An Animation-based Augmentation Approach for Action Recognition from Discontinuous Video

Xingyu Song, Zhan Li, Shi Chen, Xin-Qiang Cai and Kazuyuki Demachi

The University of Tokyo

Abstract. Action recognition, an essential component of computer vision, plays a pivotal role in multiple applications. Despite significant improvements brought by Convolutional Neural Networks (CNNs), these models suffer performance declines when trained with discontinuous video frames, which is a frequent scenario in real-world settings. This decline primarily results from the loss of temporal continuity, which is crucial for understanding the semantics of human actions. To overcome this issue, we introduce the 4A (Action Animation-based Augmentation Approach) pipeline, which employs a series of sophisticated techniques: starting with 2D human pose estimation from RGB videos, followed by Quaternion-based Graph Convolution Network for joint orientation and trajectory prediction, and Dynamic Skeletal Interpolation for creating smoother, diversified actions using game engine technology. This innovative approach generates realistic animations in varied game environments, viewed from multiple viewpoints. In this way, our method effectively bridges the domain gap between virtual and real-world data. In experimental evaluations, the 4A pipeline achieves comparable or even superior performance to traditional training approaches using real-world data, while requiring only 10% of the original data volume. Additionally, our approach demonstrates enhanced performance on In-the-wild videos, marking a significant advancement in the field of action recognition.

1 Introduction

Action recognition is a critical component of computer vision that involves identifying and classifying various actions from sequences of images or video frames. This task is essential across numerous applications, including malicious behavior identification, accident detection, and human-computer interaction [22, 38]. The significant advancements in Convolutional Neural Networks (CNNs) notably enhances performance in action recognition tasks across a range of benchmark datasets [5, 47, 39, 8].

However, when encountering scenarios with discontinuous frame sequences during training, which is a common occurrence in real-world settings, CNN-based models implemented for action recognition suffer in a significant performance decline. For instance, with continuous frame training videos achieving around 40% mean accuracy, while training with missing frames drops to below 20% (refer to Section 4 for details). On the other hand, other CNN-implemented tasks related to human motion, such as pose estimation [52, 31, 53, 31] or 3D human reconstruction [42, 26, 19], **do not** exhibit such severe performance declines with discontinuous frames [45]. For instance, [27] demonstrates less than 1% of drop in performance when training on the dataset with extracted frames compared to training on the original frame sequence from H3WB dataset [55]. Unlike these tasks, action recognition fundamentally involves a deeper **semantic** analysis. It requires the interpretation and understanding of human motion patterns, essentially deciphering the meanings or semantics behind those actions [33]. Therefore, the absence of temporal information due to missing frames directly diminishes the understanding of an action, making the action recognition task susceptible to the continuity of the video. On the other hand, the loss of semantics from the original data complicates the process of augmenting the dataset as well.

Inspired by the previous studies on data augmentation in other computer vision tasks [45, 42, 46], we propose to use synthetic human to mitigate the issue of missing frames in action recognition tasks. In this study, we introduce the 4A (Action Animation-based Augmentation Approach) pipeline, an innovative, efficient, and scalable pipeline for data augmentation within the action recognition field. This approach achieves the generation of smooth and realistic (natural-looking) synthetic human motions (termed animations), depicted across a variety of settings, appearances, and conditions from multiple viewpoints, leveraging discontinues monocular RGB videos from real world. The detailed pipeline of 4A is illustrated in Figure 1. Furthermore, we conduct experiments to evaluate the effectiveness of 4A in bridging the domain gap between virtual representations and real-world tasks.

The main contributions of our work include: (1) we discover the problem of severe decrease on performance of action recognition task training by discontinuous video, and the limitation of existing augmentation methods on solving this problem. (2) we propose a novel augmentation pipeline, 4A, to address the problem of discontinuous video for training, while achieving a smoother and much more natural-looking action representation than the latest data augmentation methodology. (3) We achieve the same performance with only 10% of the original data for training as with all of the original data from the real-world dataset, and a better performance on In-the-wild videos, by employing our data augmentation techniques.

2 Related Work

2.1 *Synthetic Human Construction*

Recent advancements have seen synthetic images of humans deployed to train visual models for tasks like 2D or 3D body pose and shape estimation [6, 13], part segmentation [41, 45], and person re-identification [34]. However, synthetic datasets built for these tasks often lack action labels, limiting their applicability for action recognition tasks.

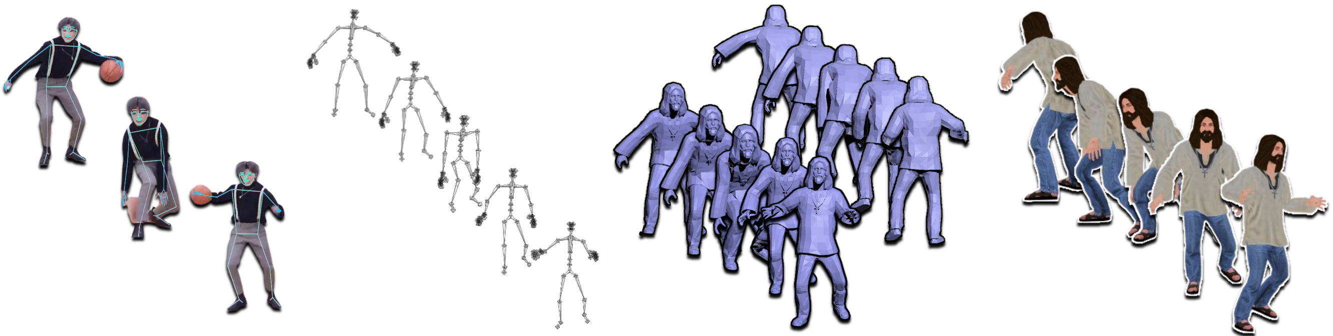
Real-world RGB frames with skeleton-based 2D pose estimation
Orientation-based skeletal representations
Game engine-based 3D motion representations
Multi-viewpoint animation presentations

Figure 1. Overview of 4A pipeline. Within 4A pipeline, we begin with a 2D human pose estimation method to extract the 2D coordinates of human skeleton coordinates from real-world RGB videos. This is followed by employing a Quaternion-based Graph Convolution Network (Q-GCN) to predict the orientation of each bone joint and the trajectory of body in 3D space. Subsequently, the Dynamic Skeletal Interpolation algorithm (DSI) ensures a smoother and more diversified action animation. After that, we use the game engine technology to generate the motion from skeleton representation sequence to form the animation. Finally, we present the animation in game environment with diverse environments and appearances, and captured in multiple viewpoints.

