MotionBERT: Unified Pretraining for Human Motion Analysis

Wentao Zhu[1] Xiaoxuan Ma[1] Zhaoyang Liu[2] Libin Liu[1] Wayne Wu[2,3] Yizhou Wang[1]

[1] Peking University [2] SenseTime Research [3] Shanghai AI Laboratory

{wtzhu,maxiaoxuan,libin.liu,yizhou.wang}@pku.edu.cn

{zyliumy,wuwenyan0503}@gmail.com

Abstract

We present MotionBERT, a unified pretraining framework, to tackle different sub-tasks of human motion analysis including 3D pose estimation, skeleton-based action recognition, and mesh recovery. The proposed framework is capable of utilizing all kinds of human motion data resources, including motion capture data and in-the-wild videos. During pretraining, the pretext task requires the motion encoder to recover the underlying 3D motion from noisy partial 2D observations. The pretrained motion representation thus acquires geometric, kinematic, and physical knowledge about human motion and therefore can be easily transferred to multiple downstream tasks. We implement the motion encoder with a novel Dual-stream Spatio-temporal Transformer (DSTformer) neural network. It could capture long-range spatio-temporal relationships among the skeletal joints comprehensively and adaptively, exemplified by the lowest 3D pose estimation error so far when trained from scratch. More importantly, the proposed framework achieves state-of-the-art performance on all three downstream tasks by simply finetuning the pretrained motion encoder with 1-2 linear layers, which demonstrates the versatility of the learned motion representations. [1]

1. Introduction

Perceiving and understanding human activities have long been a core pursuit of machine intelligence. To this end, researchers define various sub-tasks to abstract semantic representations from videos, *e.g.* skeleton keypoints [13, 31], action labels [53, 107], and surface meshes [41, 59]. While existing studies have made significant progress in each of the sub-tasks, they tend to confine and model the problems in a disjoint manner. For example, Graph Convolutional Networks (GCN) have been applied to modeling spatio-temporal relationship of human joints in both 3D pose estimation [12, 25, 101] and action recognition [83, 107], but their connections are barely explored. Intuitively, these models should all have learned to capture typical human motion patterns despite being designed for different problems. Nonetheless, current methods fail to mine and utilize such commonalities across the sub-tasks.

One major obstacle to unifying the sub-tasks of human motion analysis is data resource heterogeneity. Mocap systems [34, 63] offer high-fidelity 3D motion with markers and sensors, but the appearances of captured videos are usually constrained to simple indoor scenes. Action recognition datasets provide annotations of the action semantics, but they either contain no human pose labels [15, 82] or feature limited motion of daily activities [52, 53, 81]. In-the-wild human videos are massively accessible with diverse appearance and motion, but acquiring precise 2D pose annotations needs non-trivial efforts [3], and getting ground-truth (GT) 3D joint locations is almost impossible. Consequently, existing studies basically focus on one specific sub-task using an isolated type of motion data.

In this paper, we introduce a new framework named MotionBERT for human motion analysis across various sub-tasks, as illustrated in Figure 1. It consists of a unified pretraining stage and a task-specific finetuning stage, motivated by the recent successful practices in natural language processing [11, 26, 78] and computer vision [7, 32]. During pretraining, a motion encoder is trained to recover 3D human motion from corrupted 2D skeleton sequences, which could incorporate different kinds of human motion data sources. This challenging pretext task intrinsically requires the motion encoder to i) infer the underlying 3D human structures from its temporal movements; ii) recover the erroneous and missing observations. In this way, the motion encoder captures the human motion commonsense such as joint linkages, anatomical constraints, and temporal dynamics. In practice, we propose *Dual-stream Spatio-temporal Transformer (DSTformer)* as the motion encoder to capture the long-range

[1] Project page: https://motionbert.github.io/

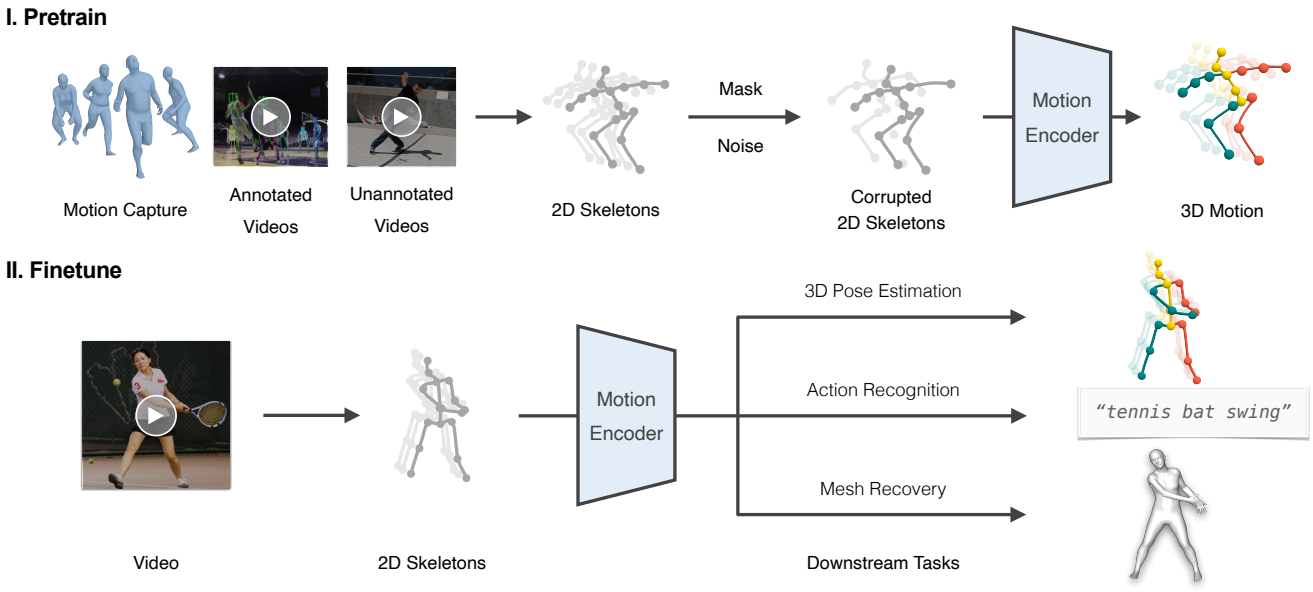

Figure 1. **Framework overview.** The proposed framework consists of two stages. During the pretraining stage, we extract the 2D skeleton sequences from a variety of motion data sources. We then corrupt the 2D skeletons by applying random masks and noises, from which the motion encoder is trained to recover the 3D motion. At the finetuning stage, we adapt the learned motion representations to different downstream tasks by jointly optimizing the pretrained motion encoder and a few linear layers.

relationship among skeleton keypoints. We suppose that the motion representations learned from large-scale and diversified data resources could be shared by all the relevant downstream tasks and benefit their performance. Therefore, for each downstream task, we adapt the pretrained motion representations with a simple regression head using task-specific training data and supervisory signals.

In summary, the contributions of this work are three-fold: 1) We propose a unified pretraining method to leverage the large-scale yet heterogeneous human motion sources and learn generalizable motion representations. Our approach could take advantage of the precision of 3D mocap data and the diversity of in-the-wild RGB videos at the same time. 2) We design a dual-stream Transformer network with cascaded spatio-temporal self-attention blocks that serves as a general backbone for human motion modeling. 3) The proposed unified pretraining approach empowers a versatile human motion representation, which could be transferred to multiple downstream tasks even with scarce labeled data. Without bells and whistles, MotionBERT outperforms the state-of-the-art methods on 3D pose estimation, skeleton-based action recognition, and mesh recovery, respectively.

