

HAA4D: Few-Shot Human Atomic Action Recognition via 3D Spatio-Temporal Skeletal Alignment

Mu-Ruei Tseng¹, Abhishek Gupta¹, Chi-Keung Tang¹, Yu-Wing Tai²

¹The Hong Kong University of Science and Technology, ²Kuaishou Technology

Abstract

Human actions involve complex pose variations and their 2D projections can be highly ambiguous. Thus 3D spatio-temporal or 4D (i.e., 3D+T) human skeletons, which are photometric and viewpoint invariant, are an excellent alternative to 2D+T skeletons/pixels to improve action recognition accuracy. This paper proposes a new 4D dataset **HAA4D** which consists of more than 3,300 RGB videos in 300 human atomic action classes. **HAA4D** is clean, diverse, class-balanced where each class is viewpoint-balanced with the use of 4D skeletons, in which as few as one 4D skeleton per class is sufficient for training a deep recognition model. Further, the choice of atomic actions makes annotation even easier, because each video clip lasts for only a few seconds. All training and testing 3D skeletons in **HAA4D** are globally aligned, using a deep alignment model to the same global space, making each skeleton face the negative z-direction. Such alignment makes matching skeletons more stable by reducing intra-class variations and thus with fewer training samples per class needed for action recognition. Given the high diversity and skeletal alignment in **HAA4D**, we construct the first baseline few-shot 4D human atomic action recognition network without bells and whistles, which produces comparable or higher performance than relevant state-of-the-art techniques relying on embedded space encoding without explicit skeletal alignment, using the same small number of training samples of unseen classes.

1. Introduction

The goal of human action recognition in computer vision is to classify the person’s action performed in a given video. Various approaches have been studied, such as RGB-based approaches [3, 6] which utilize color images and recently use CNN-based or LSTM-based methods for classification, or 2D+T approaches [13] that encode 2D skeletons into some latent space for traditional supervised learning. The use of 3D skeletons provide outstanding results compared to other methods, as 3D skeletons are photometric,

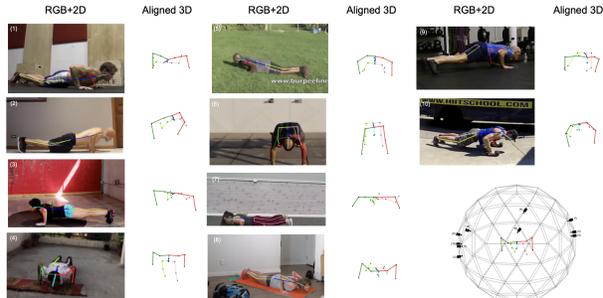


Figure 1. **2D vs 3D skeleton.** 3D skeleton is viewpoint invariant: all train/test 3D skeletons in **HAA4D** are registered by their respective orthographic camera pose estimated using our model. Thus a single training 3D skeleton per class is sufficient, in stark contrast to the use of 2D RGB images or skeletons, where a large number of training data is required and uneven sampling is an issue. Trained with as few as one 3D+T or 4D burpee skeleton, the network can readily recognize any normalized burpee action shot from any viewing angles, including those under the person which are difficult to capture.

geometric and viewpoint invariant, so that as few as one single 4D skeleton is sufficient to learn the pertinent action class (Figure 1). These invariances enable better temporal correlation by tracking the position of each 3D joint over time where each bone length is largely preserved throughout the action, in contrast to their projected 2D counterparts on images which may be distorted. Such invariances also make it easier for a computer algorithm to discern even fine differences in human poses.

Given the above advantages of 3D skeletons, previous human action dataset contributions [16, 25, 30, 31, 32] have provided valuable skeletal information for action recognition. However, these datasets suffer inherent disadvantages, such as lacking poses diversity or relatively low ground-truth accuracy. **HAA4D** is dedicated in complementing current human skeleton action datasets by providing a more comprehensive range of atomic actions in real-life scenarios, which have demonstrated excellent performance in recognizing composite and complex actions [9], while are easier to accurately annotate as each atomic action lasts for no more than 1–3 seconds. The samples in our dataset are collected from videos in-the-wild; therefore, we employ a deep

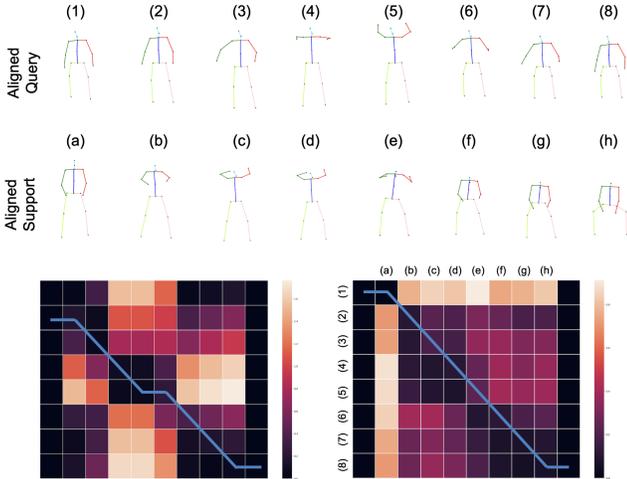


Figure 2. **Implicit alignment** (left) vs **explicit alignment** (right). Alignment matrix of the *battle-rope-power-slam* action between the query and support videos. The embeddings are fed to an OTAM module to generate a distance map of size $T \times T$ using the cosine distance between the query and supports. Darker cells indicate higher similarity, while the blue line shows the matching path.

alignment model to predict the camera pose in each video frame, followed by aligning all 3D+T skeletons in an uniform camera coordinate system (Figure 1) such that all faces are facing in the negative z -direction. As human actions in the same class often share similar trajectories in their skeletal movements, such global alignment which transforms all train/test examples to the same coordinate system can significantly improve matching the pertinent action sequences using explicit geometry features.

With the emergence of deep learning and its superior performance in image classification tasks, related works mainly focus on supervised learning techniques that directly feed 4D human skeleton data into the network. Their networks are expected to generalize human actions and learn the underlying relationships in the embedded space. However, it is very difficult for deep learning to obtain satisfactory results with excessive learnable parameters when only a few examples are presented, given that 4D human skeletons are expensive to acquire. Furthermore, if the network is to recognize as many different human actions as possible, the size of the training dataset for training will be prohibitively large.

Thus, our recognition model in this paper adopts *few-shot* learning techniques [2, 4, 12, 23, 28], making it the *first* on few-shot human atomic action recognition. Our few-shot model, combined with a *global alignment* model, performs matching using explicit 4D skeletal representation. Exploiting geometry inherent in 4D human skeletons avoids training complex models to learn proper encoding in the embedded space, which may still not produce better alignment than using explicit alignment (Figure 2) and thus hamper-

ing recognition. Our experiments validate that the proposed few-shot model working in tandem with a global alignment model produces better recognition results with fewer training examples for each action. In summary, our contributions consist of:

1. HAA4D, a human atomic action dataset where all 4D skeletons are globally aligned. HAA4D complements existing prominent 3D+T human action datasets such as NTU-RGB+D [25, 31], and Kinetics-skeleton [19]; HAA4D contains 3390 samples of human actions in the wild with 300 different kinds of activities. The samples for each human action range from 2 to 20, each provided with RGB frames and their corresponding 3D skeletons;
2. introducing an alignment network for predicting orthographic camera poses in the train/test samples, where all 4D skeletons are aligned in the same camera space each facing the negative z -direction. This allows for better recognition results with small number of training samples compared to ST-GCN[40], Shift-GCN[8], and SGN[43];
3. the first few-shot baseline for 4D human (atomic) action recognition that produces results comparable to or better than state-of-the-art techniques on unseen classes using a small number of training samples.

