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

Anonymous authors
Paper under double-blind review

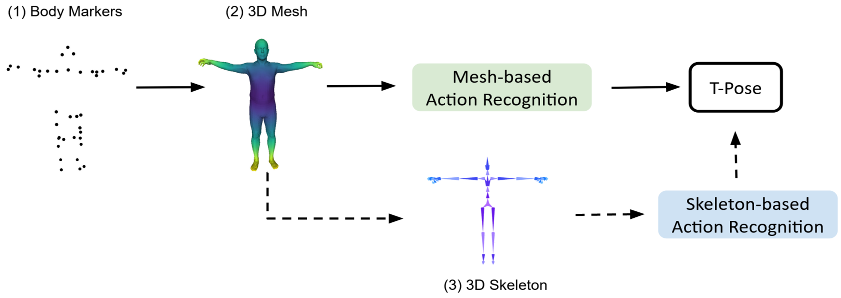
(1) Body Markers
(2) 3D Mesh
Mesh-based Action Recognition
T-Pose
Skeleton-based Action Recognition
(3) 3D Skeleton

Figure 1: Existing MoCap-based action recognition methods first converted body markers into a human body mesh and then predicted 3D skeletons from mesh vertices. Skeleton-based action recognition models were used to recognize human actions (Punnakkal et al., 2021) (dotted line). We propose a method that directly models the dynamics of raw mesh sequences.

Abstract

We study the problem of human action recognition using motion capture (MoCap) sequences. Existing methods for MoCap-based action recognition take skeletons as input, which requires an extra manual mapping step and loses body shape information. We propose a novel method that directly models raw mesh sequences which can benefit from the body prior and surface motion. We propose a new hierarchical transformer with intra- and inter-frame attention to learn effective spatial-temporal representations. Moreover, our model defines two self-supervised learning tasks, namely masked vertex modeling and future frame prediction, to further learn the global context for appearance and motion. Our model achieves state-of-the-art performance compared to skeleton-based and point-cloud-based models. We will release our code and models.

1 Introduction

Motion Capture (MoCap) is the process of digitally recording human motion. With the emergence of marker-based motion capture systems (i.e. Vicon MX), MoCap datasets enable the fine-grained capture and analysis of human motions in 3D space (Mahmood et al., 2019; Punnakkal et al., 2021). They serves as key elements for various research fields, such as action recognition (OSU, 2018; SFU; MocapClub, 2009; EyesJapan, 2018; Punnakkal et al., 2021; Müller et al., 2007), tracking (Müller et al., 2007), pose estimation (Achilles et al., 2016; Kocabas et al., 2020), imitation learning (Zhao et al., 2012), and motion synthesis (Müller et al., 2007). MoCap is also the fundamental technology for content creation and user interaction in *Metaverse*. Understanding human behaviors from MoCap data allows the *Metaverse* system to properly interact with users and non-player characters (NPCs) (Huynh-The et al., 2022). Skeleton representations are commonly used to model MoCap sequences. Some early works (Barnachon et al., 2014; Li et al., 2010) 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 (OSU, 2018), BioMotion(Troje, 2002), Eyes Japan (EyesJapan, 2018), and KIT (Mandery et al., 2015) 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 representations such as NTU RGB+D (Shahroudy et al., 2016), Punnakkal *et al.* (Punnakkal et al., 2021) first used Mosh++ to fit body markers into meshes, and then predicted a 25-joint skeleton (Liu et al., 2020a) from the vertices of the SMPL-H meshes (Romero et al., 2017). Finally, a skeleton-based model (Shi et al., 2019c) was used to perform action recognition. Although those methods achieved advanced performance, they have the following disadvantages: First, they require an extra manual step 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 temporal mesh sequences for action recognition. 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*). We consider the flexibility of a transformer architecture which allows the self-attention mechanism to freely attend to any two vertices, making it possible to learn non-local relationships among vertex patches in the same frame (spatial domains) or across frames (temporal domains). We expect the model to learn spatial-temporal correlation across the entire sequence to alleviate the requirement of explicit vertex correspondence. Specifically, we first build mesh vertex patches by learning local connectivity information. Then we propose a hierarchical transformer, which performs intra- and inter-frame attention on those patches. We define two self-supervised learning tasks, namely masked vertex modeling and future frame prediction to enable the model to learn from the global context. To reconstruct masked vertices of different body parts, the model needs to learn prior information of human body in spatial dimension. To predict future frames, the model needs to understand meaningful surface movement in the temporal dimension. To this end, our hierarchical transformer pre-trained with those two objectives can further learn spatial-temporal context across entire frames, which improves the downstream action recognition task.