2. Related Work

Learning Human Motion Representations. Early works formulate human motion with Hidden Markov Models [46, 94] and graphical models [44, 86]. Kanazawa *et al*. [38] design a temporal encoder and a hallucinator to learn representations of 3D human dynamics. Zhang *et al*. [114] predict future 3D dynamics in a self-supervised manner. Sun *et al*. [89] further incorporate action labels with an action memory bank. From the action recognition perspective, a variety of pretext tasks are designed to learn motion representations in a self-supervised manner, including future prediction [87], jigsaw puzzle [49], skeleton-contrastive [93], speed change [88], cross-view consistency [51], and contrast-reconstruction [102]. Similar techniques are also explored in tasks like motion assessment [29, 72] and motion retargeting [110, 121]. These methods leverage homogeneous motion data, design corresponding pretext tasks, and apply them to a specific downstream task. In this work, we propose a unified pretrain-finetune framework to incorporate heterogeneous data resources and demonstrate its versatility in various downstream tasks.

3D Human Pose Estimation. The 3D human pose estimation methods can be divided into two categories. The first is to estimate 3D poses with CNN directly from images [69, 91, 118]. One limitation of these approaches is that there is a trade-off between 3D pose precision and appearance diversity under current data collection techniques. The second category is to extract the 2D pose first, then lift the estimated 2D pose to 3D with a separate neural network. The lifting can be achieved via Fully Connected Network [65], Temporal Convolutional Network (TCN) [21, 77], GCN [12, 25, 101], and Transformer [48, 116, 117]. Our framework is built upon the second category as we use the proposed DSTformer to accomplish the 2D-to-3D lifting

pretext task.

Skeleton-based Action Recognition. Early works [62, 100, 111] point out the intrinsic relationship between human pose estimation and action recognition. Towards modeling the spatio-temporal relationship among human joints, previous studies employ LSTM [85, 120] and GCN [20, 47, 57, 83, 107]. Most recently, PoseConv3D [28] proposes to apply 3D-CNN on the stacked 2D joint heatmaps and achieves improved results. In addition to the fully-supervised action recognition task, NTU-RGB+D-120 [53] brings attention to the challenging one-shot action recognition problem. To this end, SL-DML [68] applies deep metric learning to multi-modal signals. Sabater *et al*. [80] explores one-shot recognition in therapy scenarios with TCN. We demonstrate that the pretrained motion representations could generalize well to action recognition tasks, and the pretrain-finetune framework is a suitable solution for the one-shot challenges.

Human Mesh Recovery. Based on the parametric human models such as SMPL [59], many research works [37, 70, 106, 115] focus on regressing the human mesh from a single image. Despite their promising per-frame results, these methods yield jittery and unstable results [41] when applied to videos. Several works [22, 38, 41, 92] take video clips as input and produce smoother results by exploiting the temporal cues. Another common problem is that paired images and GT meshes are mostly captured in constrained scenarios, which limits the generalization ability of the above methods. To that end, Pose2Mesh [23] proposes to first extract 2D skeletons using an off-the-shelf pose estimator, then lift them to 3D mesh vertices. We adopt the 2D skeletons as the intermediate representation and further extend them to video inputs, which reduces the ambiguity and could further benefit from our pretrained motion representations.

3. Method

3.1. Overview

As discussed in Section 1, our approach consists of two stages, namely unified pretraining and task-specific finetuning. In the first stage, we train a motion encoder to complete the 2D-to-3D lifting task, where we use the proposed DSTformer as the backbone. In the second stage, we finetune the pretrained motion encoder and a few linear layers on the downstream tasks. We use 2D skeleton sequences as input for both pretraining and finetuning because they could be reliably extracted from all kinds of motion sources [3, 9, 63, 73, 90], and is more robust to variations [18, 28]. Existing studies have shown the effectiveness of using 2D skeleton sequences for different downstream tasks [23, 28, 77, 95]. We will first introduce the architecture of DSTformer, and then describe the training scheme in detail.

Figure 2 shows the network architecture for 2D-to-3D lifting. Given an input 2D skeleton sequence $\mathbf{x} \in \mathbb{R}^{T\times J\times C_{\text{in}}}$, we first project it to a high-dimensional feature $\mathbf{F}^0 \in \mathbb{R}^{T\times J\times C_{\text{f}}}$, then add learnable spatial positional encoding $\mathbf{P}^{\text{S}}_{\text{pos}} \in \mathbb{R}^{1\times J\times C_{\text{f}}}$ and temporal positional encoding $\mathbf{P}^{\text{T}}_{\text{pos}} \in \mathbb{R}^{T\times 1\times C_{\text{f}}}$ to it. We then use the sequence-to-sequence model DSTformer to calculate $\mathbf{F}^i \in \mathbb{R}^{T\times J\times C_{\text{f}}}$ $(i = 1, \dots, N)$ where N is the network depth. We apply a linear layer with tanh activation to $\mathbf{F}^N$ to compute the motion representation $\mathbf{E} \in \mathbb{R}^{T\times J\times C_{\text{e}}}$. Finally, we apply a linear transformation to $\mathbf{E}$ to estimate 3D motion $\hat{\mathbf{X}} \in \mathbb{R}^{T\times J\times C_{\text{out}}}$. Here, T denotes the sequence length, and J denotes the number of body joints. C_{in}, C_{f}, C_{e}, and C_{out} denote the channel numbers of input, feature, embedding, and output respectively. We first introduce the basic building blocks of DSTformer, *i.e*. Spatial and Temporal Blocks with Multi-Head Self-Attention (MHSA), and then explain the DSTformer architecture design.

3.2. Network Architecture

Spatial Block. Spatial MHSA (S-MHSA) aims at modeling the relationship among the joints within the same time step. It is defined as

$$\begin{aligned}\text{Spatial-MHSA}(\mathbf{Q}_{\text{S}}, \mathbf{K}_{\text{S}}, \mathbf{V}_{\text{S}}) &= [\text{head}_1; \dots; \text{head}_h]\mathbf{W}^P_{\text{S}}, \\ \text{head}_i &= \text{softmax}(\frac{\mathbf{Q}^i_{\text{S}}(\mathbf{K}^i_{\text{S}})^T}{\sqrt{d_K}})\mathbf{V}^i_{\text{S}},\end{aligned} \tag{1}$$

where $\mathbf{W}^P_{\text{S}}$ is a projection parameter matrix, h is the number of the heads, $i \in 1, \dots, h$. We utilize self-attention to get the query $\mathbf{Q}_{\text{S}}$, key $\mathbf{K}_{\text{S}}$, and value $\mathbf{V}_{\text{S}}$ from input per-frame spatial feature $\mathbf{F}_{\text{S}} \in \mathbb{R}^{J\times C_{\text{e}}}$ for each head_i,

$$\mathbf{Q}^i_{\text{S}} = \mathbf{F}_{\text{S}}\mathbf{W}^{(Q,i)}_{\text{S}},\ \mathbf{K}^i_{\text{S}} = \mathbf{F}_{\text{S}}\mathbf{W}^{(K,i)}_{\text{S}},\ \mathbf{V}^i_{\text{S}} = \mathbf{F}_{\text{S}}\mathbf{W}^{(V,i)}_{\text{S}}, \tag{2}$$

where $\mathbf{W}^{(Q,i)}_{\text{S}}$, $\mathbf{W}^{(K,i)}_{\text{S}}$, $\mathbf{W}^{(V,i)}_{\text{S}}$ are projection matrices, and d_K is the dimension of $\mathbf{K}_{\text{S}}$. We apply S-MHSA to features of different time steps in parallel. Residual connection and layer normalization (LayerNorm) are used to the S-MHSA result, which is further fed into a multilayer perceptron (MLP), and followed by a residual connection and LayerNorm following [98]. We denote the entire spatial block with MHSA, LayerNorm, MLP, and residual connections by $\mathcal{S}$.