2. Related Works

2.1. Human Video Action Datasets

In skeletal human action recognition, several datasets are often used as benchmarks, see Table 1. NTU-RGB+D [31] and NTU-RGB+D 120 [25] were collected under laboratory settings that cover actions in three main categories: daily actions, mutual actions, and medical conditions. These datasets comprise of RGB videos, depth map sequences, 3D skeletal data, and infrared videos for each sample. There are other RGB+D datasets that were also captured in laboratory settings, such as UWA3D Multiview II [30] which contains 30 various activities, and SYSU 3D HOI [16] which provides 12 activities with 40 different participants focusing more on human-object interaction. Recently, Ego4D [14] has been released which contains massive-scale egocentric videos and provides long sequences of actions in real-life scenarios, where parts are given with the corresponding 3D meshes. However, although having a massive number of samples per action, these datasets are often not sufficiently diversified, subject to the drawback of not covering a wide range of different actions.

Kinetics-skeleton, on the other hand, contains more diversified human actions. This skeleton dataset was built on top of Kinetics [19], a video action datasets collected from YouTube with more than 300,000 video clips covering 400

Datasets	Samples	Actions	Views	Modalities	Year
UWA3D Multiview II [30]	1075	30	5	RGB+D+3DJoins	2015
NTU RGB+D [31]	56,880	60	80	RGB+D+IR+3DJoins	2016
SYSU 3DHOI [16]	480	12	1	RGB+D+3DJoins	2017
Kinetics 400 [19]	306,245	400	-	RGB	2017
NTU RGB+D 120 [25]	114,480	120	155	RGB+D+IR+3DJoins	2019
Kinetics-700-2020 [32]	633,728	700	-	RGB	2020
Ego4D [14]	3,025	110	-	RGB+3DMeshes+Audio	2021
HAA4D	3,390	300	2/20	RGB+3DJoins	2021

Table 1. Comparison between HAA4D and other public datasets for human action recognition. With fewer samples than many existing datasets, HAA4D provides class-balanced and diversified actions with 3D+T or 4D skeleton data useful for few-shot action learning.

action classes. OpenPose [7] was used to extract 2D human skeletons. Together with EvoSkeleton [24] used to lift the 2D key-points to 3D skeletons, we can extract the 3D skeletal data from the images. While Kinetics-skeleton introduces more variety of human poses to the human skeletal action recognition domain, the correctness of the 3D skeletons generated may be questionable, as the 2D joints prediction network can fail if parts of the human body are outside of the image frame.

To ameliorate the problem of existing datasets problem, namely, lack of pose diversity and low ground-truth accuracy, we introduce HAA4D which contains 300 human action classes, each with 20 examples with accurately annotated 2D joints position.

2.2. Video-Based and Few-Shot Action Recognition

For more recent examples on video-based action recognition, C3D [33] performs 3-dimensional convolution on the input image sequence to extract spatio-temporal features. In [34] an architecture is proposed for first processing each image using 2D convolution, and then using a bidirectional LSTM network to learn temporal information. TSN [37] divides a video into several segments and selects snippets to pass to a spatial stream ConvNets and a temporal stream ConvNets. Video-level prediction is then derived from the consensus of the snippets. However, all these models contain an excessive number of learnable parameters that require training on large-scale datasets, which can fail when training samples are few and expensive to obtain. Thus, more works have focused on few-shot video action classification.

In [29] the authors proposed to learn to generate class embeddings using a Word2Vec [27] model. NGM [15] introduces a graph matching metric on a graphical representation of a video. Dense dilated network [39] uses a dilated CNN network on videos. In contrast to these works that neglect the temporal dimension of videos, TARN [4] introduces temporal attention for temporal alignment of videos. A similar approach is used in [41] by adopting temporal attention as a temporal form of self-supervision. Our work is closest in spirit to OTAM [5] which uses an ordered tempo-

ral alignment module, inspired from dynamic time warping (DTW) [1]. However, instead of focusing only on temporal alignment, HAA4D explicitly aligns along 3D+T or 4D dimensions since similar actions, after calibration removing the view and scale variation, share similar joint distribution and moving sequences.

However, all of the aforementioned approaches directly train a video classification model using RGB-frames, which poses a deep neural network a great challenge in relating video frames as they can vary widely in background and illumination. In this paper, we use skeleton data extracted from the videos for better spatio-temporal alignment.

2.3. Skeleton-Based Action Recognition

Given the aforementioned advantages, skeleton-based action recognition has been a popular topic in computer vision that aims to classify actions using skeleton joint information. Conventional methods rely on hand-crafted features to capture dynamic movement of human joints [17, 35, 36]. With emergence of deep-learning, earlier related works represent joint coordinates as a vector, and use recurrent neural network for action classification [10, 26, 42], or represent skeleton sequence as a pseudo-image as input to a CNN-based network [20, 21, 22]. Recently, ST-GCN in [40], which consists of spatial graph convolution layers and temporal convolution layers, uses graph representation of skeleton sequence. ST-GCN has been extended in [8] by incorporating shift convolution [18, 38, 44] on a graph CNN, which achieves better performance with a lighter network. In this paper, we use Shift-GCN [8] as a backbone for our baseline model to compare the efficacy of our globally aligned 4D skeletons.

3. HAA4D

This section will introduce details and structure of our HAA4D dataset, its expandability, and the evaluation criteria. Figure 3 and 4 shows some examples of our HAA4D dataset. Full details of HAA4D are available in the supplementary materials.

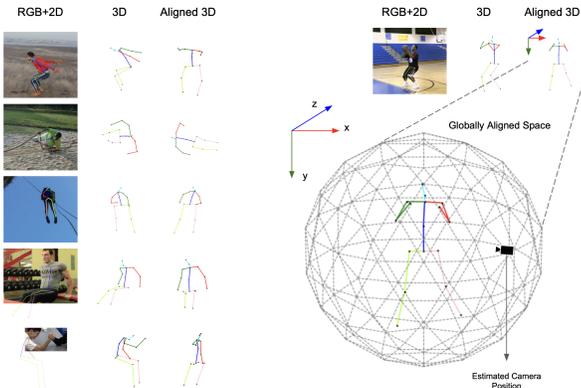


Figure 3. Samples from the HAA4D dataset. First column shows the RGB images and their 2D annotations, second and third columns represent the 3D skeletons in the original and globally aligned space (with face at negative z -direction). The right shows the predicted camera position at one of the subdivided icosahedron vertex in the globally aligned space.

Classes	Total Samples	Globally Aligned Samples
300	3390	400
Total Frames	Min/Max Frames	Average Frames
212042	7 / 757	63
2 Examples	20 Examples	Person Per Example
145 classes	155 classes	1

Table 2. Summary of HAA4D.