Prior researches leveraging synthetic human data for action recognition are few [7, 23], with some researches focusing on synthetic 2D human pose sequences [28] and point trajectories [35] for view-invariant action recognition. RGB-based synthetic training data for action recognition is a field under exploration, with few attempts addressing the manual definition of action classes for multitask learning [7]. However, scalability and relevance to target classes remain challenges in these approaches.

A study more aligned with our work [23] uses synthetic training images derived from RGB-D inputs to enhance performance on unseen viewpoints, framing a pose classification problem that serves as a basis for action recognition. Yet, the discriminative power of these features for specific action categories is questionable. In contrast, another approach [46] extracts motion sequences directly from real data, offering flexibility for incorporating new categories and assigning explicit action labels to synthetic videos. This method, however, suffers from mismatches between characters and their environments, alongside issues with action smoothness and photorealism. This is primarily due to the limitations of 3D human shape estimation or reconstruction technologies [26, 19], especially in handling videos with discontinuous frames.

2.2 Game Engine-based Action Datasets.

Video game-based datasets have been increasingly utilized for training deep learning models. JTA [9], for example, is a vast dataset created using a video game for pedestrian pose estimation and tracking in urban environments. GTA-IM [4], a pose estimation dataset, highlights human-scene interactions and employs a developed game engine interface for automatic control of characters, cameras, and actions. However, both them focus on static human poses, rather than capturing temporal movements with semantics. NCTU-GTA360 [1], an action recognition dataset featuring spherical projection captured from video games, involves the use of 360-degree cameras to record the entire surroundings of a character. Nevertheless, NCTU-GTA360 faces an imbalance in action distribution, with basic actions like “OnFoot” or “Stopped” overwhelmingly dominating more complex actions such as “Ragdoll” or “SwimmingUnderWater”. Other datasets like SIM4ACTION [36] and G3D [3] also share a common limitation in offering a constrained range of action classes. Most notably, none of these datasets, including GTA-IM and NCTU-GTA360, have succeeded in effectively importing a vast amount of self-customized actions from the real world, which is crucial for creating more comprehensive and diverse training models.

2.3 3D Skeletal Motion Representation

3D Human Pose Estimation (HPE) in videos, aiming to predict human body joint locations in 3D space, employs methods like single-stage and 2D to 3D lifting. While single-stage methods estimate 3D pose directly from images [29, 54], 2D to 3D lifting [31, 24], using ground truth 2D poses, generally performs better. However, both approaches face challenges in providing smooth motion representations from discontinuous frames, indicating a gap in effectively handling interrupted sequences for accurate 3D skeletal motion representation.

2.4 3D Human Shape Estimation

The Skinned Multi-Person Linear model (SMPL) [26] represents a pivotal approach in human motion capture, marking the first introduction of orientation-based human body representation. Subsequent developments inspired by SMPL, including models like MANO [37], SMPL-X [26], and STAR [30], have expanded the framework’s utility to encompass detailed body shape modeling, facial expressions, hand movements, and the representation of clothed human bodies. Despite these advancements, similar to challenges in 3D pose estimation, the performance of human shape estimation models is constrained by their ability to express the semantics of motion, particularly when trained on limited datasets.

3 The 4A Pipeline

The goal of 4A pipeline is to improve the performance of action recognition using synthetic data especially when the training videos are discontinuous or insufficient. There are four stages in 4A: (1) 2D skeleton extraction; (2) 3D orientation lifting; (3) Sequence smoothing; (4) Animation generation and capturing;

3.1 2D Skeleton Extraction

In this stage, we utilize a 2D human skeleton estimation technique, HRNet [43], trained on COCO-WholeBody dataset [18], to extract 2D human wholebody skeleton keypoints from the RGB frames of monocular videos. For more semantic representation, we construct a hierarchical structure of human skeleton following the biovision hierarchy file format inspired by [16]. However, unlike [16], which only contains 17 joint nodes and 16 bone joints, our configuration contains 54 node joints and 53 bone joints to form a more detailed whole-body motion including hands, feet and neck. Furthermore, the upper and lower body respectively form an hierarchical inheritance structure (start from pelvis), extending from the parent joint to the child joint of both each node joint and bone joint, where the bone joint can be regraded as a vector while the node joint is the start and end point of it.

3.2 3D Orientation Lifting

In this stage, we attempt to lift 2D skeletal representation into 3D space to present motion in multiple viewpoints. However, due to the unsatisfying performance of 3D pose and shape estimation (mentioned in Section 2.3 and 2.4), which are unable to provide viable motion representations in 3D space. It's important to note that, unlike 3D pose estimation, the reconstruction of synthetic human motion does not require precise joint coordinates; approximate dynamics of human motion are sufficient. Drawing inspiration from the SMPL model [26], we extend our hierarchical human skeleton structure by using the coordinates of the root node (pelvis) and the orientations of other bone joints to form the skeleton representation in 3D space in each frame. Consequently, predicting the 3D space orientation of each bone joint becomes our primary challenge.

Prior research utilizing Graph Convolution Network (GCN) for human pose estimation [52, 31] and action recognition [50, 21, 40] demonstrates the powerful capability of GCNs in extracting human dynamic features. Influenced by the techniques in [16] and [31], we develop a Quaternion Graph Convolution Network (Q-GCN) to predict the Quaternions (used to representing orientation) and the root 3D coordinates for each bone joint from 2D coordinates of each skeleton keypoint.