Temporal Block. Temporal MHSA (T-MHSA) aims at modeling the relationship across the time steps for a body joint. Its computation process is similar with S-MHSA except that the MHSA is applied to the per-joint temporal feature $\mathbf{F}_{\text{T}} \in \mathbb{R}^{T\times C_{\text{e}}}$ and parallelized over the spatial dimension.

$$\begin{aligned}\text{Temporal-MHSA}(\mathbf{Q}_{\text{T}}, \mathbf{K}_{\text{T}}, \mathbf{V}_{\text{T}}) &= [\text{head}_1; \dots; \text{head}_h]\mathbf{W}^P_{\text{T}}, \\ \text{head}_i &= \text{softmax}(\frac{\mathbf{Q}^i_{\text{T}}(\mathbf{K}^i_{\text{T}})^T}{\sqrt{d_K}})\mathbf{V}^i_{\text{T}},\end{aligned} \tag{3}$$

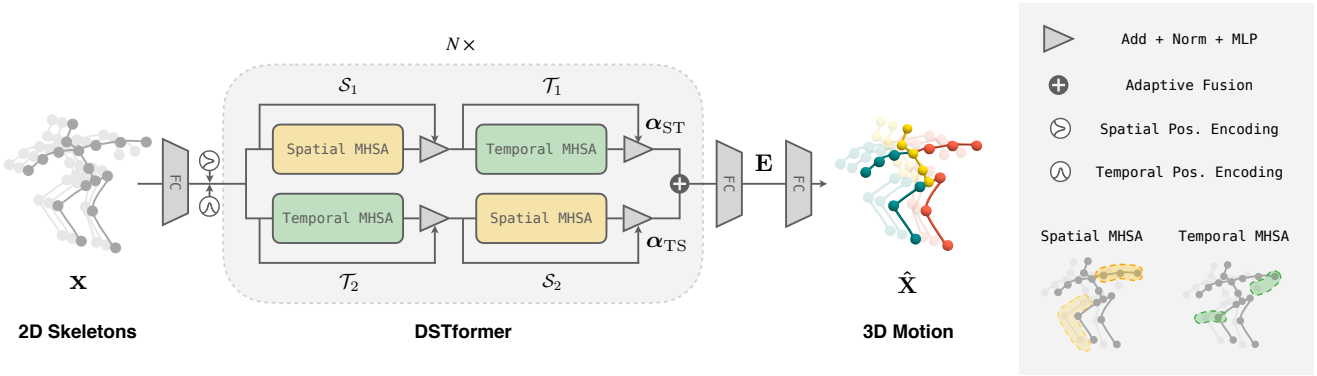

Figure 2. **Model architecture.** We propose the Dual-stream Spatio-temporal Transformer (DSTformer) as a general backbone for human motion analysis. DSTformer consists of N dual-stream-fusion modules. Each module contains two branches of spatial or temporal MHSA and MLP. The Spatial MHSA models the connection among different joints within a timestep, while the Temporal MHSA models the movement of one joint.

where $i \in 1, \dots, h$, $\mathbf{Q}_\mathrm{T}$, $\mathbf{K}_\mathrm{T}$, $\mathbf{V}_\mathrm{T}$ are computed similar with Formula 2. We denote the entire temporal block by $\mathcal{T}$.

Dual-stream Spatio-temporal Transformer. Given spatial and temporal MHSA that captures the intra-frame and inter-frame body joint interactions respectively, we assemble the basic building blocks to fuse the spatial and temporal information in the flow. We design a dual-stream architecture with the following guidelines: 1) The two streams could both model the comprehensive spatio-temporal context. 2) The two streams are specialized for different spatio-temporal aspects. 3) The two streams are fused together, and the fusion weights are dynamically balanced according to the input spatio-temporal characteristics.

Therefore, we stack the spatial and temporal MHSA blocks in different orders, forming two parallel computation branches. The output features of the two branches are fused using adaptive weights predicted by an attention regressor. The dual-stream-fusion module is then repeated for N times:

$$\mathbf{F}^i = \boldsymbol{\alpha}^i_{\mathrm{ST}} \circ \mathcal{T}^i_1(\mathcal{S}^i_1(\mathbf{F}^{i-1})) + \boldsymbol{\alpha}^i_{\mathrm{TS}} \circ \mathcal{S}^i_2(\mathcal{T}^i_2(\mathbf{F}^{i-1})), \quad i \in 1, \dots, N, \tag{4}$$

where $\mathbf{F}^i$ denotes the feature embedding at depth i, $\circ$ denotes element-wise production. Orders of $\mathcal{S}$ and $\mathcal{T}$ blocks are shown in Figure 2, and different blocks do not share weights. Adaptive fusion weights $\boldsymbol{\alpha}_{\mathrm{ST}}, \boldsymbol{\alpha}_{\mathrm{TS}} \in \mathbb{R}^{N \times T \times J}$ are given by

$$\boldsymbol{\alpha}^i_{\mathrm{ST}}, \boldsymbol{\alpha}^i_{\mathrm{TS}} = \mathrm{softmax}(\mathcal{W}([\mathcal{T}^i_1(\mathcal{S}^i_1(\mathbf{F}^{i-1})), \mathcal{S}^i_2(\mathcal{T}^i_2(\mathbf{F}^{i-1}))])), \tag{5}$$

where $\mathcal{W}$ is a learnable linear transformation. $[,]$ denotes concatenation.

3.3. Unified Pretraining

We address two key challenges when designing the unified pretraining framework: 1) How to learn a powerful motion representation with a universal pretext task. 2) How to utilize large-scale but heterogeneous human motion data in all kinds of formats.

For the first challenge, we follow the pretrained models in language [11, 26, 78] and vision [7, 32] modeling to construct the supervision signals, *i.e.* mask part of the input and use the encoded representations to reconstruct the whole input. Note that such "cloze" task naturally exists in human motion analysis, that is to recover the lost depth information from the 2D visual observations, *i.e.* 3D human pose estimation. Inspired by this, we leverage the large-scale 3D mocap data [63] and design a 2D-to-3D lifting pretext task. We first extract the 2D skeleton sequences $\mathbf{x}$ by projecting the 3D motion orthographically. Then, we corrupt $\mathbf{x}$ by randomly masking and adding noise to produce the corrupted 2D skeleton sequences, which also resemble the 2D detection results as it contains occlusions, detection failures, and errors. Both joint-level and frame-level masks are applied with certain probabilities. We use the aforementioned motion encoder to get motion representation $\mathbf{E}$ and reconstruct 3D motion $\hat{\mathbf{X}}$. We then compute the joint loss $\mathcal{L}_{\mathrm{3D}}$ between $\hat{\mathbf{X}}$ and GT 3D motion $\mathbf{X}$. We also add the velocity loss $\mathcal{L}_O$ following previous works [77, 116]. The 3D reconstruction losses are given by

$$\mathcal{L}_{\mathrm{3D}} = \sum_{t=1}^{T}\sum_{j=1}^{J} \parallel \hat{\mathbf{X}}_{t,j} - \mathbf{X}_{t,j} \parallel_2, \quad \mathcal{L}_O = \sum_{t=2}^{T}\sum_{j=1}^{J} \parallel \hat{\mathbf{O}}_{t,j} - \mathbf{O}_{t,j} \parallel_2, \tag{6}$$

where $\hat{\mathbf{O}}_t = \hat{\mathbf{X}}_t - \hat{\mathbf{X}}_{t-1}$, $\mathbf{O}_t = \mathbf{X}_t - \mathbf{X}_{t-1}$.