3.1. Dataset Structure

HAA4D is a challenging human action recognition 3D+T dataset that is built on top of the HAA500 [9] dataset. HAA500 is a curated video dataset consisting of atomic human action video clips, which provides diversified poses with variation and examples closer to real-life activities. Refer to [9] for details on vocabulary selection and video clip selection, and advantages including clean labels for each frame, human-centric atomic-action frames, and high scalability. Table 2 tabulates a summary of HAA4D.

Similar to Kinetics [19], the original dataset provides solely in-the-wild RGB videos. However, instead of using existing 2D joints prediction networks, which often produce inaccurate results when joints are hidden or not present, HAA4D consists of hand-labeled 2D human skeletons position. These accurate features can not only increase the precision of the 3D ground-truth skeleton but also benefit training for new joints prediction models that can reasonably hallucinate out-of-frame joints. The labeled 2D joints will then be raised to 3D using EvoSkeleton [24]. HAA4D is thus named with its accurate per-frame 3D spatial and temporal skeletons for human atomic actions.

HAA4D consists of 3,300 real-life RGB videos of specific human action classes with the corresponding 2D joints and 3D human skeletons. The skeleton topology follows the one in Evoskeleton [24] which consists of 17 joints which are quite sufficient in representing a wide range of

human body and part movements. The dataset has 300 action classes that are divided into two categories: primary classes and additional classes. Primary classes have 20 samples per class, while additional classes contain two examples per class. Recall the main advantage of 4D skeleton data is its viewpoint invariance with a few as one 3D skeleton per frame, given they are globally aligned with camera poses available which are estimated using our camera prediction model.

For primary classes, videos 0–9 are used for training, and the remaining are for evaluation. Actions in the additional classes are dedicated to one-shot learning that evaluates the performance of the proposed model to differentiate the action when very limited data are presented. For example, this paper trains the alignment model on the first ten examples of human movements in the primary classes with data augmentation techniques to be detailed. The alignment model is then tested with videos in the evaluation sets of the primary classes for cross-view evaluation, and with examples in the additional classes for cross-action assessment.

3.2. Mean Average Precision (mAP)

We calculated the mean average precision(mAP) of the whole dataset and each primary class using the bounding boxes of the predicted skeletons generated from AlphaPose [11] and our human-labeled ground truth skeletons to calculate the IoU. The threshold of the IoU is set to 0.5. The overall mAP of the HAA4D dataset is 63.71. However, we observed a significant variance between the mAP of each class. For example, the mAP of the action ‘abseiling’ is 92.08 while the mAP of ‘applying_cream’ is 1.28. One reason that caused the difference is that most images captured contain only the upper half of the body for activities like ‘applying_cream’. This shows the insufficiency of state-of-the-art pose prediction networks like AlphaPose in predicting joints that are out of boundary; therefore, making our perfectly labeled ground truth more valuable. Although bounding boxes cannot effectively represent each joint’s position accuracy, we will provide part-level bounding annotation for more accurate evaluation. The mAP details for each action class will be included in the supplemental material.

3.3. Annotation Tool

To help expand the dataset to more actions and classes, we have developed a simple interactive annotation tool that supports faster user labeling (Figure 5). Similar to Kinetics-skeleton use of OpenPose [7] to predict positions of 2D joints, the user can load preliminary predictions from AlphaPose [11] and can correct intermediate frames where the network prediction is not precise. Observing that human movement when viewed in a short period is relatively linear, the annotation tool supports linear interpolation so that the user can avoid frame-by-frame processing whenever possible. The predicted 3D skeleton is also shown alongside in



Figure 4. Examples of the 8-frame sampling of the HAA4D dataset.

the interface for the user to monitor the generated ground truth in real time. The annotation tool covers the end-to-end generation of 3D human skeletons from images, which is user-friendly and efficient in annotation, thus making this HAA4D dataset easily expandable. We hope that by providing this annotation tool, the dataset can become more comprehensive in the future to cover more human action poses, contributing to research in few-shot human pose estimation and action recognition that utilizes 3D+T data.

3.4. Globally Aligned Skeletons

Since examples in HAA500 were shot in the wild, each action sample has different viewpoints and scales, making

the predicted 3D skeletons in various coordinate systems. Therefore, in HAA4D, we apply the following to standardize the 3D generated skeletons.

First, all skeletons are centered such that their lower-spine joint is placed at the origin. Second, because EvoSkeleton [24] generates 3D skeletons frame-by-frame, it does not guarantee equal bone lengths throughout the video. Hence, to ensure the human skeleton’s uniformity in the action video, we use bone lengths in the first frame as the reference to replace the lengths of the rest frames. We also scale all the bones such that their l2-norm equals to one, i.e., $b_i^f = b_i^1$, where $\sqrt{\sum_{b_i^1 \in B} (b_i^1)^2} = 1$. b_i^f represents bone i in the f -th skeleton frame and B is the set of bones in the skeleton topology.

Then, we provide globally aligned skeletons (to parts of the dataset) by manually rotating them so that the human faces the negative z -direction at the start of each action. We choose the action to face in the negative z -direction because most people when taking photos will look into the camera. We calculate the alignment using only the first frame and apply the rotation to the rest of the sequence as a frame-by-frame alignment will break the relative movement of the original action. The calibration makes our data view-invariant by bringing all skeletons to the same space. We will argue in the next section that eliminating the rotation and scale factors help us better compare the skeleton movement in different actions by directly using the adjusted 3D joints position. The manually corrected, globally aligned skeletons are also used to train the alignment model, which takes any actions, regardless of their view angles and action type, and transforms them to the global coordinate system.

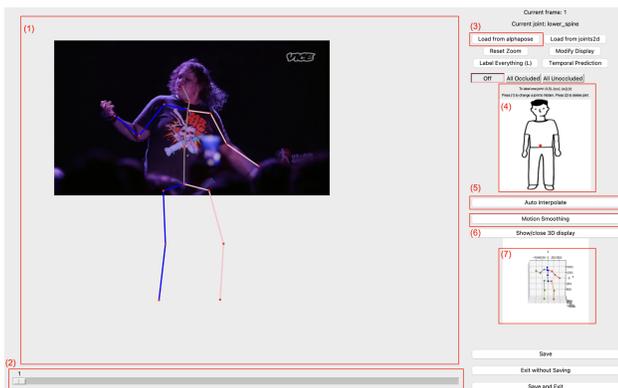


Figure 5. Main functionalities of our HAA4D annotation tool. (1): Annotating Canvas. (2): Dragging bar to skim through all the frames. (3): Load preliminary prediction from AlphaPose. (4): Map of the current label joint. (5): Linear interpolation. (6): Gaussian smooth the trajectories of all the joints. (7): Predicted 3D position from EvoSkeleton.