3.2.1 Q-GCN

Similar as [40, 16, 50, 21, 53], we first construct a graph of human body skeleton following our structure. The vertices of these graphs comprise the sequence of human poses within the 2D coordinate space, denoted as $P_{2D} = \{\mathbf{X}_{t,j} \in \mathbb{R}^2 | t = 1, 2, \ldots, T; j = 1, 2, \ldots, J\}$, where $\mathbf{X}_{t,j}$ represents the 2D coordinates of joint node j at frame t. Here, T and J respectively signify the number of frames in the sequence and the number of joints in the human skeleton. Differ from prior implementations of GCNs, we also compile a sequence of rotations for each bone joint within the 2D coordinate space to constitute the edges of graph, expressed as $R_{2D} = \{\mathbf{Z}_{t,b} \in \mathbb{R}^2 | t = 1, 2, 3, \ldots, T; b = 1, 2, 3, \ldots, B\}$, where B represents the number of bone joints and $\mathbf{Z}_{t,b}$ includes a 2-tuple consisting of the cosine and sine values of the rotation angle for bone joint b from the initial position in Local Coordinate System (LCS). Note that the initial position in LCS of a bone joint is defined by its parent node and bone joint in the hierarchical structure, overlapping with the extension of parent bone joint from the parent node as origin point, where the rotation angle $\theta \in [-\pi, \pi]$.

Formally, this temporal sequence of graphs is articulated as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_{t,j} | t = 1, 2, \ldots, T; j = 1, 2, \ldots, J\}$, and $\mathcal{E} = \{e_{t,b} | t = 1, 2, \ldots, T; b = 1, 2, 3, \ldots, B\}$, are the sets of vertices and edges respectively. Note that the features of vertex $v_{t,j}$ and edge $e_{t,b}$ are initialized with their corresponding 2D coordinates $\mathbf{X}_{t,j}$ and rotation $\mathbf{Z}_{t,b}$.

Subsequently, we employ our Q-GCN to predict the sequence of Quaternions $Q_{4D} = \{\mathbf{Q}_b \in \mathbb{R}^4 | b = 1, 2, 3, \ldots, B\}$ and the root coordinate in the 3D space $\mathbf{P}_{root} \in \mathbb{R}^3$, serving as the representation of orientation. Similar as rotations in 2D system, orientations in 3D spaces are also defined within LCS. This design is employed to enhance the understanding of the internal dynamics and influence exerted from parent nodes to child nodes according to the previous research [16].

Similar as [50], we first implement a basic spatial-temporal graph convolution block to extract the feature within the graph. We define a neighbor set $\mathcal{B}_j^v$ as a spatial graph convolutional filter for vertex $v_{t,j}$ while set $\mathcal{B}_b^e$ for edge $e_{t,b}$. Inspired by [52], both for vertex and edge filters, we define four distinct neighbor subsets: (1) self, (2) parent, and (3) child. Therefore, the kernel size K is set to 3, corresponding to the 3 subsets. To implement the subsets, mappings $h_{t,j}^v \rightarrow \{0, \ldots, K-1\}$ and $h_{t,b}^e \rightarrow \{0, \ldots, K-1\}$ are used to index each subset with a numeric label. Therefore, this convolutional operations of vertex and edge can be written as

$$f_{out}^v(v_{t,j}) = \sum_{v_{t,n_j} \in \mathcal{B}_j^v} \frac{1}{Z_{t,n_j}} f_{in}^v(v_{t,n_j}) W_v(h_{t,j}^v(v_{t,n_j})) \quad (1)$$

$$f_{out}^e(e_{t,b}) = \sum_{e_{t,n_b} \in \mathcal{B}_b^e} \frac{1}{Z_{t,n_b}} f_{in}^e(e_{t,n_b}) W_e(h_{t,b}^e(v_{t,n_b})) \quad (2)$$

where $f_{in}^v(v_{t,n_j}) : v_{t,n_j} \rightarrow \mathbb{R}^2$ and $f_{in}^e(e_{t,n_b}) : e_{t,n_b} \rightarrow \mathbb{R}^2$ denote the mappings that get the attribute feature of neighbor node joint v_{t,n_j} and neighbor bone joint e_{t,n_b} respectively. Z_{t,n_j} and Z_{t,n_b} is the normalization term that equal to the subset's cardinality. $W(h_{t,j}(v_{t,n_j}))$ and $W(h_{t,b}(v_{t,n_b}))$ are the weight functions of mapping $\mathcal{B}_j^v$ and $\mathcal{B}_b^e$ respectively, which are implemented by indexing a $(2, K)$ tensor. Within a pose frame, the determined graph convolution of a sampling strategy can be implemented by adjacent matrices of $J \times J$ for vertex and $B \times B$ for edge. Specifically, with K spatial sampling strategies $\sum_{k=0}^{K-1} A_k^v$ for vertex and $\sum_{k=0}^{K-1} A_k^e$ for edge, Equation 1 and 2 can be transformed into the expressions using matrices into:

$$H_t^v = \sum_{k=0}^{K-1} \bar{A}_k^v F_k^v W_k^v \quad (3)$$

$$H_t^e = \sum_{k=0}^{K-1} \bar{A}_k^e F_k^e W_k^e \quad (4)$$

Where $\bar{A}_k = \Lambda_k^{\frac{1}{2}} A_k \Lambda_k^{\frac{1}{2}}$ is the normalized adjacency matrix of A_k both for vertex and edge, with its elements indicating whether a vertex v_{t,n_v} or a edge e_{t,n_e} is included in the neighbor subset. Similar as [20], $\Lambda_k^{ii} = \sum_n (\bar{A}_k^{in}) + \alpha$ is a diagonal matrix with α set to 0.001 to prevent empty rows. W_k denotes the weighting function of Equation 1 and 2, which is a weight tensor of the 1×1 convolutional operation. F_k is the attribute features of all the neighbor joints sampled into the subset k. Therefore, a convolution layer in Q-GCN is realized with a $1 \times T$ classical 2D convolution layer, where T is the temporal kernel size that we set to 10. And the output of layer H_t is both followed by a batch normalization layer and a ReLU layer and a dropout layer after them to form a convolutional block. In addition, a residual connection [15] is added as well.

The whole architecture of Q-GCN is shown in Figure 2. Inspired by [31], we also construct a trajectory prediction block to predict the global position of root node in a sequence. Finally, the Quaternion of each bone joint can also be predicted.