For the second challenge, we notice that 2D skeletons could serve as a universal medium as they can be extracted from all sorts of motion data sources. We further incorporate in-the-wild RGB videos into the 2D-to-3D lifting framework for unified pretraining. For RGB videos, the 2D skeletons $\mathbf{x}$ could be given by manual annotation [3] or 2D pose estimator [13, 90], and the depth channel of the extracted 2D skeletons is intrinsically "masked". Similarly, we add extra masks and noises to degrade $\mathbf{x}$ (if $\mathbf{x}$ already contains de-

tection noise, only masking is applied). As 3D motion GT $\mathbf{X}$ is not available for these data, we apply a weighted 2D re-projection loss which is calculated by

$$\mathcal{L}_{\text{2D}} = \sum_{t=1}^{T}\sum_{j=1}^{J} \boldsymbol{\delta}_{t,j} \|\hat{\mathbf{x}}_{t,j} - \mathbf{x}_{t,j}\|_2, \tag{7}$$

where $\hat{\mathbf{x}}$ is the 2D orthographical projection of the estimated 3D motion $\hat{\mathbf{X}}$, and $\boldsymbol{\delta} \in \mathbb{R}^{T\times J}$ is given by visibility annotation or 2D detection confidence.

The total pretraining loss is computed by

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{3D}} + \lambda_O \mathcal{L}_O}_{\text{for 3D data}} + \underbrace{\mathcal{L}_{\text{2D}}}_{\text{for 2D data}}, \tag{8}$$

where λ_O is a constant coefficient to balance the training loss.

3.4. Task-specific Finetuning

The learned feature embedding $\mathbf{E}$ serves as a 3D-aware and temporal-aware motion representation. For downstream tasks, we adopt the *minimalist* design principle, *i.e.* implementing a shallow downstream network and training without bells and whistles. In practice, we use an extra linear layer or an MLP with one hidden layer. We then finetune the whole network end-to-end.

3D Pose Estimation. As we utilize 2D-to-3D lifting as the pretext task, we simply reuse the whole pretraining network (motion encoder + linear). During finetuning, the input 2D skeletons are estimated from videos without extra masks or noises.

Skeleton-based Action Recognition. We simply perform a global average pooling over different persons and timesteps. The result is then fed into an MLP with one hidden layer. The network is trained with cross-entropy classification loss. For one-shot learning, we apply a linear layer after the pooled features to extract clip-level action representation. We introduce the detailed setup of one-shot learning in Section 4.4.

Human Mesh Recovery. We use SMPL [59] model to represent the human mesh and regress its parameters. The SMPL model consists of pose parameters $\theta \in \mathbb{R}^{72}$ and shape parameters $\beta \in \mathbb{R}^{10}$, and calculates the 3D mesh as $\mathcal{M}(\theta, \beta) \in \mathbb{R}^{6890\times 3}$. To regress the pose parameters for each frame, we feed the motion embeddings $\mathbf{E}$ to an MLP with one hidden layer and get $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{T\times 72}$. To estimate shape parameters, considering that the human shape over a video sequence is supposed to be consistent, we first perform an average pooling of $\mathbf{E}$ over the temporal dimension and then feed it into another MLP to regress a single $\hat{\beta}$ and then expand it to the entire sequence as $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{T\times 10}$. The shape MLP has the same architecture as the pose regression one, and they are initialized with the mean shape and pose, respectively, as in [41]. The overall loss is computed as

$$\mathcal{L} = \lambda_{\text{3D}}^{\text{m}} \mathcal{L}_{\text{3D}}^{\text{m}} + \lambda_\theta \mathcal{L}_\theta + \lambda_\beta \mathcal{L}_\beta, \tag{9}$$

where each term is calculated as

$$\mathcal{L}_{\text{3D}}^{\text{m}} = \|\hat{\mathbf{X}}^{\text{m}} - \mathbf{X}^{\text{m}}\|_1, \quad \mathcal{L}_\theta = \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1, \quad \mathcal{L}_\beta = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1. \tag{10}$$

Note that each 3D pose in motion $\mathbf{X}^{\text{m}}$ at frame t is regressed from mesh vertices by $\mathbf{X}_t^{\text{m}} = \mathbf{J}\mathcal{M}(\boldsymbol{\theta}_t, \boldsymbol{\beta}_t)$, where $\mathbf{J} \in \mathbb{R}^{J\times 6890}$ is a pre-defined matrix [9]. $\lambda_{\text{3D}}^{\text{m}}$, λ_θ and λ_β are constant coefficients to balance the training loss.

4. Experiments

4.1. Implementation

We implement the proposed motion encoder DSTformer with depth $N = 5$, number of heads $h = 8$, feature size $C_{\text{f}} = 256$, embedding size $C_{\text{e}} = 512$. For pretraining, we use sequence length $T = 243$. The pretrained model could handle different input lengths thanks to the Transformer-based backbone. During finetuning, we set the backbone learning rate to be $0.1\times$ of the new layer learning rate. We introduce the experiment datasets in the following sections respectively. Please refer to Appendix A.1 for more experimental details.

4.2. Pretraining

We collect diverse and realistic 3D human motion from two datasets, Human3.6M [34] and AMASS [63]. Human3.6M [34] is a commonly used indoor dataset for 3D human pose estimation which contains 3.6 million video frames of professional actors performing daily actions. Following previous works [65, 77], we use subjects 1, 5, 6, 7, 8 for training, and subjects 9, 11 for testing. AMASS [63] integrates most existing marker-based Mocap datasets [1, 2, 5, 10, 14, 17, 30, 33, 45, 58, 60, 64, 71, 84, 96, 97] and parameterizes them with a common representation. We do not use the videos or 2D detection results of the two datasets during pretraining; instead, we use orthographic projection to get the uncorrupted 2D skeletons. We further incorporate two in-the-wild RGB video datasets PoseTrack [3] (annotated) and InstaVariety [38] (unannotated) for higher motion diversity. We align the body keypoint definitions with Human3.6M and calibrate the camera coordinates to pixel coordinates following [24]. We randomly zero out 15% joints, and sample noises from a mixture of Gaussian and uniform distributions [16]. We first train on 3D data only for 30 epochs, then train on both 3D data and 2D data for 50 epochs, following the curriculum learning practices [8, 103].

4.3. 3D Pose Estimation

We evaluate the 3D pose estimation performance on Human3.6M [34] and report the mean per joint position error

Table 1. **Quantitative comparison of 3D human pose estimation.** Numbers are MPJPE (mm) on Human3.6M. T denotes the clip length used by the method. We select the best results reported by each work. † denotes using HRNet [90] for 2D detection.

MPJPE	T	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Martinez *et al*. [65] ICCV'17	1	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Pavlakos *et al*. [76] CVPR'18	1	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Ci *et al*. [25] ICCV'19	1	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Xu *et al*. [105] CVPR'21	1	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
Pavllo *et al*. [77] CVPR'19	243	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Cai *et al*. [12] ICCV'19	7	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Yeh *et al*. [112] NeurIPS'19	243	44.8	46.1	43.3	46.4	49.0	55.2	44.6	44.0	58.3	62.7	47.1	43.9	48.6	32.7	33.3	46.7
Liu *et al*. [56] CVPR'20	243	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Cheng *et al*. [21] AAAI'20 †	128	36.2	38.1	42.7	35.9	**38.2**	45.7	36.8	42.0	**45.9**	**51.3**	41.8	41.5	43.8	33.1	28.6	40.1
Wang *et al*. [101] ECCV'20 †	96	38.2	41.0	45.9	39.7	41.4	51.4	41.6	41.4	52.0	57.4	41.8	44.4	41.6	33.1	30.0	42.6
Zheng *et al*. [117] ICCV'21	81	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Wehrbein *et al*. [104] ICCV'21 †	200	38.5	42.5	39.9	41.7	46.5	51.6	39.9	40.8	49.5	56.8	45.3	46.4	46.8	37.8	40.4	44.3
Li *et al*. [48] CVPR'22	351	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
Zhang *et al*. [116] CVPR'22 †	243	36.7	39.0	36.5	39.4	40.2	**44.9**	39.8	36.9	47.9	54.8	39.6	37.8	39.3	29.7	30.6	39.8
Ours (Scratch)	243	36.3	38.7	38.6	33.6	42.1	50.1	36.2	35.7	50.1	56.6	41.3	37.4	37.7	25.6	26.5	39.2
Ours (Pretrained)	243	**36.1**	**37.5**	**35.8**	**32.1**	40.3	46.3	**36.1**	**35.3**	46.9	53.9	**39.5**	**36.3**	**35.8**	**25.1**	**25.3**	**37.5**