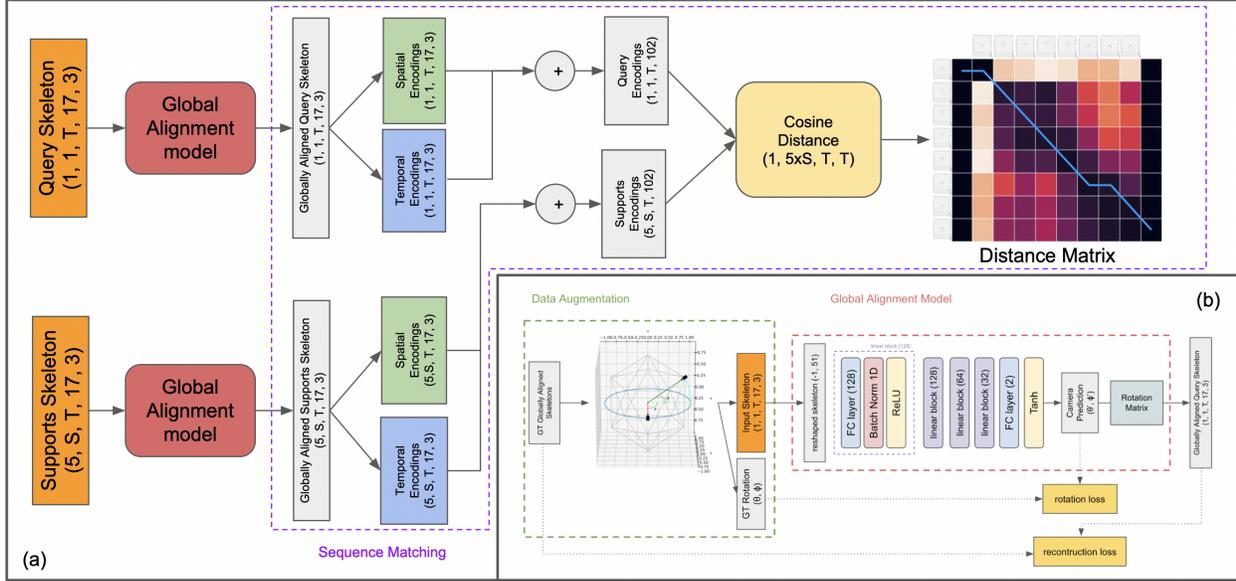


Figure 6. (a) Overview of the globally aligned few-shot action recognition architecture. (b) Architecture of the Global Alignment Model together with data augmentation.

4. Few-Shot Action Recognition Model

In this section, we describe our few-shot skeleton-based action recognition model. Figure 6 shows the overall architecture which consists of two modules: the global alignment module (with data augmentation) and the sequence matching module.

Following [5, 37], given a video example, we divide it into t_n segments and randomly select a frame within each segment. We randomly selected one frame from each segment to introduce more randomness to each input sequence. As we work on atomic actions, given the same action type, the start and end will match. However, each person may act at a different pace, and thus, we do not hardcode any frame to represent each segment.

We then concatenate the t_n skeletons to represent the video, denoted by S . This allows each video to be represented by the same number of frames. Similarly, the support sets are constructed by randomly selecting s_n samples from each w_n class, including the query's. The global alignment model transforms both the query and support into the globally aligned space before action recognition, by comparing the sequence matching results with the support sets.

4.1. Global Alignment Model (GAM)

As query and support videos can be captured in-the-wild, we propose the global alignment model to bring all inputs to the same space, such that actions face in the negative z -direction at the beginning. The camera angle will also be available after our global alignment.

To prepare or synthesize training data, since we have the ground truth globally aligned skeletons as described in section 3.4, we perform data augmentation by uniformly

sampling different camera views from vertices on an n -frequency icosaheron (i.e., subdivided icosahedron incident on a unit sphere as shown in Figure 1). Our globally aligned skeletons are viewed from the default orthographic camera C_D located at $(0, 0, -1)$. By simplifying the camera to have no self-rotation, we only need to predict two rotation parameters, namely, azimuth and altitude angles, denoted by β (θ and ϕ) with respect to the default camera position $(0, 0, -1)$. We obtain θ and ϕ by:

$$\theta = \pi - \arctan\left(\frac{C_x}{C_z}\right), \quad \theta = \begin{cases} \theta - 2\pi, & \text{if } \theta > \pi \\ \theta, & \text{otherwise} \end{cases} \quad (1)$$

$$\phi = \arctan\left(\frac{C_y}{\sqrt{(C_x)^2 + (C_z)^2}}\right) \quad (2)$$

We then use Rodrigues rotation to obtain the rotation matrix $R = R_2 \cdot R_1$, where:

$$\text{Rodrigues}(\hat{n}, \alpha) = I_3 + \sin \alpha K + (1 - \cos \alpha) K^2, \quad (3)$$

where $K = \text{Skew}(\hat{n})$

We first rotate in θ using:

$$R_1 = \text{Rodrigues}(\hat{n}_1, \theta), \quad \hat{n}_1 = \langle 0 \quad -1 \quad 0 \rangle \quad (4)$$

Then rotate in ϕ : let $\vec{v}_1 = \langle 0 \quad 1 \quad 0 \rangle$, $\vec{v}_2 = R_1 \cdot C_D$, then

$$R_2 = \text{Rodrigues}(\hat{n}_2, |\phi|), \text{ where } \hat{n}_2 = \begin{cases} \vec{v}_2 \times \vec{v}_1, & \text{if } \phi > 0 \\ \vec{v}_1 \times \vec{v}_2, & \text{otherwise} \end{cases} \quad (5)$$

We obtain the augmented view skeleton S_A by rotating the globally aligned skeletons S_G :

$$S_A = R^{-1} \cdot S_G, \text{ where } R = R_2 \cdot R_1 \quad (6)$$

We sample data from 92 camera angles on a 3-frequency subdivision icosahedron, trained on 73 views, and test on the rest 19 views. We employ rotation loss and reconstruction loss defined as follows:

$$\mathcal{L}_{rot} = \text{MSE}(\cos \beta_{pr}, \cos \beta_{gt}) + \text{MSE}(\sin \beta_{pr}, \sin \beta_{gt}) \quad (7)$$

$$\mathcal{L}_{rec} = \text{RMSE}(S_{pr}, S_A), \text{ where } S_{pr} = (R(\beta_{pr}))^{-1} \cdot S_G \quad (8)$$

The final loss for our global alignment model is

$$\mathcal{L} = w_1 \mathcal{L}_{rot} + w_2 \mathcal{L}_{rec}, \quad (9)$$

where we set w_1 to 10 and w_2 to 1.

4.2. Sequence Matching

After transforming all input skeletons to the same global space, where we denote each transformed skeleton as S , we encode the 3D positions of the joints $f_{pos}(S)$ and their respective trajectory $f_{tra}(S)$ as embeddings for sequence matching. The encoding functions are defined as follows:

$$f_{pos}(S) = S^1 \oplus S^2 \oplus \dots \oplus S^{t_n} \quad (10)$$

$$f_{tra}(S) = \mathbf{1} \oplus (S^2 - S^1) \oplus \dots \oplus (S^{t_n} - S^{t_n-1}) \quad (11)$$

$$f_{skel}(S) = f_{pos}(S) \oplus f_{tra}(S) \quad (12)$$

where \oplus denotes concatenation between the two frames. Inspired by OTAM [5] and dynamic time warping [1], we use the matching score between the explicit query $f_{skel}(S_q)$ and supports $f_{skel}(S_s)$ in embedding to determine the action class of the given query. We generate the distance matrix $D = 1 - \text{cosine_similarity}(f_{skel}(S_q), f_{skel}(S_s))$. The score of the query and the support is represented by $-\sum_{d \in path} d$, where $path$ is a relaxed-minimum path of D defined in [5]. The final score of a query and a support class is the average of scores for each k support data of the class. The top final score corresponds to the action class of the query video.