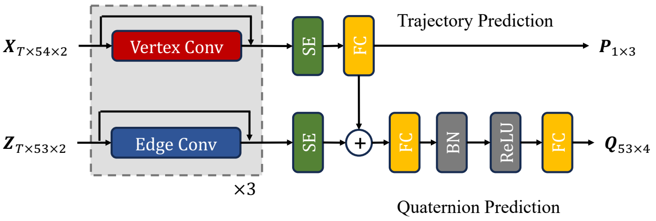

Figure 2. Whole architecture of Q-GCN. It starts with three vertex and edge convolutional blocks, with residual connection operation in each block. After extract the neighbor features, both layers are followed by a Squeeze and Excitation (SE) Block. Then, the concatenation of vertex graph and edge graph is followed by two fully connection layer with batch normalization and ReLU function in between.

As for the loss function for the trajectory prediction, we adopt the weighted mean per-joint position error [31] (WMPJPE). As for Quaternion prediction, we develop our Average Angular Distance (AAD) loss function to minimize the angular distance between the ground truth and predicated value:

$$\mathcal{L}_{angular} = \frac{1}{T}\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B} 2\arccos\left(Re(\bar{\boldsymbol{Q}}_{t,b} \times conj(\boldsymbol{Q}_{t,b})\right) \quad (5)$$

Where $\bar{\mathbf{Q}}_{t,b}$ and $\mathbf{Q}_{t,b}$ stand for the ground truth and the predicted value of Quaternion of bone joint b at frame t. And the functions $Re(\cdot)$ and $conj(\cdot)$ return the real part and the conjugate of a Quaternion respectively. In addition, we have also proposed a whole-body 3D orientation lifting dataset for training Q-GCN, derived from H3WB [55].

3.3 *Sequence Smoothing*

After assembling the Quaternion of each bone joint for an individual frame, we proceed to compile them to create a continuous sequence across a stream of frames to depict a complete representation of action. However, directly combining Quaternion sequence will lead to frequent mismatches and jitters among the representation. Unlike typical time series data, directly applying a piecewise polynomial interpolation algorithm to orientation sequences can lead to significant issues, such as missing specific poses and decreased movement amplitude (detailed in Section 4).

To tackle these challenges, we have developed the Dynamic Skeletal Interpolation (DSI) algorithm. This algorithm dynamically segments the Quaternion sequence into unit motions based on the variation in the range of motion. It then automatically interpolates the Quaternion sequence across different frame counts, ensuring a smoother and more natural-looking animation. Finally, it randomly generates a series of variants for each sequence, ensuring a diverse representation.

The algorithm is detailed in Algorithm 1. Here, $\mathbf{Q}$ stands for the sequence of Quaternion, While $\mathbf{Q}'$ is the sequence after interpolation. d^f denoted the weighted average of the Angular Distance of each bone joint between adjacent frames, with w_b representing the weight of each bone joint. The functions $Re(\cdot)$ and $conj(\cdot)$ return the real part and the conjugate of a Quaternion respectively. $p(x)_{[a,b]}$ refers to the Lagrange interpolating polynomial in the interval $[a, b]$, while $L(x)$ represents the Lagrange basis polynomial. $Linespace([a, b], n)$ is the function used to create evenly space numbers over interval $[a, b]$. The parameter δ is the interpolation rate, indicting the ratio of original frames to interpolated frames. η is the interpolation coefficient. To enhance the diversity of an action, we develop Random Variation function $\mathcal{V}(\cdot)$ to generate a series of variants for each sequence. To smooth the piecewise interpolated data, we employ Supersmoother [12] $\mathcal{S}(\cdot)$, a non-parametric smoothing method. V, B, F, F' denote the number of variants, bone joints, frames before and after interpolation.

Algorithm 1 Dynamic Skeletal Interpolation

```
Input: $\boldsymbol{Q} \in \mathbb{R}^{F\times B\times4}$
Define $\boldsymbol{Q}' \in \mathbb{R}^{F'\times B\times4}$
for $f = 2$ to $F$ do
    $d^f = \frac{1}{J}\sum_{b=1}^{B} w_b \cdot 2\arccos\left(Re(\boldsymbol{q}_b^f \times conj(\boldsymbol{q}_b^{f-1})\right)$
    if $d^f > threshold$ then
        $p_{[i,f-1]}(x) = \sum_{b=1}^{B} w_b \sum_{t=i}^{f-1} \boldsymbol{q}_b^t \cdot L_t(x)$
        $p_{[f-1,f]}(x) = \sum_{b=1}^{B} w_b(\boldsymbol{q}_b^{f-1} \cdot L_{f-1}(x) + \boldsymbol{q}_b^f \cdot L_f(x))$
        $\boldsymbol{x}_{nor} = Linespace([i, f-1], \frac{1}{\delta})$
        $\boldsymbol{x}_{edge} = Linespace([f-1, f], Int(\frac{\eta\cdot d^f}{\delta}))$
        $\boldsymbol{A}'_{[i,f-1]} = p_{[i,f-1]}(\boldsymbol{x}_{nor})$
        $\boldsymbol{A}'_{[f-1,f]} = p_{[f-1,f]}(\boldsymbol{x}_{edge})$
        $i = f - 1$
    end if
end for
$\boldsymbol{\Phi} = Supersmoother(\mathcal{V}(\boldsymbol{Q}', V))$
Output: $\boldsymbol{\Phi} \in \mathbb{R}^{V\times F'\times B\times4}$
```

3.4 *Animation Generation and Capturing*

In this stage, we adopt the game engine 3DS Max [2] to generate the mesh from the sequence of skeleton representation, to further form the animation. The action animations are then showcased using FiveM [11], a modification platform for GTAV, enabling players to play multi-players on customized dedicated server, in multiple viewpoints. We also adopt scene customization to craft scenes for these action animations in FiveM, which accommodate unique action situations and enhanced data diversity. Environmental customization involves altering weather conditions, time of day, and in-game locations. Through FiveM scripts, we facilitate random weather variations, time adjustments, and repositioning across different in-game locales to achieve varied scenarios. Character customization facilitates modifying both native FiveM characters and player-created avatars, enabling the simulation of actions by individuals in diverse ages, genders, and professions, reflecting the variety found in real-world scenarios. Map customization enables editing of the in-game landscape and the attributes of entities, which also supports the integration of player-created custom entities.