(MPJPE) in millimeters, which measures the average distance between the predicted joint positions and the GT after aligning the root joint. We use the Stacked Hourglass (SH) networks [73] to extract the 2D skeletons from videos, and finetune the entire network on Human3.6M [34] training set. In addition, we train a separate model of the same architecture, but with random initialization rather than pretrained weights. Both of our models outperform the state-of-the-art methods, as shown in Table 1. The model trained from scratch demonstrates the effectiveness of the proposed DSTformer in terms of learning 3D geometric structures and temporal dynamics. The proposed pretraining stage additionally lowers the error, which proves the advantage of the unified pertaining framework.

4.4. Skeleton-based Action Recognition

We further explore the possibility to learn action semantics with the pretrained motion representations. We use the human action dataset NTU-RGB+D [81] which contains 57K videos of 60 action classes, and we follow the data splits Cross-subject (X-Sub) and Cross-view (X-View). The dataset has an extended version, NTU-RGB+D-120 [53], which contains 114K videos of 120 action classes. We follow the suggested *One-shot* action recognition protocol on NTU-RGB+D-120. For both datasets, we use HRNet [90] to extract 2D skeletons following [28]. Similarly, we train a scratch model with random initialization for comparison. All the models are evaluated without ensemble. As Table 2 shows, our methods are comparable or superior to the state-of-the-art approaches. Notably, the pretraining stage accounts for a large performance gain.

In addition, we study the one-shot setting which is of practical importance. In real-world deployment, action recognition in certain scenarios (*e.g*. education, sports, healthcare) are highly demanded. However, only scarce annotations are available for the novel action classes that are unseen in the public datasets. As proposed in [53], we report the results on the evaluation set of 20 novel classes using only 1 labeled video for each class. The auxiliary set contains the other 100 classes, and all samples of these classes can be used for learning. We train the model on the auxiliary set using the supervised contrastive learning technique [39]. For a batch of auxiliary data, samples of the same class are pulled together, while samples of different classes are pushed away in the action embedding space. During the evaluation, we calculate the cosine distance between the test examples and the exemplars, and use 1-nearest neighbor to determine the class. Table 2 illustrates that the proposed models outperform state-of-the-art by a considerable margin. It is also worth noting the pretrained model achieves optimal performance with only 1-2 epochs of finetuning. The results imply that the pretraining stage indeed helps to learn a good motion representation which could generalize to novel downstream tasks even with limited data annotations.

4.5. Human Mesh Recovery

We conduct experiments on Human3.6M [34] and 3DPW [99] datasets. For Human3.6M, we follow the conventional training and test split [65] as in Section 4.2. For 3DPW, we add COCO [50] dataset for training following [23]. We report results with and without the 3DPW training set as in [22, 41]. Following the common practice [37, 41, 43], we report MPJPE (mm) and PA-MPJPE (mm) of 14 joints obtained by $\mathbf{J}\mathcal{M}(\theta, \beta)$. PA-MPJPE calculates MPJPE after aligning with GT in translation, rotation, and scale. We further report the mean per vertex error (MPVE) (mm) of the mesh $\mathcal{M}(\theta, \beta)$, which measures the average distance between the estimated and GT vertices after aligning the root joint.

As shown in Table 3, our pretrained version outperforms

Table 2. **Quantitative comparison of skeleton-based action recognition accuracy.** (Left) Cross-subject and Cross-view recognition accuracy on NTU-RGB+D. (Right) One-shot recognition accuracy on NTU-RGB+D-120. All results are top-1 accuracy (%).

Method	X-Sub	X-View
ST-GCN [107] AAAI'18	81.5	88.3
2s-AGCN [83] CVPR'19	88.5	95.1
MS-G3D [57] CVPR'20	91.5	96.2
Shift-GCN [20] CVPR'20	90.7	96.5
CrosSCLR [51] CVPR'21	86.2	92.5
MCC (Pretrained) [88] ICCV'21	89.7	96.3
SCC (Pretrained) [109] ICCV'21	88.0	94.9
UNIK (Pretrained) [108] BMVC'21	86.8	94.4
CTR-GCN [19] ICCV'21	92.4	96.8
PoseConv3D [28] CVPR'22	**93.1**	95.7
Ours (Scratch)	87.6	93.4
Ours (Pretrained)	92.8	**97.1**

Method	Accuracy
ST-LSTM + AvgPool [54]	42.9
ST-LSTM + FC [55]	42.1
ST-LSTM + Attention [55]	41.0
APSR [53]	45.3
TCN OneShot [80]	46.5
SL-DML [68]	50.9
Skeleton-DML [67]	54.2
Ours (Scratch)	61.0
Ours (Pretrained)	**67.4**

Table 3. **Quantitative comparison of human mesh recovery on Human3.6M and 3DPW datasets.** On the 3DPW dataset, models are separately compared based on whether using 3DPW training data or not. † denotes using 3DPW training data.

Method	Input	Human3.6M			3DPW		
		MPVE	MPJPE	PA-MPJPE	MPVE	MPJPE	PA-MPJPE
HMR [37] CVPR'18	image	-	88.0	56.8	-	130.0	81.3
SPIN [42] ICCV'19	image	-	-	41.1	116.4	96.9	59.2
Pose2Mesh [23] ECCV'20	2D pose	83.9	64.9	46.3	-	88.9	58.3
I2L-MeshNet [70] ECCV'20	image	68.1	55.7	41.1	110.1	93.2	57.7
PyMAF [115] ICCV'21	image	-	57.7	40.5	110.1	92.8	58.9
TemporalContext [6] CVPR'19	video	-	77.8	54.3	-	-	72.2
HMMR [38] CVPR'19	video	-	-	56.9	139.3	116.5	72.6
DSD-SATN [92] ICCV'19	video	-	59.1	42.4	-	-	69.5
VIBE [41] CVPR'20	video	-	65.9	41.5	113.4	93.5	56.5
TCMR [22] CVPR'21	video	-	62.3	41.1	111.3	95.0	55.8
Ours (Scratch)	2D motion	76.7	61.9	44.6	113.9	100.0	61.9
Ours (Pretrained)	2D motion	**65.5**	**54.1**	**34.8**	**99.7**	**85.5**	**51.7**
MEVA [61] ACCV'20 †	video	-	76.0	53.2	-	86.9	54.7
VIBE [41] CVPR'20 †	video	-	65.6	41.4	99.1	82.9	51.9
TCMR [22] CVPR'21 †	video	-	-	-	102.9	86.5	52.7
Ours (Pretrained) †	2D motion	**66.0**	**54.3**	**35.2**	**94.2**	**80.9**	**49.1**

all the previous methods on the Human3.6M dataset for all metrics. We further evaluate the generalization ability of our approach on the 3DPW dataset, a challenging in-the-wild benchmark. Without using the 3DPW training set, our method with pretrained motion representations surpasses all previous works by a notable margin, especially in MPVE and MPJPE (improved by around 10mm). When using additional 3DPW training data, our method further reduces the estimation errors and constantly outperforms previous methods. The performance advantage demonstrates the benefits of the unified pretraining framework and the learned motion representations. Note that most previous works [22, 37, 41, 42, 61] use more datasets other than COCO [50] for training, such as LSP [35], MPI-INF-3DHP [66], *etc*.

