder is shared for the two pretext tasks. Since mesh vertices are unordered, the reconstruction loss and future prediction loss should be permutation-invariant. Therefore, we use Chamfer Distance (CD) as the loss function to measure the difference between the model predictions and ground truth mesh sequences.

$$CD(M_{pred}, M_{gt}) = \frac{1}{|M_{pred}|} \sum_{x \in M_{pred}} \min_{y \in M_{gt}} \|x - y\|_2 + \frac{1}{|M_{gt}|} \sum_{y \in M_{gt}} \min_{x \in M_{pred}} \|y - x\|_2 \tag{10}$$

CD (10) calculates the average closest euclidean distance between the predicted mesh sequences M_{pred} and the ground truth sequences M_{gt}. The overall loss is a weighted sum of masked vertex reconstruction loss and future frame prediction loss:

$$L = \lambda_1 CD(M_{pred}^{MVM}, M_{gt}) + \lambda_2 CD(M_{pred}^{FFP}, M_{gt}) \tag{11}$$

In the fine-tuning stage, we replace the PCN decoder with an MLP head. Cross-entropy loss is used for model training.

4. Experiment

4.1. Datasets

Following previous MoCap-based action recognition methods [50,63], we evaluate our model on the most widely used benchmarks: KIT [42] and BABEL [50]. **KIT** is one of the largest MoCap datasets. It has 56 classes with 6,570 sequences in total. (2) **BABEL** is the largest 3D MoCap dataset that unifies 15 different datasets. BABEL has 43 hours of MoCap data performed by over 346 subjects. We use the 60-class subset from BABEL, which contains 21,653 sequences with single-class labels. We randomly split each dataset into training, test, and validation set, with ratios of 70%, 15%, and 15%, respectively. Note that existing action recognition datasets with skeletons only are not suitable for our experiments, as they do not provide full 3D surfaces or SMPL parameters to obtain the mesh representation.

Motion Representation. Both KIT and BABEL's MoCap sequences are obtained from AMASS dataset in SMPL-H format. A MoCap sequence is an array of pose parameters over time, along with the shape and dynamic parameters. For skeleton-based action recognition, we follow previous work [50] which predicted the 25-joint skeleton from the vertices of the SMPL-H mesh. The movement sequence is represented as $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_L)$, where $\mathbf{x}_i \in \mathbb{R}^{J \times 3}$ represents the position of the J joints in the skeleton in Cartesian coordinates. For point-cloud-based action recognition, we directly use the vertices of SMPL-H model as the model input. The point-cloud sequence is represented as $\mathbf{P} = (\mathbf{p}_1, \cdots, \mathbf{p}_L)$, where $\mathbf{p}_i \in \mathbb{R}^{V \times 3}$, and V is the number of vertices. For mesh-based action recognition, we represent the motion as a series of mesh vertices and their adjacent matrix over time, as introduced in Section 3.1. See Sup. Mat. for more details about datasets and pre-processing.

4.2. Baseline Methods

We compare our model with state-of-the-art 3D skeleton-based and point cloud-based action recognition models, as there is no existing literature on mesh-based action recognition. 2s-AGCN [61], CTR-GCN [7], and MS-G3D [37] are used as skeleton-based baselines. Among those methods, 2s-AGCN trained with focal loss and cross-entropy loss are used as benchmark methods in the BABEL dataset [50]. For the comparison with point-cloud baselines, we choose PSTNet [18], SequentialPointNet [30], and P4Transformer [16]. Those methods achieved top performance on common point-cloud-based action recognition benchmarks.