4 Experimental Results

In this section, we first access our pipeline's capacity of generating sophisticate semantic representations of human motion from discontinues video. This includes comparisons with both real-world data and synthetic data produced by the state-of-the-art (SOTA) methods. Next, we conduct a component-wise comparative analysis to highlight the significance of each critical stage within our pipeline. Finally, we demonstrate our approach using in-the-wild videos.

4.1 Representation Evaluation

To evaluate the efficacy of the 4A pipeline in representing human motion, we execute a series of comparative experiments, comprising both Qualitative and Quantitative Experiments. These experiments cover representations of human motion for both major-part (focusing on essential bone joints excluding the hands and feet) and whole-body. It's important to note that the major-body representation for evaluation is augmented from the NTU-RGB+D [39] dataset, while the whole-body representation is derived from the Human3.6M (H36M) [17] dataset.

4.1.1 Qualitative Experiment

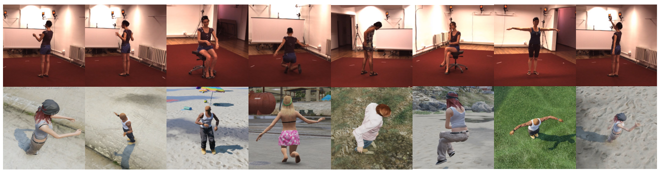

Figure 4. Qualitative results of whole-body representation by 4A in multiple viewpoints, comparing with the original RGB frames in H36M.

The synthetic representation of human motion derived by NTU-RGB+D (comparing with the synthetic representation from SURREACT) and H36M (comparing with the real-world video) are depicted in Figure 3 and 4.

4.1.2 Quantitative Experiment

This evaluation focuses on comparing the effectiveness of models trained on datasets generated by 4A pipeline from discontinuous videos against those trained on datasets generated by other SOTA approaches, or corresponding real-world datasets for human action recognition tasks.

For benchmarking, we establish **NTU-Original** as a baseline training dataset, incorporating all 49 single-person action classes from the NTU RGB+D dataset, exclusively containing continuous video footage from the real world. To mitigate the impact of multiple viewpoints, we selectively use videos where the human is positioned directly in front of the camera (at 0 degrees). In contrast, **NTU-4A**, a dataset created using the 4A process from NTU-Original videos, comprises only synthetic videos derived from discontinuous frames. To assess 4A's proficiency in capturing semantic information from discontinuous videos, we employ a **Random Extracting Strategy** (RES). This strategy involves initially extracting one frame from every five frames of each action video, followed by a random extraction of 50% of these frames, ensuring the avoidance of isolated single frames, to simulate discontinuous video conditions. It's noteworthy that, post-RES, the remaining frames constitute only 10% of the original video frames. Comparative training datasets, **NTU-HMMR** and **NTU-VIBE**, also generated from NTU-Original utilizing SOTA methods as described in SURREACT [46], apply the RES to simulate similar conditions. Benchmark results for various action recognition models, evaluated on a real-world video subset of the NTU dataset named **NTU-Test**, are detailed in Table 1.

Human Whole-body Motion Representation by 4A includes the comprehensive modeling of all 53 bone joints. For benchmark purposes, We utilize the Human3.6M [17] (H36M) for comparison in whole-body motion representation. The H36M dataset offers two key advantages for augmentation: multiple viewpoints and precise 2D coordinate annotations. From H36M, we derive four baseline datasets of action recognition task training, as follows:

1. **H36M-Original**: This dataset consists of complete action videos from "S1", "S5", "S6", "S7" sessions of the H36M dataset, providing four distinct views for each action.
2. **H36M-Single**: A subset of H36M-Original, while only single view for each action is randomly selected.
3. **H36M-Extracted**: From H36M-Original, this dataset is formed by adopting RES process to create a discontinuous dataset.
4. **H36M-SingleExtracted**: Applying RES process as H36M-Extracted, but starting from the H36M-Single dataset, to create a version that is both single-view and sparsely sampled.

Additionally, for the purpose of testing, two specific datasets derived from different sections of the H36M dataset were used:

1. **H36M-Original-Test**: This test dataset includes all videos from sections "S8", "S9", "S11" of H36M, featuring each action captured from four distinct views.
2. **H36M-Segment-Test**: In this dataset, every video from H36M-Original-Test has been manually segmented into individual unit actions, with the removal of indistinguishable clips.

For our experimental purposes, we develop the **H36M-4A** dataset for training. This dataset is derived from each frame of the H36M-SingleExtracted dataset, utilizing the 4A framework for generating

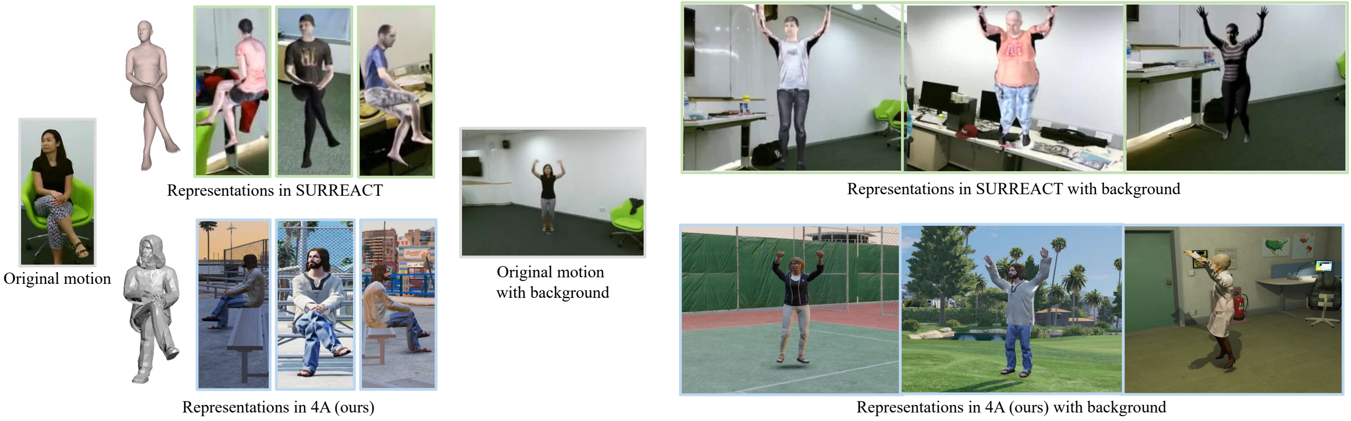
Representations in SURREACT
Original motion
Representations in 4A (ours)
Original motion with background
Representations in SURREACT with background
Representations in 4A (ours) with background