4.6. Ablation Studies

Model Architecture. We first study the design choices of DSTformer and report the results in Table 6. From (a) to (f), we compare the different structures of basic Transformer modules. (a) and (b) are single stream versions with different orders. (c) limits each stream to either temporal or spatial modeling before fusion. (d) directly connects S-MHSA and T-MHSA without the MLP in between. (e) replaces the adaptive fusion with average pooling on two streams. (f) is the proposed DSTformer design. For all the variants, we control the total number of self-attention blocks to be the same, so the network capacity would not be the defining factor. The result confirms our design principles that both streams should be capable and meanwhile complementary, as introduced in Section 3.2. In addition, we find out that

Table 4. **Comparison of different pretraining settings.** The pretrained networks are separately finetuned and evaluated on downstream tasks. We report results of 3D Pose and Mesh on Human3.6M, Action on NTU-RGB+D.

Pretrain	Noise	Mask	2D Data	MPJPE (3D Pose)↓	MPVE (Mesh)↓	Accuracy (Action)↑
-	-	-	-	39.2mm	76.7mm	87.2%
✓	-	-	-	38.8mm	70.6mm	89.4%
✓	✓	-	-	38.1mm	68.4mm	90.7%
✓	✓	✓	-	**37.4mm**	67.8mm	91.9%
✓	✓	✓	✓	37.5mm	**65.5mm**	**92.8%**

Table 5. **Ablation study of partial finetuning.** The networks are separately finetuned and evaluated on downstream tasks with backbone frozen. We report results of 3D Pose on Human3.6M, Mesh on Human3.6M and 3DPW (without 3DPW training set), and Action on NTU-RGB+D and NTU-RGB+D-120.

Backbone	MPJPE (3D Pose) ↓	MPVE (Mesh)↓ (Human3.6M)	MPVE (Mesh)↓ (3DPW)	Accuracy (Action) ↑ (NTU-RGB+D X-View)	Accuracy (Action) ↑ (NTU-RGB+D-120 1-Shot)
Random	404.4mm	114.4mm	155.9mm	47.6%	46.8%
Pretrained	**40.3mm**	**77.1mm**	**108.4mm**	**85.4%**	**60.7%**

Table 6. **Comparison of model architecture design.** All the methods are trained on Human3.6M from scratch and measured by MPJPE (mm).

Method	Depth(N)	Heads(h)	Channels(C_e)	Basic Module	MPJPE
(a)	5	8	512	S-T	40.3
(b)	5	8	512	T-S	41.2
(c)	5	8	512	S + T	41.8
(d)	5	8	512	ST-MHSA	41.5
(e)	5	8	512	S-T + T-S (Average)	39.6
(f)	5	8	512	S-T + T-S (Adaptive)	**39.2**
(g)	4	8	512	S-T + T-S (Adaptive)	39.7
(h)	6	8	512	S-T + T-S (Adaptive)	40.4
(i)	5	6	512	S-T + T-S (Adaptive)	39.6
(j)	5	10	512	S-T + T-S (Adaptive)	39.5
(k)	5	8	256	S-T + T-S (Adaptive)	40.4

pairing each self-attention block with an MLP is crucial, as it could project the learned feature interactions and bring nonlinearity. From (g) to (k), we trial different model sizes. In general, we optimize the model architecture design based on their performance for the 3D human pose estimation task when trained from scratch, and apply the design to all the tasks without additional adjustment.

Pretraining Strategies. We evaluate how different pretraining strategies influence the performance of downstream tasks. Starting from the scratch baseline, we apply the proposed strategies one by one. As shown in Table 4, a vanilla 2D-to-3D pretraining stage brings benefits to all the downstream tasks. Introducing corruptions additionally improves the learned motion embeddings. Unified pretraining with in-the-wild videos enjoy higher motion diversity, which further helps several downstream tasks.

Partial Finetuning. In addition to end-to-end finetuning, we experiment to freeze the motion encoder backbone and only train the downstream network for each subtask. In order to verify the effectiveness of the pretrained motion representations, we compare the pretrained motion encoder and a randomly initialized motion encoder. The results are shown in Table 5. It can be seen that based on the frozen pretrained motion representations, our method could still achieve competitive performance on multiple downstream tasks. The performance advantages over the random initialized ones demonstrate that generalizable motion representations are learned via pretraining. In real-world applications, it is often desirable to simultaneously obtain predictions for multiple sub-tasks (*e.g.* action + mesh). Existing methods require separate models to handle specific sub-tasks. Pretraining and partial finetuning make it possible for all the downstream tasks to share the same backbone, which largely reduces the computation overhead.

5. Conclusion

In this paper, we present MotionBERT to unify various sub-tasks related to human motion with the pretrain-finetune paradigm. We design the 2D-to-3D lifting pretext task for pretraining on large-scale human motion data. We also propose the DSTformer as a universal human motion encoder. Experimental results on multiple benchmarks demonstrate the effectiveness of our method. As for the limitations, by now this work mainly focuses on learning from the skeletons of a single person. Future work may explore fusing the learned motion representations with image features and explicitly model human interactions.

References

[1] Advanced Computing Center for the Arts and Design. ACCAD MoCap Dataset. 5

[2] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, pages 1446–1455, June 2015. 5

[3] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, pages 5167–5176, 2018. 1, 3, 4, 5, 14

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. Human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 14

[5] Andreas Aristidou, Ariel Shamir, and Yiorgos Chrysanthou. Digital dance ethnography: Organizing large dance collections. *JOCCH*, 12(4):1–27, 2019. 5

[6] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*, pages 3395–3404, 2019. 7

[7] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *ICLR*, 2022. 1, 4

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, page 41–48, 2009. 5

[9] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, pages 561–578, 2016. 3, 5, 14

[10] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *CVPR*, pages 6233–6242, 2017. 5

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 1, 4

[12] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, pages 2272–2281, 2019. 1, 2, 6

[13] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017. 1, 4, 14

[14] Carnegie Mellon University. CMU MoCap Dataset. 5

[15] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1

[16] Ju Yong Chang, Gyeongsik Moon, and Kyoung Mu Lee. Poselifter: Absolute 3d human pose lifting network from a single noisy 2d human pose. *arXiv preprint arXiv:1910.12029*, 2019. 5, 14

[17] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zioulis, Shishir Subramanyam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas, et al. Human4d: A human-centric multimodal dataset for motions and immersive media. *IEEE Access*, 8:176241–176262, 2020. 5

[18] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *CVPR*, 2019. 3

[19] 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 *ICCV*, 2021. 7

[20] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, 2020. 3, 7

[21] Yu Cheng, Bo Yang, Bo Wang, and Robby T Tan. 3d human pose estimation using spatio-temporal networks with explicit occlusion training. In *AAAI*, 2020. 2, 6

[22] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, pages 1964–1973, 2021. 3, 6, 7

[23] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, pages 769–787, 2020. 3, 6, 7, 14

[24] Hai Ci, Xiaoxuan Ma, Chunyu Wang, and Yizhou Wang. Locally connected network for monocular 3d human pose estimation. *IEEE TPAMI*, pages 1–1, 2020. 5, 14

[25] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, pages 2262–2271, 2019. 1, 2, 6