4.3. Implementation Details

For skeleton-based baselines, we use the official implementations of 2s-ACGN, CTR-GCN, and MS-G3D from

Method	Input	KIT		BABEL-60	
		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
2s-AGCN-FL [61] (CVPR'19)	3D Skeleton	42.44	75.60	49.62	79.12
2s-AGCN-CE [61] (CVPR'19)	3D Skeleton	57.46	81.54	63.57	86.77
CTR-GCN [7] (ICCV'21)	3D Skeleton	64.65	87.90	67.30	88.50
MS-G3D [37] (CVPR'20)	3D Skeleton	65.38	87.90	67.43	87.99
PSTNet [18] (ICLR'21)	Point Cloud	56.93	88.21	61.94	84.11
SequentialPointNet [30] (arXiv'21)	Point Cloud	59.75	88.01	62.92	84.58
P4Transformer [16] (CVPR'21)	Point Cloud	62.15	88.01	63.54	86.55
STMT(Ours)	Mesh	**65.59**	**90.09**	**67.65**	**88.68**

Table 1. Experimental Results on KIT and BABEL Dataset.

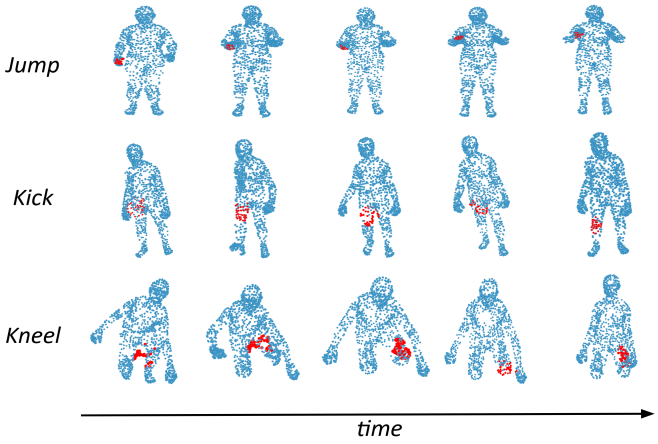

Figure 3. Visualization of inter-frame attention. Red denotes the highest attention.

[62], [8], and [38]. For point-cloud-based baselines, we use the official implementations of PSTNet, SequentialPointNet, P4Transformer from [19], [31], and [17]. We pre-train *STMT* for 200 epochs with a batch size of 32. The model is fine-tuned for 50 epochs with a batch size of 64. Adam optimizer [26] is used with a learning rate of 0.0001 for both pre-training and fine-tuning. See Sup. Mat. for more implementation details.

4.4. Main Results

Comparison with State-of-the-Art Methods. As indicated in Table 1, *STMT* outperforms all other state-of-the-art models. Our model can outperform point-cloud-based models by 3.44% and 4.11% on KIT and BABEL datasets in terms of top-1 accuracy. Moreover, compared to skeleton-based methods which involve manual efforts to convert mesh vertices to skeleton representations, our model achieves better performance by directly modeling the dynamics of raw mesh sequences.

We visualize the inter-frame attention weights of our hierarchical transformer in Figure 3. We observe that the model can pay attention to key regions across frames. This supports the intuition that our hierarchical transformer can take the place of explicit vertex tracking by learning spatial-temporal correlations.

Intrinsic	Extrinsic	MVM	FFP	Top-1 (%)
✓				63.40
✓	✓			64.03
✓	✓	✓		64.96
✓	✓		✓	64.13
✓	✓	✓	✓	**65.59**

Table 2. Performance of ablated versions. Intrinsic and Extrinsic stand for the intrinsic (geodesic) and extrinsic (euclidean) features in surface field convolution. MVM stands for Masked Vertex Modelling. FFP stands for Future Frame Prediction.

4.5. Ablation Study

Ablation Study of *STMT*. We test various ablations of our model on the KIT dataset to substantiate our design decisions. We report the results in Table 2. Note that Joint Shuffle is used in all of the self-supervised learning experiments (last three rows). We observe that each component of our model gains consistent improvements. The comparison of the first two rows proves the effectiveness of encoding both intrinsic and extrinsic features in vertex patches. Comparing the last three rows with the second row, we observe a consistent improvement using self-supervised pre-training. Moreover, the downstream task can achieve better performance with MVM compared to FFP. One probable reason is that the single task for future frame prediction is more challenging than masked vertex modeling, as the model can only see the person movement in the past. The model can achieve the best performance with both MVM and FFP, which demonstrates that the two self-supervised tasks are supplementary to each other.