Figure 3. Qualitative results of major-part representation derived from NTU-RGB+D dataset, comparing SURREACT with 4A. Our pipeline outperforms in terms of fidelity and realism in motion representation and excels in depicting character details. Furthermore, it achieves superior integration of characters within their environments, along with enhanced lighting and scene coverage.

Model Info	NTU-Original			NTU-4A (ours)			NTU-HMMR			NTU-VIBE		
Method	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**
Random	-	-	2.0	-	-	2.0	-	-	2.0	-	-	2.0
VideoMAE [44]	82.7	86.1	85.7	82.6	86.4	79.3	32.6	71.4	30.1	31.7	70.9	32.3
TANet [25]	81.6	87.4	79.5	77.6	86.2	74.6	34.5	72.3	32.0	35.6	73.2	34.5
TPN [51]	86.0	90.6	78.9	80.2	86.5	80.2	37.1	69.7	39.0	38.9	71.1	40.3
X3D [10]	78.2	85.2	80.9	68.8	73.3	65.0	27.9	56.9	32.1	30.0	59.7	33.1
I3D [5]	76.0	79.7	74.3	70.1	76.3	70.5	25.1	60.4	21.5	26.7	61.0	22.6
I3D NL [48]	76.9	82.2	76.9	70.3	77.4	75.4	30.2	55.7	28.4	31.1	56.8	29.8

Table 1. Benchmark results on NTU-based datasets. NTU-4A, the dataset created by our pipeline from only 10% of the frames of the NTU-Original dataset, manages to sustain a comparable accuracy when models are trained with real-world continuous videos. This performance is notably superior when compared to NTU-HMMR and NTU-VIBE.

Model Info	H36M-Original			H36M-Single			H36M-Extracted			H36M-SingleExtracted			H36M-4A (Ours)		
Method	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**
Random	-	-	6.7	-	-	6.7	-	-	6.7	-	-	6.7	-	-	6.7
VideoMAE [44]	40.2	79.2	37.5	24.7	77.8	22.5	15.3	50.7	17.2	11.4	45.9	12.5	44.5	88.2	45.0
TANet [25]	33.3	75.0	34.4	25.5	77.5	19.7	11.6	49.5	12.4	13.9	50.1	14.0	43.9	83.4	46.6
TPN [51]	35.8	78.1	33.6	22.8	69.4	20.1	11.4	55.3	11.4	8.9	48.3	8.9	51.4	86.7	45.4
X3D [10]	32.1	67.7	29.6	27.9	75.7	20.3	11.2	47.3	13.2	13.3	46.5	12.6	37.3	80.2	36.5
I3D [5]	28.7	70.6	30.6	20.4	69.5	19.5	11.7	50.0	12.4	10.5	58.5	9.6	32.5	75.6	30.6
I3D NL [48]	34.2	80.5	32.4	23.3	68.6	19.6	11.9	50.5	11.9	9.8	60.1	11.1	40.6	78.9	39.7

Table 2. Benchmark results on **H36M-Original-Test**. In addition to a notable improvement over H36M-SingleExtracted, the original dataset used for generating H36M-4A, our method enables H36M-4A to achieve results that surpass even those of H36M-Original, considering H36M-4A uses only 2.5% of the number of frames present in H36M-Original as input.

whole-body representations. The primary objective is to assess 4A's capabilities of multi-viewpoint data generation and frame interpolation. The benchmark results on various action recognition methods, evaluated using H36M-Original-Test and H36M-Segment-Test as test datasets, are documented in Table 2 and Table 3 respectively.

4.2 Component-wise Comparative Analysis

4.2.1 Q-GCN

We initiate our evaluation by performing comparative experiments on the task of lifting 2D coordinates to 3D orientations. Due to the limited exploration in this area [32], we employ our mean Average Angular Distance (mAAD) loss for evaluation. All baseline methods included in the comparison are specifically tailored for the 2D to 3D pose lifting task. The models are both trained and evaluated on the H3WB [55] dataset. The results of these comparisons are detailed in Table 4.

4.2.2 DSI

In this section, we assess the performance of our Dynamic Skeletal Interpolation (DSI) compared to other interpolation methods. We introduce the Absolute Angular Distance (AAD) metric to measure the angular distance from the current position to the initial position, with the initial position set along the x-axis where the Quaternion equals 1. This metric allows us to track the fluidity of motion across a sequence of frames. Furthermore, to evaluate DSI's efficacy in segmenting Quaternion sequences, we employ a strategy similar to the Random Extracting Strategy (RES). This involves extracting one frame from every five in a continuous Quaternion sequence, then randomly selecting five segments of varying lengths to compile into a complete sequence (termed the Original sequence). We compare DSI against Polynomial Interpolation (PI) and Point-wise Polynomial Interpolation (PW-PI). The interpolation results from these three algorithms, alongside the Original sequence, are illustrated in Figure 5.

4.3 Evaluation on In-the-Wild Video

In this part, we extend the evaluation of our approach to in-the-wild videos. In-the-wild videos are captured in uncontrolled and natural environments, presenting a greater challenge for processing and analysis. We employ the same evaluation strategy as described in [46]. Initially, we utilize the ResNeXt-101 [14] 3D CNN model, pre-trained on the Mini-Kinetics-200 [49] dataset, as our feature extractor. Subsequently, we employ the Kinetics-15, a 15-class subset of the Kinetics-400 dataset, devised by [46]. From these 15 actions, one training video per class is randomly selected for synthetic data generation, with the remaining 725 videos designated for validation purposes. Additionally, random selection and nearest neighbor approaches, leveraging pre-trained features, serve as baselines for comparison. The results provided by [46] are included for comparison. The outcomes of various methods are detailed in Table 5.