5. Experiments

5.1. Skeletal Representation

Implicit vs Explicit Without representative skeletal-based few-shot action recognition models, we create a baseline model to evaluate the performance. The baseline model uses Shift-GCN [8] as the backbone to encode input skeletons in an embedded space and perform the temporal alignment. The Shift-GCN model is first pre-trained with the embedding cosine similarity loss and the skeleton reconstruction loss. To simulate the few sample situations in the

	NTU-RGB+D xview [31]		HAA4D	
	5way 1shot	5way 5shot	5way 1shot	5way 5shot
Baseline (I)				
- 2D	24.7	25.5	23.4	25.6
- 3D	46.8	51.7	47.0	50.3
Ours (E)				
- 2D	44.7	49.8	45.8	48.3
- 3D	57.8	70.0	52.1	62.3

Table 3. Performance under different skeletal representation. (I) stands for using implicit representation, and (E) means using explicit representation of human skeleton.

dataset, we randomly select ten samples from each NTU-RGB+D [31] and for training, and use the other ten examples for testing. The encoded skeleton in the embedded space is the implicit representation of the skeleton sequence used for sequence matching.

2D vs 3D Here we compare the effectiveness between the 2D and 3D skeletons. The most notable difference between using 2D and 3D skeletons is that for 2D skeletons, we cannot perform data augmentation as rotation cannot be done in the 2D space without breaking the original skeleton structure. We experiment on both the baseline model and other proposed architectures. In the baseline model, we follow the same approach to encode the 2D skeletons to an implicit representation in the embedded space. However, for testing on our model, we directly feed 2D skeletons for sequence matching without the proposed 3D global alignment.

Results Table 3 shows the effectiveness of using different representations of human skeleton sequences for action recognition. The baseline model utilizes implicit representation, while our model uses explicit information. We test both models with 2D and 3D as input skeletons and demonstrate that explicit 3D representation, i.e., globally aligned information, produces the best accuracies in action recognition.

5.2. Ablation Studies

In this section, we conduct ablation studies to examine the effect of individual part of our model. We tested on three different factors: the number of sampled frames, the sequence matching approach, and the explicit information used as encodings. We monitor the results on both HAA4D and NTU-RGB+D.

Sampling Rate For each human skeleton action sequence, we divide them into several segments and select one skeleton from each to represent the segment. This sampling technique allows all the action sequences to be in uniform length while increasing the difference between the neighboring frames. We test on different numbers of frames to represent the sequence, as oversampling can lead to redundancy and unnecessary computational load, while undersampling can lead to temporal aliasing. We padded the

	NTU-RGB+D xview [31]		HAA4D	
	5way 1shot	5way 5shot	5way 1shot	5way 5shot
Mean				
- $t_n = 8$	47.5	51.0	45.9	54.8
- $t_n = 32$	51.8	59.8	47.9	56.5
- $t_n = \text{all}$	52.0	60.5	47.5	53.2
Regular DTW				
- $t_n = 8$	58.0	69.8	52.4	62.3
- $t_n = 32$	59.3	71.3	53.5	62.7
- $t_n = \text{all}$	59.1	69.0	53.4	62.5
OTAM				
- $t_n = 8$	56.5	67.8	51.9	62.0
- $t_n = 32$	57.8	70.0	52.1	62.3
- $t_n = \text{all}$	-	-	-	-

Table 4. Performance of few-shot action classification under different sequence matching methods and sampling rates. We do not consider $t_n = \text{all}$ for OTAM as it constrains the distant matrix to be squared and having equal length paths.

last frame to extend the input sequence for input with fewer frames than the segment size.

Sequence Matching Our action recognition model determines the action by comparing the sequence matching score between the query set and support sets. We experiment the following matching techniques: Mean, Regular DTW, and OTAM [1, 5]. The Mean approach for sequence matching directly computes the average between the two distance matrix D and thus neglects the temporal ordering. The other two methods consider temporal ordering. Regular DTW assumes the two sequence matches at the start and end, which allows movement in the direction of \searrow , \rightarrow , and \downarrow when computing the aligned path in the distance matrix. OTAM on the other hand assumes the alignment can start and end from any time in the sequence, thus only allowing \searrow and \rightarrow to ensure that all the alignment paths have equal length.

Explicit Encodings In addition to directly using the aligned 3D coordinates (0^{th} order) of the skeleton sequence as our explicit embeddings, we evaluate the effect of using different combinations between the 0^{th} , 1^{st} , and 2^{nd} order representations. The 1^{st} order of the skeleton sequence consists of the joints movement throughout temporal frames. The 2^{nd} order representation includes the curvature of the joint path.

Results From our experimental results tabulated in Table 4, the 32-frame sampling outperforms others in both HAA4D and NTU-RGB+D datasets. In addition, we observe that techniques that consider temporal ordering, such as regular DTW and OTAM, tend to produce better accuracy in action prediction for the sequence matching methods. Notably, the actions in HAA4D are atomic, which are diverse including atomic motions that form more complex actions, where the actions in the same class match at the beginning as well as the end of the sequence. This feature benefits the regular DTW to obtain more reliable results.

	0^{th} only	$0^{th}+1^{st}$	$0^{th}+1^{st}+2^{nd}$
HAA4D			
- 5way 1shot	51.4	52.1	50.7
- 5way 5shot	56.8	62.3	59.6

Table 5. Performance of the accuracy between using different explicit encoding.

Table 5 shows that by using both the globally aligned position and the joints temporal movement, the model produces the best performance.

5.3. Comparison with State-of-the-Arts

This section evaluates state-of-the-art skeleton-based action recognition models on our HAA4D dataset, and compares the results of the testing on other public datasets. Although few-shot learning has been used on action recognition [5], few approaches have been conducted via the 3D or 3D+T skeleton-based method. In addition, one of our claims is the use of few-shot learning on 3D skeleton human action recognition to improve the prediction accuracy in sparse training data. We compare our result with the state-of-the-art supervised approaches to support our claim.

The skeletons of HAA4D are processed according to the description in section 3.4, which centers the origin and constrains all skeletons to have equal bone length throughout the video. The inputs of ST-GCN [40] and Shift-GCN [8] follow the shape (batch-size \times channels \times frames \times joints \times people). We let the channels be the (X, Y, Z) of coordinates of the 3D skeletons and pad zeros to the second person in people dimensions, as HAA4D targets only single-person action. SGN [43] splits a multi-person frame into multiple frames, in which each frame contains only a single person. This model also requires segmenting the skeleton sequence into 20 clips, where one frame is randomly sampled from each segment to form the new sequence. We follow the same data processing techniques to evaluate our HAA4D.

Table 6 shows that all models suffer from the lack of accuracy in Kinetics-skeleton. One possible reason is that the skeletons are directly extracted from the Kinetics using OpenPose [7] without correction done on the corrupted skeletons, where the 2D predictions are not sufficiently accurate to be lifted to the corresponding 3D skeleton. On the other hand, our HAA4D is a curated dataset with precise 2D joints location; even out-of-frame joints are manually completed reasonably. Thus the ground truth 3D skeleton is more reliable. This leads to better accuracy for the state-of-the-art models on our dataset. However, noting that HAA4D is designed for few-shot action recognition and due to the limited size of HAA4D, there is still a gap in prediction precision between HAA4D and NTU-RGB+D [31].