We evaluate our model on common MoCap benchmarks. Our proposed *STMT* 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 (Wang et al., 2018; Sanchez-Caballero et al., 2020a; Xiao et al., 2019; Sanchez-Caballero et al., 2020b; Liu et al., 2020b) and point clouds (Qi et al., 2017; Liu et al., 2019; Wang et al., 2020; Fan et al., 2021d). Depth-maps provide reliable 3D structural and geometric information which characterizes informative human actions. In MVDI (Xiao et al., 2019), dynamic images (Bilen et al., 2016) were extracted through multi-view projections from depth videos for 3D action recognition. 3D-FCNN (Sanchez-Caballero et al., 2020a) directly exploited a 3D-CNN to model depth videos. Another popular category of 3D human action recognition is based on 3D point clouds. PointNet (Qi et al., 2016) and PointNet++ (Qi et al., 2017) are the pioneering works contributing towards permutation invariance of 3D point sets for representing 3D geometric structure. Along this avenue, MeteorNet (Liu et al., 2019) stacked multi-frame point clouds and aggregates local features for action recognition. 3DV (Wang et al., 2020) transferred point cloud sequences into regular voxel sets to characterize 3D motion compactly via temporal rank pooling.

PSTNet (Fan et al., 2021d) 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 (OSU, 2018; SFU; MocapClub, 2009; EyesJapan, 2018; Punnakkal et al., 2021; Müller et al., 2007) serve as key elements for various research fields, such as action recognition (OSU, 2018; SFU; MocapClub, 2009; EyesJapan, 2018; Punnakkal et al., 2021; Müller et al., 2007), tracking (Müller et al., 2007), pose estimation (Achilles et al., 2016; Kocabas et al., 2020), imitation learning (Zhao et al., 2012), and motion synthesis (Müller et al., 2007). MoCap-based action recognition was formulated as a skeleton-based action recognition problem (Punnakkal et al., 2021). Various architectures have been investigated to incorporate skeleton sequences. In (Du et al., 2015; Zhang et al., 2017; Liu et al., 2017), skeleton sequences were treated as time-series inputs to RNNs. (Hou et al., 2018; Wang et al., 2016) respectively transformed skeleton sequences into spectral images and trajectory maps then adopted CNNs for feature learning. In (Yan et al., 2018), 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 prior body information and surface motion.

Masked Autoencoder. Masked autoencoder has gained attention in Natural Language Processing and Computer Vision to learn effective representations using auto-encoding. Among masked vision autoencoders, one of the early works is (Vincent et al., 2010), which treated the masking as a noise type and proposed denoising autoencoders which were trained locally to denoise corrupted versions of their inputs. (Vincent et al., 2008) used CNN to inpaint missing regions and learn context information. ViT (Dosovitskiy et al., 2021) proposed a self-supervised pre-training task to reconstruct masked tokens. More recently, BEiT (Bao et al., 2021) proposed to learn visual representations by predicting the discrete tokens (Ramesh et al., 2021). MAE (He et al., 2021) proposed a simple yet effective asymmetric framework for masked image modeling. In 3D point cloud analysis, Wang *et al.* (Wang et al., 2021) chose to first generate partial point clouds by calculating occlusion from random camera viewpoints, and then completed occluded point clouds using autoencoding. Point-BERT (Yu et al., 2022) followed the success of BERT (Devlin et al., 2019) to predict the masked tokens learned from points. Self-supervised learning models for temporal 3D sequences (*i.e.* point cloud, 3D skeleton) have not been fully explored. One of the probable reasons is that applying 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 input 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, and $\mathbf{P}_i \in \mathbb{R}^{N\times 3}$ represents the position of the N vertices of the body mesh in Cartesian coordinates. $\mathbf{A}_i \in \mathbb{R}^{N\times N}$ 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 (Loper et al., 2015), SMPL-H (Romero et al., 2017), and SMPL-X (Pavlakos et al., 2019). In this work, we use SMPL-H body models from AMASS (Mahmood et al., 2019) to obtain the mesh sequences, but our method can be easily adopted to other body models.