5 Conclusion

In this paper, we present a comprehensive study on enhancing action recognition models using synthetic data augmentation, particularly focusing on the challenges posed by discontinuous frames and the limitations of existing methods. We propose 4A pipeline, leveraging game engine technology to generate sophisticated semantic representations of human motion. Our comparative analyses across various benchmarks demonstrate the superior performance of 4A pipeline in maintaining high accuracy levels even when trained on significantly reduced data. The components (Q-GCN and DSI) of our pipeline are pivotal in achieving these results, effectively capturing the nuanced dynamics of human motion and facilitating the generation of

Model Info	H36M-Original			H36M-Single			H36M-Extracted			H36M-SingleExtracted			H36M-4A (Ours)		
Method	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**	**Top-1**	**Top-5**	**Mean**
Random	-	-	6.7	-	-	6.7	-	-	6.7	-	-	6.7	-	-	6.7
VideoMAE [44]	38.3	74.3	35.0	26.1	84.3	24.5	15.9	50.9	17.4	11.2	46.7	11.7	56.1	89.7	58.2
TANet [25]	30.9	81.6	35.6	24.7	70.7	18.8	11.7	52.7	13.6	14.9	45.2	14.4	51.7	75.0	51.9
TPN [51]	29.2	80.1	33.6	16.6	65.4	18.4	10.6	56.7	11.6	10.2	51.4	10.4	65.6	88.1	58.9
X3D [10]	30.5	66.9	34.1	25.3	69.2	26.0	13.4	51.6	10.4	7.5	54.5	11.3	48.6	80.3	40.0
I3D [5]	34.2	66.4	28.9	28.4	80.7	21.0	10.1	44.4	13.2	12.1	50.8	11.6	48.8	74.0	42.4
I3D NL [48]	26.4	72.2	29.3	19.2	68.0	20.9	10.8	51.1	11.7	9.7	62.4	9.7	36.6	77.2	39.5

Table 3. Benchmark results on **H36M-Segment-Test**. Apart from the similar improvement observed in testing on H36M-Original-Test, our method enables H36M-4A to attain a better results on segmented video data. This enhancement could be attributed to the automatic segmentation inherent in the Dynamic skeletal interpolation process.

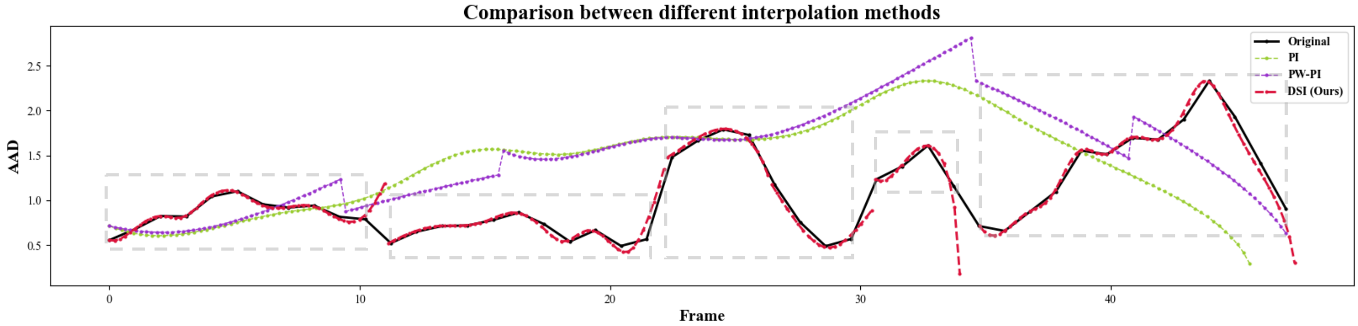

Figure 5. Comparative analysis of different interpolation method. In the provided figure, the plots represented in black, green, purple, and red correspond to the Absolute Angular Distance (AAD) of the original sequence (Original), and sequences interpolated using Polynomial Interpolation (PI), Point-wise Polynomial Interpolation (PW-PI), and Dynamic Skeletal Interpolation (DSI), respectively. The five gray dot boxes illustrate the five randomly selected Quaternion sequence segments. The PI method yields a smooth sequence but lacks dynamic segmentation capabilities. PW-PI produces a segmented sequence but leads to a decrease in movement amplitude during interpolation, evident from the overly smoothed curve. DSI stands out by not only accurately segmenting the sequence but also preserving the semantic integrity of the motion, showcasing its superior capability in maintaining both fluidity and semantic richness in the interpolated sequence.

Method	Whole-body	Major-part	Upper-body	Lower-body	Hands
SMPL-X [26]	123	72	89	64	167
Jointformer [27]	77	66	72	49	103
GLA-GCN [52]	79	54	63	41	91
Q-GCN (ours)	67	32	41	27	83

Table 4. Comparative results of different methods 2D to 3D orientation lifting task. Results are presented in terms of the mean Average Angular Distance (mAAD) loss in table, scaled by 10^3, with a lower score indicating superior performance. Our method Q-GCN, outperforms all other methods, demonstrating the highest effectiveness in this task.

	Accuracy (%)		
Method	RGB	Flow	RGB+Flow
Random	6.7	6.7	6.7
Nearest neighbor	8.6	13.1	13.9
Real	26.2	20.6	28.4
SURREACT	9.4	10.3	11.6
Synth + Real	32.7	22.3	34.6
4A	11.1	10.8	12.4
4A + Real	36.5	24.2	37.1

Table 5. Evaluation on In-the-Wild Videos. It reveals that training solely with synthetic data, regardless of the approach used, results in significantly lower performance compared to training with real data. Nonetheless, our method achieves a slight improvement when combined with real training data, achieving higher accuracy than training exclusively with real data or using the SURREACT.

semantically rich synthetic datasets. Evaluation on in-the-wild videos further validates the effectiveness of our approach, illustrating a enhanced ability of understanding actions in the real world.