4.6. Analysis

Different Pre-Training Strategies. We pre-train our model with different datasets and summarize the results in Table 3. The first row shows the case without pre-training. The second shows the result for the model pre-trained on the KIT dataset (without Joint Shuffle augmentation). The third shows the result for the model pre-trained on KIT dataset (with Joint Shuffle). We observe our model can achieve better performance with Joint Shuffle, as it can synthesize large-scale mesh sequences.

Method	Top-1 (%)
w/o pre-training	64.03
pre-training w/o JS	64.13
pre-training w/ JS	**65.59**

Table 3. Comparison of Different Pre-Training Strategies. JS stands for Joint Shuffle.

r	Pre-Train Loss ($\times 10^4$)	Fine-Tune Accuracy (%)
0.1	0.39	64.44
0.3	0.41	64.55
0.5	0.40	**65.59**
0.7	0.43	64.19
0.9	0.48	65.07
Rand	0.43	64.75

Table 4. Effect of Different Masking Ratios.

Different Masking Ratios. We investigate the impact of different masking ratios. We report the converged pre-training loss and the fine-tuning top-1 classification accuracy on the test set in Table 4. We also experiment with the random masking ratio in the last row. For each forward pass, we randomly select one masking ratio from 0.1 to 0.9 with step 0.1 to mimic flexible masked token length. The model with a random masking ratio does not outperform the best model that is pre-trained using a single ratio (*i.e.* 0.5). We observe that as the masking ratio increases, the pre-training loss mostly increases as the task becomes more challenging. However, a challenging self-supervised learning task does not necessarily lead to better performance. The model with a masking ratio of 0.7 and 0.9 have a high pre-train loss, while the fine-tune accuracy is not higher than the model with a 0.5 masking ratio. The conclusion is similar to the comparison of MVM and FFP training objectives, where a more challenging self-supervised learning task may not be optimal.

Different Number of Mesh Sequences for Pre-Training. We test the effect of different numbers of mesh sequences used in pre-training. We report the fine-tuning top-1 classification accuracy in Figure 4. We observe that a large number of pre-training data can bring substantial performance improvement. The proposed Joint Shuffle method can greatly enlarge the dataset size without any manual cost, and has the potential to further improve model performance.

Experimental Results on Noisy Body Pose Estimations. Body pose estimation has been a popular research field [25, 27, 32], but how to leverage the estimated 3D meshes for downstream perception tasks has not been fully explored. We apply the state-of-the-art body pose estimation model VIBE [27] on videos of NTU RGB+D dataset to obtain 3D mesh sequences. Skeleton and point cloud representations are derived from the estimated meshes to train the baseline

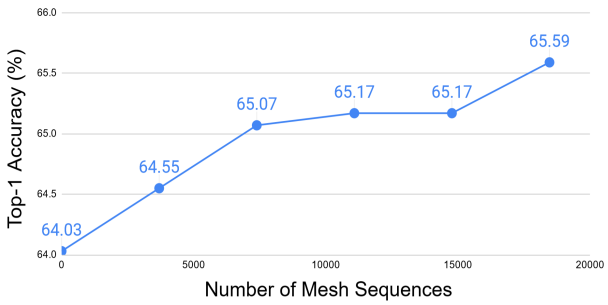

Figure 4. Effect of Different Number of Mesh Sequences.

Method	Input	Top-1 (%)
2s-AGCN-FL [61]	3D Skeleton	58.67
2s-AGCN-CE [61]	3D Skeleton	57.49
CTR-GCN [7]	3D Skeleton	62.25
MS-G3D [37]	3D Skeleton	60.01
PSTNet [18]	Point Cloud	51.48
SequentialPointNet [30]	Point Cloud	60.60
P4Transformer [16]	Point Cloud	57.84
STMT(Ours)	Mesh	**64.04**

Table 5. Experimental results on body poses estimated by VIBE [27] on NTU RGB+D dataset. The skeleton, point cloud, and mesh representations are derived from the same noisy body estimations.

models (see Sup. Mat.). We report the results in Table 5. We observe that *STMT* can outperform the best skeleton-based and point cloud-based action recognition model by 1.79% and 3.44% respectively. This shows that *STMT* with meshes as input, is more robust to input noise compared to other state-of-the-art methods with 3D skeletons or point clouds as input.