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1, 4, 14

[27] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition, 2022. 14

[28] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, 2022. 3, 6, 7, 14

[29] Mark Endo, Kathleen L. Poston, Edith V. Sullivan, Li Fei-Fei, Kilian M. Pohl, and Ehsan Adeli. Gaitforemer: Self-supervised pre-training of transformers via human motion forecasting for few-shot gait impairment severity estimation. In *MICCAI*, pages 130–139, 2022. 2

[30] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. MoVi: A large multipurpose motion and video dataset, 2020. 5

[31] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018. 1

[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 4, 14

[33] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O'Sullivan. Sleight of hand: Perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2012. 5

[34] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 2014. 1, 5, 6, 14

[35] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010. 7

[36] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *3DV*, pages 42–52. IEEE, 2021. 14

[37] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 3, 6, 7

[38] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, pages 5614–5623, 2019. 2, 3, 5, 7, 14

[39] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 6, 14

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 14

[41] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, pages 5253–5263, 2020. 1, 3, 5, 6, 7, 14

[42] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, pages 2252–2261, 2019. 7, 14

[43] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, pages 4501–4510, 2019. 6

[44] Björn Krüger, Anna Vögele, Tobias Willig, Angela Yao, Reinhard Klein, and Andreas Weber. Efficient unsupervised temporal segmentation of motion data. *IEEE Transactions on Multimedia*, 2017. 2

[45] Bio Motion Lab. BMLhandball Motion Capture Database. 5

[46] Andreas Lehrmann, Peter V. Gehler, and Sebastian Nowozin. Efficient non-linear markov models for human motion. In *CVPR*, 2014. 2

[47] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 3

[48] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, 2022. 2, 6

[49] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *ACM International Conference on Multimedia*, pages 2490–2498, 2020. 2

[50] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 6, 7, 14

[51] Li Linguo, Wang Minsi, Ni Bingbing, Wang Hang, Yang Jiancheng, and Zhang Wenjun. 3d human action

representation learning via cross-view consistency pursuit. In *CVPR*, 2021. 2, 7

[52] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017. 1

[53] 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 TPAMI*, 42(10):2684–2701, 2019. 1, 3, 6, 7

[54] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE TPAMI*, 40(12):3007–3021, 2017. 7

[55] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, pages 1647–1656, 2017. 7

[56] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *CVPR*, 2020. 6

[57] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, pages 143–152, 2020. 3, 7

[58] Matthew Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Transactions on Graphics*, 33(6), Nov. 2014. 5

[59] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015. 1, 3, 5

[60] Eyes JAPAN Co. Ltd. Eyes Japan MoCap Dataset. 5

[61] Zhengyi Luo, S Alireza Golestaneh, and Kris M Kitani. 3d human motion estimation via motion compression and refinement. In *ACCV*, 2020. 7

[62] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, pages 5137–5146, 2018. 3

[63] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, pages 5441–5450, Oct. 2019. 1, 3, 4, 5, 14

[64] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *ICAR*, pages 329–336. IEEE, 2015. 5

[65] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, pages 2640–2649, 2017. 2, 5, 6

[66] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, pages 506–516. IEEE, 2017. 7

[67] Raphael Memmesheimer, Simon Häring, Nick Theisen, and Dietrich Paulus. Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition. In *WACV*, pages 3702–3710, 2022. 7

[68] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus. Sl-dml: Signal level deep metric learning for multimodal one-shot action recognition. In *ICPR*, 2021. 3, 7

[69] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019. 2

[70] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, pages 752–768, 2020. 3, 7, 14

[71] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database HDM05. Technical Report CG-2007-2, Universität Bonn, June 2007. 5

[72] Mahdiar Nekoui and Li Cheng. Enhancing human motion assessment by self-supervised representation learning. In *BMVC*, 2021. 2

[73] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3, 6

[74] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019. 14

[75] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019. 14

[76] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, pages 7307–7316, 2018. 6

[77] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pages 7753–7762, 2019. 2, 3, 4, 5, 6, 14

[78] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020. 1, 4

[79] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017. 14

[80] Alberto Sabater, Laura Santos, Jose Santos-Victor, Alexandre Bernardino, Luis Montesano, and Ana C. Murillo. One-shot action recognition in challenging therapy scenarios. In *CVPR Workshop*, 2021. 3, 7

[81] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016. 1, 6

[82] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, pages 2616–2625, 2020. 1

[83] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 1, 3, 7

[84] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1):4–27, 2010. 5

[85] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, page 4263–4270, 2017. 3

[86] Yang Song, Luis Goncalves, and Pietro Perona. Unsupervised learning of human motion models. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *NeurIPS*, volume 14. MIT Press, 2001. 2

[87] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *CVPR*, pages 9631–9640, 2020. 2

[88] Yukun Su, Guosheng Lin, and Qingyao Wu. Self-supervised 3d skeleton action representation learning with motion consistency and continuity. In *ICCV*, 2021. 2, 7

[89] Jiangxin Sun, Zihang Lin, Xintong Han, Jian-Fang Hu, Jia Xu, and Wei-Shi Zheng. Action-guided 3d human motion prediction. In *NeurIPS*, volume 34, pages 30169–30180. Curran Associates, Inc., 2021. 2

[90] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3, 4, 6, 14

[91] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 2

[92] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, Yili Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *ICCV*, pages 5349–5358, 2019. 3, 7

[93] Fida Mohammad Thoker, Hazel Doughty, and Cees GM Snoek. Skeleton-contrastive 3d action representation learning. In *ACM International Conference on Multimedia*, pages 1655–1663, 2021. 2

[94] Dorra Trabelsi, Samer Mohammed, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. An unsupervised approach for automatic activity recognition based on hidden markov model regression. *IEEE Transactions on Automation Science and Engineering*, 2013. 2

[95] Shashank Tripathi, Siddhant Ranade, Ambrish Tyagi, and Amit K. Agrawal. Posenet3d: Learning temporally consistent 3d human pose via knowledge distillation. In *8th International Conference on 3D Vision*, 2020. 3

[96] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2–2, Sept. 2002. 5

[97] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John P. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*, 2017. 5

[98] 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 *NeurIPS*, 2017. 3

[99] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, pages 601–617, 2018. 6, 14

[100] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. An approach to pose-based action recognition. In *CVPR*, pages 915–922, 2013. 3

[101] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020. 1, 2, 6

[102] Peng Wang, Jun Wen, Chenyang Si, Yuntao Qian, and Liang Wang. Contrast-reconstruction representation learning for self-supervised skeleton-based action recognition. *arXiv preprint arXiv:2111.11051*, 2021. 2

[103] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE TPAMI*, 2022. 5

[104] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *ICCV*, 2021. 6

[105] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *CVPR*, 2021. 6

[106] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, pages 7760–7770, 2019. 3

[107] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 1, 3, 7, 14

[108] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Unik: A unified framework for real-world skeleton-based action recognition. In *BMVC*, 2021. 7

[109] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C. Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *ICCV*, 2021. 7

[110] Zhuoqian Yang, Wentao Zhu, Wayne Wu, Chen Qian, Qiang Zhou, Bolei Zhou, and Chen Change Loy. Transmomo: Invariance-driven unsupervised video motion retargeting. In *CVPR*, 2020. 2

[111] Angela Yao, Juergen Gall, and Luc Gool. Coupled action recognition and pose estimation from multiple views. *Int. J. Comput. Vision*, page 16–37, 2012. 3

[112] Raymond Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. In *NeurIPS*, 2019. 6

[113] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, June 2020. 14

[114] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *CVPR*, 2019. 2

[115] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, pages 11446–11456, 2021. 3, 7

[116] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *CVPR*, 2022. 2, 4, 6

[117] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *ICCV*, 2021. 2, 6

[118] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *ICCV*, 2019. 2

[119] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 14

[120] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016. 3

[121] Wentao Zhu, Zhuoqian Yang, Ziang Di, Wayne Wu, Yizhou Wang, and Chen Change Loy. Mocanet: Motion retargeting in-the-wild via canonicalization networks. In *AAAI*, 2022. 2

A. Appendix

A.1. Experimental Details

Setup. We implement the proposed model with PyTorch [74]. All the experiments are conducted on a Linux machine with 8 NVIDIA V100 GPUs, which is intended for accelerating pretraining. A single GPU is usually sufficient for finetuning and inference.