References

[1] S. Ardianto and H.-M. Hang. Nctu-gtav360: A 360° action recognition video dataset. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2019.

[2] Autodesk. 3ds max 2023. https://www.autodesk.co.jp/products/3ds-max.

[3] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *2012 IEEE Computer society conference on computer vision and pattern recognition workshops*, pages 7–12. IEEE, 2012.

[4] Z. Cao, H. Gao, K. Mangalam, Q. Cai, M. Vo, and J. Malik. Long-term human motion prediction with scene context. *CoRR*, abs/2007.03672, 2020. URL https://arxiv.org/abs/2007.03672.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017. URL http://arxiv.org/abs/1705.07750.

[6] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 479–488. IEEE, 2016.

[7] C. R. de Souza12, A. Gaidon, Y. Cabon, and A. M. López. Procedural generation of videos to train deep action recognition networks. 2017.

[8] A. Diba, M. Fayyaz, V. Sharma, M. Paluri, J. Gall, R. Stiefelhagen, and L. V. Gool. Holistic large scale video understanding. *CoRR*, abs/1904.11451, 2019. URL http://arxiv.org/abs/1904.11451.

[9] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 2018.

[10] C. Feichtenhofer. X3d: Expanding architectures for efficient video recognition, 2020.

[11] Fivem. Fivem. https://fivem.net. Accessed March 8, 2023.

[12] J. H. Friedman. *A variable span smoother*. Laboratory for Computational Statistics, Department of Statistics, Stanford . . . , 1984.

[13] M. F. Ghezelghieh, R. Kasturi, and S. Sarkar. Learning camera viewpoint using cnn to improve 3d body pose estimation. In *3DV*, pages 685–693. IEEE, 2016.

[14] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace

the history of 2d cnns and imagenet? *CoRR*, abs/1711.09577, 2017. URL http://arxiv.org/abs/1711.09577.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[16] W. Hu, C. Zhang, F. Zhan, L. Zhang, and T. Wong. Conditional directed graph convolution for 3d human pose estimation. *CoRR*, abs/2107.07797, 2021. URL https://arxiv.org/abs/2107.07797.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[18] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo. Whole-body human pose estimation in the wild. In *Proceedings of the ECCV*, 2020.

[19] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik. Learning 3d human dynamics from video. *CoRR*, abs/1812.01601, 2018.

[20] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[21] M. Korban and X. Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *ECCV*, pages 761–776. Springer, 2020.

[22] Z. Li, X. Song, S. Chen, and K. Demachi. Data, language and graph-based reasoning methods for identification of human malicious behaviors in nuclear security. *Expert Systems with Applications*, 236:121367, 2024. ISSN 0957-4174.

[23] J. Liu and A. Mian. Learning human pose models from synthesized data for robust RGB-D action recognition. *CoRR*, abs/1707.00823, 2017. URL http://arxiv.org/abs/1707.00823.

[24] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *CVPR*, pages 5064–5073, 2020.

[25] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu. Tam: Temporal adaptive module for video recognition. *arXiv preprint arXiv:2005.06803*, 2020.

[26] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[27] S. Lutz, R. Blythman, K. Ghostal, M. Matthew, C. Simms, and A. Smolic. Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation. *ICPR*, 2022.

[28] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, pages 1–8, 2007. doi: 10.1109/CVPR.2007.383131.

[29] X. Ma, J. Su, C. Wang, H. Ci, and Y. Wang. Context modeling in 3d human pose estimation: A unified perspective. In *CVPR*, pages 6238–6247, 2021.

[30] A. A. A. Osman, T. Bolkart, and M. J. Black. STAR: A sparse trained articulated human body regressor. In *ECCV*, pages 598–613, 2020. URL https://star.is.tue.mpg.de.

[31] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *CoRR*, abs/1811.11742, 2018.

[32] D. Pavllo, D. Grangier, and M. Auli. Quaternet: A quaternion-based recurrent model for human motion. *CoRR*, abs/1805.06485, 2018.

[33] H. H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin. Video-based human action recognition using deep learning: A review, 2022.

[34] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the ECCV*, pages 650–667, 2018.

[35] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *CVPR*, pages 2458–2466, 2015. doi: 10.1109/CVPR.2015.7298860.

[36] A. Roitberg, D. Schneider, A. Djamal, C. Seibold, S. Reiß, and R. Stiefelhagen. Let's play for action: Recognizing activities of daily living by learning from life simulation video games. *CoRR*, abs/2107.05617, 2021. URL https://arxiv.org/abs/2107.05617.

[37] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[38] J. Seo, S. Han, S. Lee, and H. Kim. Computer vision techniques for construction safety and health monitoring. *Advanced Engineering Informatics*, 29(2):239–251, 2015. ISSN 1474-0346.

[39] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[40] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019.

[41] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011.

[42] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: viewpoint estimation in images using cnns trained with rendered 3d model views. *CoRR*, abs/1505.05641, 2015. URL http://arxiv.org/abs/1505.05641.

[43] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR (CVPR)*, June 2019.

[44] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

[45] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. *CoRR*, abs/1701.01370, 2017. URL http://arxiv.org/abs/1701.01370.

[46] G. Varol, I. Laptev, C. Schmid, and A. Zisserman. Synthetic humans for action recognition from unseen viewpoints. In *IJCV*, 2021.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.

[48] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *CVPR*, 2018.

[49] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017. URL http://arxiv.org/abs/1712.04851.

[50] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[51] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou. Temporal pyramid network for action recognition. In *CVPR (CVPR)*, 2020.

[52] B. X. Yu, Z. Zhang, Y. Liu, S.-h. Zhong, Y. Liu, and C. W. Chen. Glagcn: Global-local adaptive graph convolutional network for 3d human. *arXiv preprint arXiv:2307.05853*, 2023.

[53] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pages 3425–3435, 2019.

[54] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *ICCV*, pages 2344–2353, 2019.

[55] Y. Zhu, N. Samet, and D. Picard. H3wb: Human3.6m 3d wholebody dataset and benchmark. In *ICCV*, pages 20166–20177, October 2023.