Mesh's local connectivity provides fine-grained information. Previous methods (Hanocka et al., 2019; Sharp et al., 2022) in mesh classification prove that explicitly using surface (e.g., mesh) connectivity can achieve higher accuracy. However, unlike structured 3D skeletons, there is no vertex-level correspondence across frames for mesh sequences, which prevents graph-based models from directly aggregating vertex features in the temporal dimension. Therefore, we propose to first

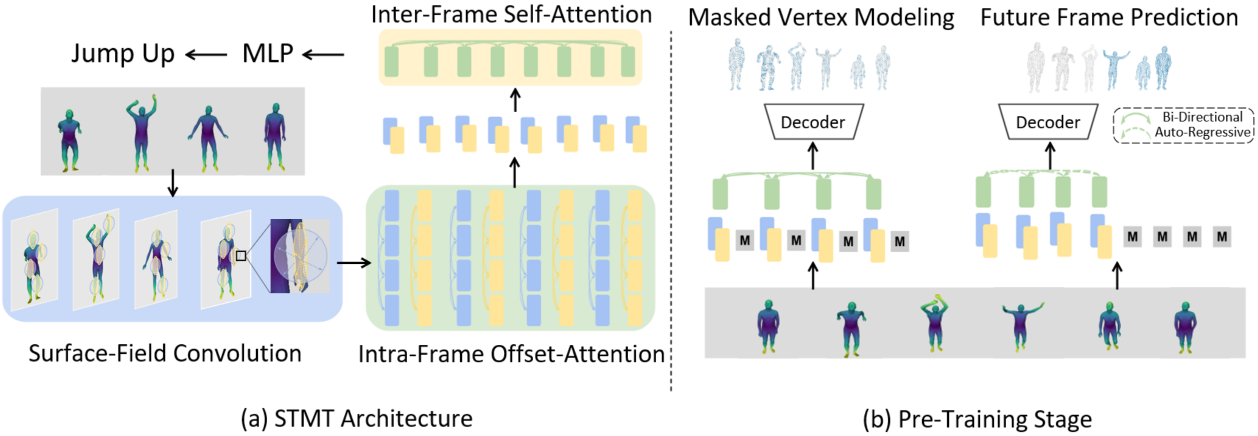

Figure 2: Overview of the proposed framework. **(a) Overview of *STMT*.** Given a mesh sequence, we first develop vertex patches by extracting both intrinsic (geodesic) and extrinsic (euclidean) features using surface field convolution. The intrinsic and extrinsic features are denoted by yellow and blue 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.

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 (Qi et al., 2016; 2017). 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. We propose a light-weighted surface field convolution to build local patches, which can be denoted as:

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

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

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 which 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 aggregated in the temporal domain because vertices are unordered. (2) Frame-level offset-attention can be used to mimic 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 (Bruna et al., 2014) 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 (Guo et al., 2021). 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:

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

where ϕ denotes a non-linear operator. $\boldsymbol{F}_{in} - \boldsymbol{F}_{sa}$ is proved to be analogous to discrete Laplacian operator, *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:

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

Here $F'^{(x,y,z)}_{VG} \in \mathbb{R}^{N\times d_g}$ and $F'^{(x,y,z)}_{VE} \in \mathbb{R}^{N\times d_e}$ are local patches learned using Equ. 1 and Equ. 2. $F'^{(x,y,z)}_{V} \in \mathbb{R}^{N\times d}$ 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. We show that the separate transformers can learn diverse attentions in Section 4.4.

3.3.2 Inter-Frame Self-Attention

Given F'_V which encodes local connectivity information, we use self-attention (SA) (Vaswani et al., 2017) 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:

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

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:

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

$$\boldsymbol{F}_{sa} = \boldsymbol{A}\cdot\boldsymbol{V} \tag{7}$$

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: $\boldsymbol{F}^{(x,y,z)}_{sa} = \sum \boldsymbol{A}^{(x,y,z),(x',y',z')} \times \boldsymbol{V}^{(x',y',z')}$, 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 (He et al., 2021). 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 (Cole et al., 2022). However, existing MoCap benchmarks are relatively small compared to 2D datasets like ImageNet (Deng et al., 2009). (2) Self-supervised learning for 3D data sequences is computationally expensive in terms of both 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.