	ST-GCN [40]	Shift-GCN [8]	SGN [43]
NTU-RGB+D [31]			
- xview	88.8	95.1	94.5
- xsub	81.6	87.8	89.0
Kinetics-skeleton [19]	31.6	17.7	20.8
Ours (HAA4D)	21.2	53.0	53.3

Table 6. Performance of state-of-the-art skeleton-based action recognition models on different datasets.

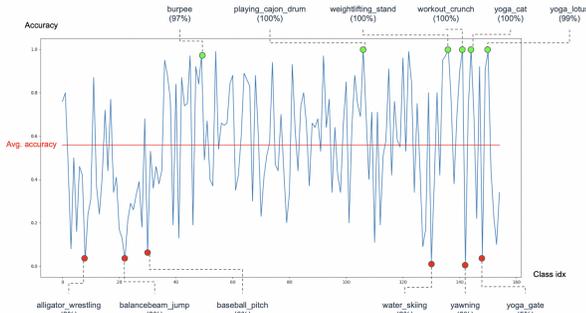


Figure 7. Accuracy respect to each action classes

6. Discussion

6.1. Substantial need for the clean HAA4D dataset

One may concerned that “clean” dataset is not practical in real-life scenarios. However, current 2D pose prediction networks are still far from the accuracy of hand-corrected labeling. With much effort, introducing this cleaner dataset can advance this area of studies as all images collected in HAA4D are “in the wild” with more accurate ground truth 2D skeletons. Moreover, we observed several actions in NTU RGB+D 120 [25] and Kinetics-skeleton [19] are corrupted, including wrong subjects or unreasonable joint movements, that humans cannot even recognize. Our highly diversified HAA4D does not have these problems to distract learning. Also, having a clean dataset, we can add and control noises to generate false actions and even train networks to identify these errors.

6.2. Evaluation on failure cases

Figure 7 shows the worst 5 classes for our action recognition model are yawning (0%), water_skiing (2%), alligator_wrestling (3%), balancebeam_jump (3%), yoga_gate (5%). Actions such as yawning mainly involve mouth movements, which is difficult to reflect using skeletons. The model is subject to cases that lack significant movements in parts of the skeleton. Similar cases can be found in water_skiing and alligator_wrestling. The model also obtains low accuracy when the actions contain significant variances and lack consistency. For actions like balancebeam_jump, different dancers have unique styles and thus poses. It is the balance beam (interacting object) that helps humans identify the action even though each example looks completely different.

7. Conclusion

This paper proposes a new 4D dataset **HAA4D** which consists of more than **3300** RGB videos in **300** human atomic action classes. HAA4D is clean, diverse, class-balanced where each class is viewpoint-balanced with the use of 3D+T or 4D skeletons. All training and testing 3D skeletons in HAA4D are globally aligned. Such alignment makes matching skeletons more stable, and thus with fewer training samples per class needed for action recognition leveraging full 3D+T information, in stark contrast to the 2D counterparts where massive amount is required to cover adequate viewpoints.

Given the high diversity and skeletal alignment in HAA4D, we construct the first baseline few-shot 4D human atomic action recognition network, which produces comparable or higher performance than relevant state-of-the-art techniques, using the same small number of training samples of unseen classes. Through ablation studies, we have verified the advantages of 3D skeletons over 2D counterparts. We have also found that existing 3D skeletal action datasets (e.g., NTU-RGB+D [31] and Kinetics-skeleton [19]) significantly benefit from the proposed explicit alignment in the few-shot setting. With its high scalability where only a small number of 3D+T skeletons to be annotated for class extension using our annotation tool, and 3D+T global alignment contributed by this paper, we hope HAA4D will spawn fruitful future works on skeletal human action recognition. Dataset, codes, models and tools will be released upon publication.

References

- [1] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. 3, 7, 8
- [2] Evgeniy Bart and Shimon Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *CVPR 2005*. 2
- [3] Amel Ben Mahjoub and Mohamed Atri. Human action recognition using rgb data. In *2016 11th International Design Test Symposium (IDT)*, pages 83–87, 2016. 1
- [4] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. In *BMVC 2019*. 2, 3
- [5] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *CVPR 2020*. 3, 6, 7, 8
- [6] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1
- [7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019. 3, 4, 8
- [8] Ke Cheng, Yifan Zhang, Xiangyu He, Weihang Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition

- with shift graph convolutional network. In *CVPR 2020*. 2, 3, 7, 8, 9
- [9] Jihoon Chung, Cheng hsin Wu, Hsuan ru Yang, Yu-Wing Tai, and Chi-Keung Tang. Haa500: Human-centric atomic action dataset with curated videos. In *ICCV 2021*. 1, 4
- [10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 3
- [11] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017. 4
- [12] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 2
- [13] Sumaira Ghazal, Umar Khan, Mubasher Saleem, Nasir Rashid, and Javaid Iqbal. Human activity recognition using 2d skeleton data and supervised machine learning. *IET Image Processing*, 13, 09 2019. 1
- [14] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrahm Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Mery Ramazanov, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2021. 2, 3
- [15] Michelle Guo, Edward Chou, De-An Huang, Shuran Song, Serena Yeung, and Li Fei-Fei. Neural graph matching networks for fewshot 3d action recognition. In *ECCV 2018*. 3
- [16] JF Hu, WS Zheng, J Lai, and J Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2186–2200, 2017. 1, 2, 3
- [17] Mohamed E Hussein, Marwan Toriki, Mohammad A Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Twenty-third international joint conference on artificial intelligence*, 2013. 3
- [18] Yunho Jeon and Junmo Kim. Constructing fast network through deconstruction of convolution. *arXiv preprint arXiv:1806.07370*, 2018. 3
- [19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 2, 3, 4, 9
- [20] QiuHong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017. 3
- [21] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1623–1631. IEEE, 2017. 3
- [22] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 597–600. IEEE, 2017. 3
- [23] Fei Fei Li, Rufin VanRullen, Christof Koch, and Pietro Perona. Rapid natural scene categorization in the near absence of attention. *Proceedings of the National Academy of Sciences*, 99(14):9596–9601, 2002. 2
- [24] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 4, 5
- [25] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2020. 1, 2, 3, 9
- [26] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, pages 816–833. Springer, 2016. 3
- [27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*. 3
- [28] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000. 2
- [29] Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, S Arulkumar, Piyush Rai, and Anurag Mittal. A generative approach to zero-shot and few-shot action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 372–380. IEEE, 2018. 3
- [30] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian. Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(12):2430–2443, dec 2016. 1, 2, 3

- [31] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 1, 2, 3, 7, 8, 9
- [32] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset, 2020. 1, 3
- [33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2015. 3
- [34] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access*, 6:1155–1166, 2018. 3
- [35] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR 2014*. 3
- [36] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR 2012*. 3
- [37] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition, 2016. 3, 6
- [38] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9127–9135, 2018. 3
- [39] Baohan Xu, Hao Ye, Yingbin Zheng, Heng Wang, Tianyu Luwang, and Yu-Gang Jiang. Dense dilated network for few shot action recognition. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 379–387, 2018. 3
- [40] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2, 3, 8, 9
- [41] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *ECCV 2020*. 3
- [42] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017. 3
- [43] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 8, 9
- [44] Huasong Zhong, Xianggen Liu, Yihui He, and Yuchun Ma. Shift-based primitives for efficient convolutional neural networks. *arXiv preprint arXiv:1809.08458*, 2018. 3

HAA4D: Few-Shot Human Atomic Action Recognition via 3D Spatio-Temporal Skeletal Alignment (Supplementary Material)

Mu-Ruei Tseng¹, Abhishek Gupta¹, Chi-Keung Tang¹, Yu-Wing Tai²

¹The Hong Kong University of Science and Technology, ²Kuaishou Technology

1. Action Classes

This section provides the complete list of actions contained in our HAA4D dataset. The dataset can be characterized in two parts: **primary classes** and **additional classes**, where the former contains 155 actions and the latter contains 145 actions.