Pretraining. For AMASS [63], we first render the parameterized human model SMPL+H [79], then extract 3D keypoints with a pre-defined regression matrix [9]. We extract 3D keypoints from Human3.6M [34] by camera projection [24]. We sample motion clips with length $T = 243$ for 3D mocap data. For 2D data, we utilize the provided annotations of PoseTrack [3]. We further include 2D motion from an unannotated video dataset InstaVariety [38] extracted by OpenPose [13]. Since the valid sequence lengths for in-the-wild videos are much shorter, we use $T = 30$ (PoseTrack) and $T = 81$ (InstaVariety). We set the input channels $C_{\text{in}} = 3$ (x, y coordinates and confidence) following [28, 107]. Random horizontal flipping is applied as data augmentation. The whole network is trained for 80 epochs with learning rate 0.0005 and batch size 64 using an Adam [40] optimizer. The weights of the loss terms are $\lambda_O = 20$. We set the 2D skeleton masking ratio = 15%, same as BERT [26]. More specifically, we use 10% frame-level masks and 5% joint-level masks. We vary the proportion of the two types of masks and only observe marginal differences. We follow [16] to fit a mixture of distributions and sample per-joint noises from it. To keep the noise smooth and avoid severe jittering, we first sample the noise $\boldsymbol{z} \in \mathbb{R}^{T_K \times J}$ for $T_K = 27$ keyframes, then upsample it to $\boldsymbol{z}' \in \mathbb{R}^{T \times J}$, and add a small temporal gaussian noise $\mathcal{N}(0, 0.002^2)$.

3D Pose Estimation. The 2D skeletons are provided by 2D pose estimator trained on MPII [4] and Human3.6M [34] following the common practice [24, 77]. For training from scratch, we train for 60 epochs with learning rate 0.0005 and batch size 32. For finetuning, we load the pretrained weights and train with learning rate 0.0003 and batch size 32.

Skeleton-based Action Recognition. We use the 2D skeleton sequences extracted with HRNet [90] provided by PYSKL [27]. We then upsample the skeleton sequences to a uniform length $T = 243$. For NTU-RGB+D, the DSTformer output after global average pooling is fed into an MLP including dropout $p = 0.5$, a hidden layer of 2048 channels, BatchNorm, and ReLU. We train for 200 epochs with learning rate 0.001 and batch size 32 for training from scratch. We set learning rate 0.0001 for the backbone, and learning rate 0.001 for the downstream MLP for finetuning. For one-shot recognition on NTU-RGB+D-120, we apply dropout $p = 0.1$, and use a linear layer to get action representation of size 2048. We train with batch size 16, and each batch includes 8 action pairs. Samples from the same action class are set as positives against the negatives from the remainder of the batch. We use the supervised contrastive loss [39] with temperature 0.1.

Human Mesh Recovery. For the Human3.6M dataset, we use the same 2D skeleton sequences as the 3D pose estimation task. The SMPL ground-truth (GT) parameters are fitted with SMPLify-X [75] provided by [70]. For the 3DPW dataset, we obtain its detected 2D skeleton sequences from DarkPose [113] provided by [23]. The SMPL GT parameters of 3DPW are obtained using IMUs [99]. For 3DPW benchmark, we use Human3.6M and COCO [50] datasets for training, following [23]. The pseudo-GT SMPL parameters of COCO are provided by [36]. 3DPW training data is used optionally according to different settings. Following [41, 42], we use the 6D continuous rotation representations [119] instead of original axis angles when estimating pose parameters. The shape and pose MLPs are the same, including one hidden layer of 2048 channels, BatchNorm, and ReLU. We sample motion clips with length $T = 81$ and stride 27 for both datasets. We train the whole network end-to-end for 140 epochs with batch size 32, with learning rates 0.0001 for the backbone, and 0.001 for the downstream network, respectively.

A.2. Additional Ablation Studies

Finetune vs Scratch. We compare the training progress of finetuning the pretrained model and training from scratch. As Figure 3 shows, models initialized with pretrained weights enjoy better performance and faster convergence on all three downstream tasks. It indicates that the model learns transferable knowledge about human motion during pretraining, facilitating the learning of multiple downstream tasks.

Mask Ratio. We study the effect of different masking ratios on both pretraining and downstream finetuning. We measure the performance by MPJPE on the Human3.6M test set. Frame-level mask and joint-level mask are fixed to 2 : 1. As Figure 4 shows, when the masking ratio is less than 45%, the pretrained models could learn 2D-to-3D lifting well, and pretraining improves the finetuning performance (better than the scratch baseline). As we continue to increase the masking ratio, the pretrained models perform much worse at 2D-to-3D lifting, and pretraining is no longer beneficial for the finetuning performance. Our observation is slightly different from MAE [32] where a high masking ratio (75%) works well. One possible explanation is that

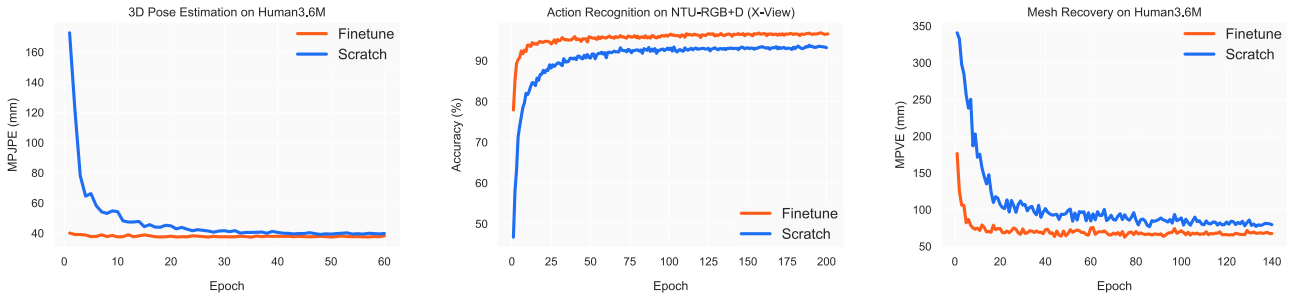

Figure 3. **Learning curves of finetuning and training from scratch.** We visualize the training process on the three downstream tasks respectively.

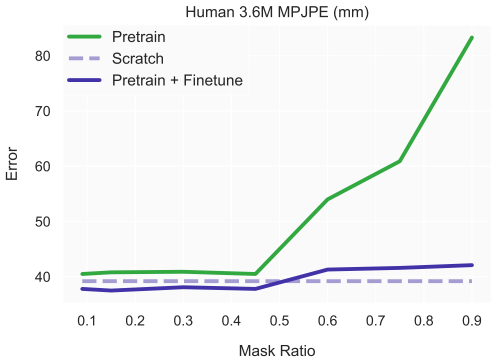

Figure 4. **Ablation study of pretraining mask ratios.** For each mask ratio, We test the Human3.6M MPJPE of the pretrained models with/without finetuning respectively.

skeletons are highly abstract representations of human motion. A high masking ratio leaves too few cues to complete the whole sequence, therefore the model tends to overfit. It is also worth noting that in our pretraining task (2D-to-3D lifting), the depth channel ($\frac{1}{3}$ of original data) is intrinsically masked.