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. Pseudo-code 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 $\theta_s$
    $\theta_{new} \leftarrow Initialize(t_{max}, n, 3)$
    $P \leftarrow \{I_{bone}, I_{leg}^{left}, I_{leg}^{right}, I_{arm}^{left}, I_{arm}^{right}\}$
    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 $\theta_{new}$
```

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 (Brown et al., 2020). However, directly using RNN-based models (Cho et al., 2014) 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 unidirectional attention helps the model to understand movement patterns and trajectories, which is beneficial for the downstream action recognition task.

3.5 Training

In the pre-training stage, we use PCN (Yuan et al., 2018) as the decoder to reconstruct masked vertices and predict future frames. The decoder is shared for the two pre-text 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 (Punnakkal et al., 2021; Sun et al., 2022), we evaluate our model on two common benchmarks: KIT(Mandery et al., 2015) and BABEL (Punnakkal et al., 2021). **KIT** is one of the largest MoCap datasets. It has 56 classes with 6,570 sequences in total. (2) **BABEL** is the largest 3D dataset of dense action labels that are precisely aligned with their corresponding movement. It leverages the large-scale AMASS dataset (Mahmood et al., 2019) for MoCap sequences, and 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 skeleton-based action recognition datasets (*e.g.* NTU RGB+D (Shahroudy et al., 2016)), are not suitable for our experiments, as they do not provide full 3D surfaces or SMPL parameters.

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 (Punnakkal et al., 2021) 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

Method	Input	KIT		BABEL-60	
		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
2s-AGCN-FL (Shi et al., 2019b) (CVPR'19)	3D Skeleton	42.44	75.60	49.62	79.12
2s-AGCN-CE (Shi et al., 2019b) (CVPR'19)	3D Skeleton	57.46	81.54	63.57	86.77
CTR-GCN (Chen et al., 2021a) (ICCV'21)	3D Skeleton	64.65	87.90	67.30	88.50
MS-G3D (Liu et al., 2020c) (CVPR'20)	3D Skeleton	65.38	87.90	67.43	87.99
PSTNet(Fan et al., 2021d) (ICLR'21)	Point Cloud	56.93	88.21	61.94	84.11
SequentialPointNet(Li et al., 2021b) (arXiv'21)	Point Cloud	59.75	88.01	62.92	84.58
P4Transformer(Fan et al., 2021a) (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.

the position of the J joints in the skeleton, in Cartesian co-ordinates, (x, y, z). 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 (Shi et al., 2019b), CTR-GCN (Chen et al., 2021a), and MS-G3D (Liu et al., 2020c) 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 (Punnakkal et al., 2021). For the comparison with point-cloud baselines, we choose PSTNet (Fan et al., 2021d), SequentialPointNet(Li et al., 2021b), and P4Transformer (Fan et al., 2021a). 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 (Shi et al., 2019a), (Chen et al., 2021b), and (Liu et al., 2020d), respectively. We train models for 250 epochs with a batch size of 64. For point-cloud-based baselines, we use the official implementations of PSTNet, SequentialPointNet, P4Transformer from (Fan et al., 2021c), (Li et al., 2021a), and (Fan et al., 2021b). All models use Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.0001. We train models for 200 epochs with a batch size of 32. Our *STMT* model is pre-trained using the Adam optimizer with a learning rate of 0.0001. The model is pre-trained for 120 epochs with batch size 128. The hyper-parameters for fine-tuning are the same as the point-cloud baselines for a fair comparison.

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.

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 that there is 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 attend to frames in the past and no movement information is available from nearby frames. The model can achieve the best performance with both MVM and FFP.

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

Table 2: Performance of ablated versions of our model.

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	Pretrain Loss ($\times 10^4$)	Finetune 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: Effectiveness of Different Masking Ratios.

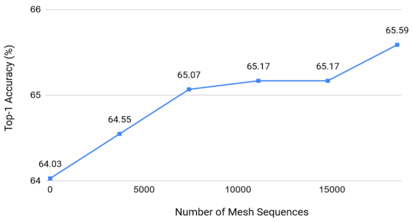

Figure 3: Effectiveness of Different Number of Mesh Sequences.

4.6 Analysis

Different Pre-Training Strategies We pre-trained 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). The third shows the result for the model pre-trained on KIT dataset (with Joint Shuffle augmentation). We observe our model can achieve better performance with Joint Shuffle, as it can synthesize large-scale mesh sequences.

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 on 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 with 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 high pre-training loss, while the fine-tuning 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 effectiveness of different numbers of mesh sequences used in pre-training. We report the fine-tuning top-1 classification accuracy in Figure 3. 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.

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 *STMT* 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.

REFERENCES

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*, pp. 491–499. Springer, 2016.

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

Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. *Pattern Recognition*, 47(1):238–247, 2014. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2013.06.020. URL `https://www.sciencedirect.com/science/article/pii/S0031320313002720`.