1.1. Primary Classes

Table 1 shows actions in the primary classes. Primary classes are actions that contain 20 samples per class. We also include the mean average precision of each class.

1.2. Additional Classes

Table 2 shows actions in the primary classes. Additional classes are actions that contain two samples per class, which are dedicated to one-shot learning. It helps evaluate the performance of the proposed model to differentiate the action when very limited data are presented.

2. Globally Aligned Skeletons

Globally aligned skeletons are skeletons that we manually rotated so that all the samples are facing the negative z -direction at the start of the action. This adjustment eliminates view variances so that all actions to be performed in the same coordinates space. There are 40 different actions in the primary classes containing globally aligned skeletons. We will provide the list in Table 3.

3. HAA4D Evaluation Benchmark

For actions with 20 examples per class, video indexes 0 to 9 are used for training, while videos from 10 to 19 are used for testing. We perform data augmentation on the first ten samples, and among all, videos 8 and 9 are used for validation. For actions that contain only two samples, the one with a smaller index serves as the query, and the other serves as the support.

4. Training Configuration

For training the global alignment network, we perform data augmentation by sampling camera views from a 3-

frequency subdivision icosahedron. This gives us 92 additional training samples per example. Since there are 400 examples in HAA4D that are provided with globally aligned skeletons, with the help of data augmentation, we have 36,800 examples of training our global alignment network. We split all the data into training and validated with a ratio of 0.8 under two settings: cross-views and cross-actions. We select 73 views on the icosahedron sphere for cross-views and test the rest 19 views to ensure that our network generates predictions decently while encountering unseen views. We also trained our network on cross-views, i.e., 32 out of the 40 classes, to secure that the model is used to generalize different actions. Here are our training environment and configurations in more details:

- GPU: GeForce GTX TITAN X and GeForce GTX 1080 Ti
- Epochs: 300
- Batch size: 64
- Optimizer: Adam (starting learning rate: 1e-4, weight decay rate: 1e-6)
- Loss weight: 10:1 (rotation loss : reconstruction loss)

5. Limitations

Our dataset contains only a few samples per class, which is intended for enhancing its scalability and extension (e.g. 1 for train and 1 for test, see our 5-way 1-shot experiments in main paper), as unlike the 2D counterparts, HAA4D's 3D+T skeletons have more degree of freedom making meaningful data augmentation easy for training on large datasets. In our case, we sample the 3D skeletons from different viewpoints and rotate the skeletons accordingly. We can also use mirroring or combining skeleton samples in the same class to introduce more variation to the dataset. Unlike 2D skeletons that can only have one rotation parameter, the properties of 3D skeletons help better perform data augmentation without breaking the original skeleton structure.

Our few-shot skeleton-based action recognition architecture currently supports only single-person actions. To

1: ALS_IceBucket_Challenge (71.65)	2: CPR (93.86)	3: abseiling (92.08)	4: add_new_car_tire (97.45)	5: adjusting_glasses (14.61)
6: air_drumming (67.30)	7: air_guitar (60.32)	8: air_hockey (30.83)	9: alligator_wrestling (95.37)	10: applauding (13.73)
11: applying_cream (1.28)	12: archery (22.82)	13: arm_wave (76.82)	14: arm_wrestling (24.19)	15: atlant_throw (64.64)
16: axe_throwing (54.45)	17: backflip (81.08)	18: backward_roll (77.50)	19: badminton_overswing (57.07)	20: badminton_serve (57.07)
21: badminton_underswing (64.02)	22: balancebeam_flip (33.77)	23: balancebeam_jump (65.63)	24: balancebeam_rotate (76.87)	25: balancebeam_spin (34.47)
26: balancebeam_walk (73.19)	27: bandaging (43.44)	28: baseball_bunt (96.53)	29: baseball_catch_catcher (79.88)	30: baseball_catch_groundball (91.92)
31: baseball_pitch (52.28)	32: baseball_swing (91.10)	33: basketball_dribble (97.64)	34: basketball_jabstep (93.37)	35: basketball_layup (91.75)
36: basketball_pass (78.67)	37: basketball_shoot (62.81)	38: battle-rope_jumping-jack (96.08)	39: battle-rope_power-slam (93.65)	40: battle-rope_rainbow (98.22)
41: battle-rope_russian-twist (97.14)	42: battle-rope_sideplank (96.29)	43: battle-rope_snake (97.66)	44: battle-rope_wave (97.42)	45: beer_pong_throw (44.87)
46: bench_dip (88.79)	47: bending_back (76.46)	48: bowing_waist (61.65)	49: brushing_floor (91.05)	50: burpee (94.06)
51: climb_stair (86.19)	52: fish_hunting_hold (95.77)	53: floss_dance (79.09)	54: gym_squat (86.11)	55: jumping_jack (82.69)
56: play_accordian (36.03)	57: play_bagpipes (49.74)	58: play_bangu (85.08)	59: play_banjo (41.72)	60: play_castanets (33.98)
61: play_cello (83.35)	62: play_clarinet (20.90)	63: play_cornett (15.00)	64: play_cymbals (51.04)	65: play_diabolo (84.00)
66: play_doublebass (77.34)	67: play_erhu (69.33)	68: play_gong (63.88)	69: play_grandpiano (81.06)	70: play_guitar (70.81)
71: play_handpan (79.50)	72: play_harmonica (34.92)	73: play_harp (25.47)	74: play_hula hoop (94.84)	75: play_hulusi (19.34)
76: play_kendama (86.82)	77: play_leaf_flute (27.48)	78: play_lute (66.84)	79: play_maracas (39.83)	80: play_melodic (41.41)
81: play_noseflute (15.29)	82: play_ocarina (50.90)	83: play_otamatone (12.18)	84: play_panpipe (7.74)	85: play_piccolo (15.21)
86: play_recorder (10.25)	87: play_sanxian (62.29)	88: play_saw (59.23)	89: play_saxophone (36.25)	90: play_serpent (72.47)
91: play_sheng (44.04)	92: play_sitar (66.23)	93: play_suona (26.76)	94: play_tambourine (49.16)	95: play_thereminvox (59.91)
96: play_timpani (76.29)	97: play_triangle (24.66)	98: play_trombone (12.92)	99: play_trumpet (1.77)	100: play_ukulele (27.16)
101: play_viola (39.31)	102: play_violin (32.91)	103: play_xylophone (79.20)	104: play_yangqin (26.54)	105: play_yoyo (69.92)
106: playing_bass_drum (41.44)	107: playing_cajon_drum (94.21)	108: playing_conga_drum (71.20)	109: playing_nunchucks (84.58)	110: playing_rubiks_cube (31.42)
111: playing_seesaw (63.97)	112: playing_snare_drum (82.11)	113: playing_taiko_drum (49.75)	114: pottery_wheel (90.88)	115: pouring_wine (63.42)
116: pull_ups (84.94)	117: punching_sandbag (71.29)	118: punching_speedbag (26.99)	119: push_car (76.94)	120: push_wheelchair (73.89)
121: push_wheelchair_alone (86.03)	122: putting_scarf_on (54.48)	123: quadruped_hip-extension (90.14)	124: racewalk_walk (65.38)	125: read_newspaper (22.82)
126: reading_book (26.17)	127: rescue_breathing (36.16)	128: sandboarding_forward (82.46)	129: side_lunge (84.16)	130: situp (88.16)
131: water_skiing (90.00)	132: watering_plants (75.21)	133: wear_face_mask (8.97)	134: wear_helmet (29.33)	135: weightlifting_hang (96.55)
136: weightlifting_overhead (98.39)	137: weightlifting_stand (97.49)	138: whipping (96.19)	139: whistle_one_hand (22.97)	140: whistle_two_hands (4.89)
141: workout_chest-pull (84.69)	142: workout_crunch (95.77)	143: yawning (18.66)	144: yoga_bridge (98.18)	145: yoga_cat (93.47)
146: yoga_dancer (97.69)	147: yoga_firefly (96.16)	148: yoga_fish (96.51)	149: yoga_gate (97.60)	150: yoga_locust (90.82)
151: yoga_lotus (97.93)	152: yoga_pigeon (97.67)	153: yoga_tree (98.45)	154: yoga_triangle (98.03)	155: yoga_updog (92.47)