5. Conclusion

In this work, we propose a novel approach for MoCap-based action recognition. Unlike existing methods that rely on skeleton representation, our proposed model directly models the raw mesh sequences. Our method encodes both intrinsic and extrinsic features in vertex patches, and uses a hierarchical transformer to freely attend to any two vertex patches in the spatial and temporal domain. Moreover, two self-supervised learning tasks, namely Masked Vertex Modeling and Future Frame Prediction are proposed to enforce the model to learn global context. Our experiments show that *STMT* can outperform state-of-the-art skeleton-based and point-cloud-based models.

Acknowledgment

This work was supported by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-5-0003. This work used Bridges-2 GPU resources provided by PSC through allocation CIS220012 from the ACCESS program.

References

[1] Felix Achilles, Alexandru-Eugen Ichim, Huseyin Coskun, Federico Tombari, Soheyl Noachtar, and Nassir Navab. Patient mocap: Human pose estimation under blanket occlusion for hospital monitoring applications. In *International conference on medical image computing and computer-assisted intervention*, pages 491–499. Springer, 2016. 1, 2

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3

[3] Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. *Pattern Recognition*, 47(1):238–247, 2014. 1

[4] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3034–3042, 2016. 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 6

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2014. 4

[7] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021. 6, 7, 8

[8] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. https://github.com/Uason-Chen/CTR-GCN, 2021. 7

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 6

[10] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14755–14764, 2022. 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. 3

[14] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015. 2

[15] EyesJapan. Eyes Japan. https://mpcapdata.com, 2018. 1, 2

[16] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2021. 6, 7, 8

[17] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. https://github.com/hehefan/P4Transformer, 2021. 7

[18] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan S. Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2, 6, 7, 8

[19] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan S. Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. https://github.com/hehefan/Point-Spatio-Temporal-Convolution, 2021. 7

[20] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr 2021. 4

[21] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):90:1–90:12, 2019. 3

[22] Bradley Hayes and Julie A Shah. Interpretable models for fast activity recognition and anomaly explanation during collaborative robotics tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6586–6593. IEEE, 2017. 1

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 3, 5

[24] Yonghong Hou, Zhaoyang Li, Pichao Wang, and Wanqing Li. Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(3):807–811, 2018. 2

[25] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *Proceedings of the European conference on computer vision (ECCV)*, 2022. 8

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[27] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1, 2, 8

[28] Alyssa Kubota, Tariq Iqbal, Julie A Shah, and Laurel D Riek. Activity recognition in manufacturing: The roles of motion capture and semg+ inertial wearables in detecting fine vs. gross motion. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6533–6539. IEEE, 2019. 1

[29] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–14, 2010. 1

[30] Xing Li, Qian Huang, Zhijian Wang, Zhenjie Hou, and Tianjin Yang. Sequentialpointnet: A strong frame-level parallel point cloud sequence network for 3d action recognition, 2021. 6, 7, 8

[31] Xing Li, Qian Huang, Zhijian Wang, Zhenjie Hou, and Tianjin Yang. Sequentialpointnet: A strong frame-level parallel point cloud sequence network for 3d action recognition. https://github.com/XingLi1012/SequentialPointNet, 2021. 7

[32] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 8

[33] 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):2684–2701, 2020. 2

[34] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Skeleton-based online action prediction using scale selection network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6):1453–1467, 2020. 2

[35] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[36] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *ICCV*, 2019. 2

[37] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 6, 7, 8

[38] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. https://github.com/kenziyuliu/MS-G3D, 2020. 7

[39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. In *SIGGRAPH Asia*, 2015. 3

[40] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 1, 3

[41] Adrien Malaisé, Pauline Maurice, Francis Colas, François Charpillet, and Serena Ivaldi. Activity recognition with multiple wearable sensors for industrial applications. In *ACHI 2018-Eleventh International Conference on Advances in Computer-Human Interactions*, 2018. 1