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)*, pp. 3034–3042, 2016. doi: 10.1109/CVPR.2016.331.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations*, 2014. URL `http://arxiv.org/abs/1312.6203`.

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*, pp. 13359–13368, 2021a.

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`, 2021b.

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.

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*, pp. 14755–14764, 2022.

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*, pp. 248–255. Ieee, 2009.

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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

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.

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)*, pp. 1110–1118, 2015. doi: 10.1109/CVPR.2015.7298714.

EyesJapan. Eyes Japan. `https://mpcapdata.com`, 2018.

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*, 2021a.

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

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`, 2021c.

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, 2021d. URL `https://openreview.net/forum?id=O3bqkf_Puys`.

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. ISSN 2096-0662. doi: 10.1007/s41095-021-0229-5. URL `http://dx.doi.org/10.1007/s41095-021-0229-5`.

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.

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.

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. doi: 10.1109/TCSVT.2016.2628339.

Thien Huynh-The, Quoc-Viet Pham, Xuan-Qui Pham, Thanh Thi Nguyen, Zhu Han, and Dong-Seong Kim. Artificial intelligence for the metaverse: A survey. *ArXiv*, abs/2202.10336, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

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*, pp. 9–14, 2010. doi: 10.1109/CVPRW.2010.5543273.

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`, 2021a.

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, 2021b. URL `https://arxiv.org/abs/2111.08492`.

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.

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, 2020a.

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, 2020b. doi: 10.1109/TPAMI.2019.2898954. URL `https://doi.org/10.1109/TPAMI.2019.2898954`.

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

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*, pp. 143–152, 2020c.

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`, 2020d.

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

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.

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)*, pp. 329–336, 2015.

MocapClub. Motion Capture Club. `http://www.mocapclub.com/`, 2009.

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

OSU. ACCAD. `https://accad.osu.edu/research/motion-lab/system-data`, 2018.

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)*, pp. 10975–10985, 2019.

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)*, pp. 722–731, June 2021.

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.

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.

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

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), November 2017.

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, 2020a. URL `https://arxiv.org/abs/2006.07743`.

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, 2020b. URL https://arxiv.org/abs/2006.07744.

SFU. SFU Motion Capture Database. http://mocap.cs.sfu.ca/.

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.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3), mar 2022. ISSN 0730-0301. doi: 10.1145/3507905. URL https://doi.org/10.1145/3507905.

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, 2019a.

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

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*, pp. 12026–12035. Computer Vision Foundation / IEEE, 2019c. doi: 10.1109/CVPR.2019.01230. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Shi_Two-Stream_Adaptive_Graph_Convolutional_Networks_for_Skeleton-Based_Action_Recognition_CVPR_2019_paper.html.

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.

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

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 (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. 2008.

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.

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.

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. URL http://arxiv.org/abs/1611.02447.

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. doi: 10.1109/TMM.2018.2818329. URL https://doi.org/10.1109/TMM.2018.2818329.

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)*, pp. 508–517, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00059. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00059`.

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. doi: 10.1016/j.ins.2018.12.050. URL `https://doi.org/10.1016/j.ins.2018.12.050`.

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 (eds.), *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*, pp. 7444–7452. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17135`.

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.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pp. 728–737, 2018. doi: 10.1109/3DV.2018.00088.

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)*, pp. 148–157, 2017. doi: 10.1109/WACV.2017.24.

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*, pp. 33–42, 2012.