Table 1. Actions in the primary classes with their mean average precision (mAP). Overall mAP: 63.71.

156: balloon_animal	157: base_jumping	158: baseball_catch_flyball	159: baseball_run	160: basketball_dunk
161: basketball_hookshot	162: belly_dancing	163: bike_fall	164: billiard_hit	165: blow_gun
166: blowdrying_hair	167: blowing_balloon	168: blowing_glass	169: blowing_gum	170: blowing_kisses
171: blowing_leaf	172: blowing_nose	173: bmx_jumping	174: bmx_riding	175: bowing_fullbody
176: bowling	177: bowls_throw	178: breakdancing_flare	179: breakdancing_flip	180: breakdancing_rotate
181: breakdancing_support	182: brushing_hair	183: brushing_teeth	184: building_snowman	185: burping
186: calfropes_catch	187: calfropes_ropes	188: calfropes_subdue	189: canoeing_slalom	190: canoeing_sprint
191: card_throw	192: carrying_with_head	193: cartwheeling	194: cast_net	195: chainsaw_log
196: chainsaw_tree	197: chalkboard	198: chewing_gum	199: chopping_meat	200: chopping_wood
201: cleaning_mirror	202: remove_car_tire	203: ride_bike	204: ride_horse	205: ride_motorcycle
206: ride_scooter	207: riding_elephant	208: riding_mechanical_bull	209: rock_balancing	210: rock_paper_scissors
211: roller-skating_backward	212: roller-skating_forward	213: rolling_snow	214: rowing_boat	215: running_in_place
216: running_on_four	217: runway_walk	218: sack_race	219: salute	220: screw_car_tire
221: scuba_diving	222: shake_cocktail	223: shaking_head	224: shaving_beard	225: shoe_shining
226: shoot_dance	227: shooting_handgun	228: shooting_shotgun	229: shotput_throw	230: shoveling_snow
231: shuffle_dance	232: skateboard_forward	233: skateboard_grind	234: skateboard_jump	235: ski_backflip
236: ski_cork	237: ski_frontflip	238: ski_jump_land	239: ski_jump_midair	240: ski_jump_slide
241: skydiving	242: sledgehammer_strike_down	243: sling	244: slingshot	245: smoking_exhale
246: smoking_inhale	247: snorkeling	248: snow_angel	249: snowboard_jump	250: snowboard_slide
251: soccer_dribble	252: soccer_header	253: soccer_save	254: soccer_shoot	255: soccer_throw
256: softball_pitch	257: speed_stack	258: speedskating_forward	259: spinning_basketball	260: spinning_book
261: spinning_plate	262: spitting_on_face	263: split_leap	264: spraying_wall	265: sprint_kneel
266: sprint_run	267: sprint_start	268: squash_backhand	269: squash_forehand	270: stomping_grapes
271: stone_skipping	272: styling_hair	273: surfing	274: swimming_backstroke	275: swimming_breast_stroke
276: swimming_butterfly_stroke	277: swimming_freestyle	278: swinging_axe_on_a_tree	279: sword_swallowing	280: taekwondo_high_block
281: taekwondo_kick	282: taekwondo_low_block	283: taekwondo_middle_block	284: taekwondo_punch	285: taichi_fan
286: taking_photo_camera	287: taking_selfie	288: talking_megaphone	289: talking_on_phone	290: tap_dancing
291: tennis_backhand	292: tennis_forehand	293: tennis_serve	294: three_legged_race	295: throw_boomerang
296: throw_paper_plane	297: throwing_bouquet	298: tight-rope_walking	299: tire_pull	300: tire_sled

Table 2. Actions in the additional classes.

accommodate actions involving more than one person, we can use a similar technique as ST-GCN, which utilizes the two skeletons with the highest confidence score in the sequences. For actions that have only one subject, we assume them to be all zeros. With this, we can modify our architecture so that instead of having the shape of $(n_ways, n_shots,$

$n_segments, n_encodings)$ for explicit skeletons encodings, we add one additional dimension so that the skeletons are in the shape of $(n_ways, n_shots, 2, n_segments, n_encodings)$. We then calculate the distance, respectively. Since there are four skeleton sequences, assuming $q_s1, q_s2, s_s1,$ and s_s2 all with the shape of $(1, n_segments, n_encodings)$, we use

1: ALS_IceBucket_Challenge	3: abseiling	4: add_new_car_tire	5: adjusting_glasses	6: air_drumming
7: air_guitar	8: air_hockey	9: alligator_wrestling	10: applauding	11: applying_cream
12: archery	13: arm_wave	14: arm_wrestling	15: atlatl_throw	16: axe_throwing
17: backflip	18: backward_roll	19: badminton_overswing	20: badminton_serve	21: badminton_underswing
25: balancebeam_spin	26: balancebeam_walk	27: bandaging	28: baseball_bunt	30: baseball_catch_groundball
31: baseball_pitch	32: baseball_swing	33: basketball_dribble	34: basketball_jabstep	35: basketball_layup
36: basketball_pass	37: basketball_shoot	38: battle-rope_jumping-jack	39: battle-rope_power-slam	40: battle-rope_rainbow
41: battle-rope_russian-twist	42: battle-rope_sideplank	43: battle-rope_snake	46: bench_dip	50: burpee

Table 3. Actions the contain globally aligned skeletons.

the minimum distance between the possible combination $d(q_{-s1}, s_{-s1}) + d(q_{-s2}, s_{-s2}), d(q_{-s1}, s_{-s2}) + d(q_{-s2}, s_{-s1})$. Therefore, we can make this adjustment for multi-person interaction, and the rest of the architecture can remain unchanged.