[42] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *International Conference on Advanced Robotics (ICAR)*, pages 329–336, 2015. 1, 6

[43] Rahil Mehrizi, Xi Peng, Xu Xu, Shaoting Zhang, Dimitris N. Metaxas, and Kang Li. A computer vision based method for 3d posture estimation of symmetrical lifting. *Journal of biomechanics*, 69:40–46, 2018. 1

[44] Celso Melo, Brandon Rothrock, Prudhvi Gurram, Oytun Ulutan, and B. Manjunath. Vision-based gesture recognition in human-robot teams using synthetic data. pages 10278–10284, 10 2020. 2

[45] Matteo Menolotto, Dimitrios-Sokratis Komaris, Salvatore Tedesco, Brendan O'Flynn, and Michael Walsh. Motion capture technology in industrial applications: A systematic review. *Sensors*, 20(19):5687, 2020. 1

[46] MocapClub. Motion Capture Club. http://www.mocapclub.com/, 2009. 1, 2

[47] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database hdm05, 2007. 1, 2

[48] OSU. ACCAD. https://accad.osu.edu/research/motion-lab/system-data, 2018. 1, 2

[49] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 3

[50] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, June 2021. 1, 2, 6

[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 2, 3

[52] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2, 3

[53] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. 2021. 3

[54] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 2, 3

[55] Adrian Sanchez-Caballero, Sergio de López Diz, David Fuentes-Jiménez, Cristina Losada-Gutiérrez, Marta Marrón Romera, David Casillas-Perez, and Mohammad Ibrahim Sarker. 3dfcnn: Real-time action recognition using 3d deep neural networks with raw depth information. *CoRR*, abs/2006.07743, 2020. 2

[56] Adrian Sanchez-Caballero, David Fuentes-Jiménez, and Cristina Losada-Gutiérrez. Exploiting the convlstm: Human action recognition using raw depth video-based recurrent neural networks. *CoRR*, abs/2006.07744, 2020. 2

[57] SFU. SFU Motion Capture Database. http://mocap.cs.sfu.ca/. 1, 2

[58] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 2

[59] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3), mar 2022. 3

[60] 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, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12026–12035. Computer Vision Foundation / IEEE, 2019. 2

[61] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 6, 7, 8

[62] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. https://github.com/abhinanda-punnakkal/BABEL/tree/main/action_recognition, 2019. 7

[63] Jiankai Sun, Bolei Zhou, Michael J Black, and Arjun Chandrasekaran. Locate: End-to-end localization of actions in 3d with transformers. *arXiv preprint arXiv:2203.10719*, 2022. 6

[64] Nikolaus F. Troje. Decomposing biological motion: a framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2 5:371–87, 2002. 1

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 4

[66] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 2010. 3

[67] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matthew J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *International Conference on Computer Vision, ICCV*, 2021. 3, 5

[68] Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, and Philip O. Ogunbona. Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Trans. Multim.*, 20(5):1051–1061, 2018. 2

[69] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps using convolutional neural networks. *CoRR*, abs/1611.02447, 2016. 2

[70] Y. Wang, Y. Xiao, F. Xiong, W. Jiang, Z. Cao, J. Zhou, and J. Yuan. 3dv: 3d dynamic voxel for action recognition in depth video. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 508–517, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 2

[71] Yang Xiao, Jun Chen, Yancheng Wang, Zhiguo Cao, Joey Tianyi Zhou, and Xiang Bai. Action recognition for depth video using multi-view dynamic images. *Inf. Sci.*, 480:287–304, 2019. 2

[72] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7444–7452. AAAI Press, 2018. 2

[73] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[74] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737, 2018. 6

[75] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157, 2017. 2

[76] Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation*, pages 33–42, 2012. 1, 2

[77] Chun Zhu and Weihua Sheng. Motion-and location-based online human daily activity recognition. *Pervasive and Mobile Computing*, 7(2):256–269, 2011. 1