Appendix for "Revisiting Skeleton-based Action Recognition"

Haodong Duan[1,3] Yue Zhao[2] Kai Chen[3,5] Dahua Lin[1,3] Bo Dai[3,4] ✉
[1]The Chinese University of HongKong [2]The University of Texas at Austin
[3]Shanghai AI Laboratory [4]S-Lab, Nanyang Technological University [5]SenseTime Research

Figure 5. **The extracted skeletons of the NTURGB+D dataset.** The actions of the visualized frames are: "cheer up", "touch other person's pocket", "jump up", "put the palms together", "taking a selfie", "shake fist".

1. Visualization

We provide more visualization of the extracted pose of the four datasets: FineGYM, NTURGB+D, Kinetics400, Volleyball to demonstrate the performance of the proposed pose extraction approach qualitatively. You can watch the corresponding videos at https://youtu.be/oS7fX9Eg2ws.

NTURGB+D [12, 17]. Figure 5 displays some examples of extracted skeletons of NTURGB+D. Our pose extractor achieves almost perfect performance on NTURGB+D due to the simple scenarios: the background scene is not complicated, while there are two persons at most in each frame, with little occlusion.

FineGYM [18]. Figure 6 displays some examples of extracted skeletons of FineGYM. Although we perform pose extraction with ground-truth bounding boxes of the athletes, the extracted 2D poses are far from perfect. The pose extractor is extremely easy to make mistakes for poses the rarely occur in COCO-keypoint [11] or when motion blur occurs. Even though the quality of extracted skeletons are not satisfying, they are still discriminative enough for skeleton-based action recognition.

Figure 6. **The extracted skeletons of the FineGYM dataset.** The extracted skeletons are far from perfect, but discriminative enough for action recognition.

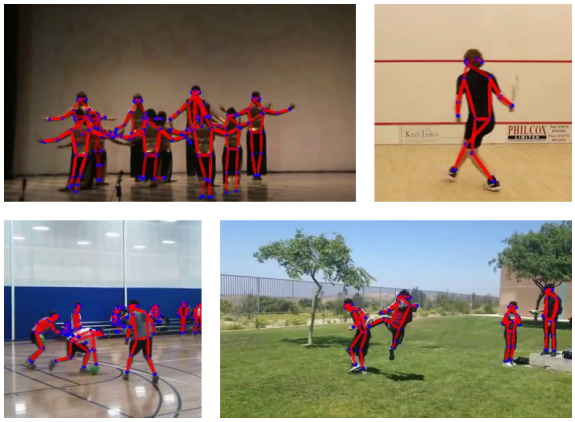


Figure 7. **The extracted skeletons of the Kinetics400 dataset.**

Kinetics400 [1]. Kinetics400 is not a human-centric dataset for action recognition. In Kinetics videos, the person locations, scales, and the number of persons may vary a lot, which makes extracting human skeletons of Kinetics400 much more difficult than NTURGB+D or FineGYM.

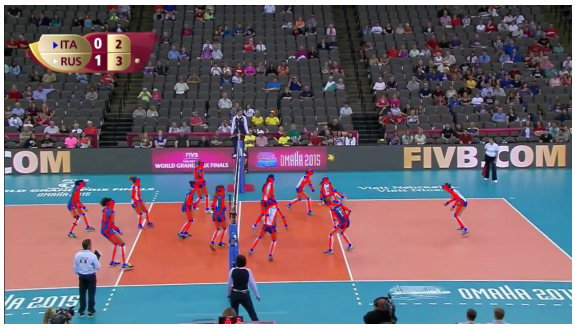


Figure 8. **The extracted skeletons of the Volleyball dataset.**

In Figure 7, we provide some examples that our pose estimator accurately predicts the human skeletons. We also discuss some failure cases in Sec 4.7.

Volleyball [7]. Volleyball is a group activity recognition dataset. Each frame of a video contains around a dozen people (six for each team). Most of the human poses in a volleyball video are regular ones (unlike FineGYM). In Figure 8, we see that our pose extractor can predict the human pose of each person accurately.

2. Generating Pseudo Heatmap Volumes.

In this section, we illustrate how we generate the pseudo heatmap volumes, the input of PoseConv3D. We also provide a jupyter notebook named `GenPseudoHeatmaps.ipynb` in supplementary materials, which can extract skeleton keypoints from RGB videos (optional) and generate pseudo heatmaps based on the skeleton keypoints.

Figure 9 illustrates the pipeline of pose extraction (RGB video $\rightarrow$ coordinate-triplets) and generating pseudo heatmap volumes (coordinate-triplets $\rightarrow$ 3D heatmap volumes). The visualization in Figure 9 is just for one frame, while you can find the visualization for the entire video in the jupyter notebook. Since the heatmaps are of K channels ($K = 17$ for COCO-keypoints), we visualize the heatmap in one 2D image with color encoding. The pose extraction part is straight-forward: we use a Top-Down pose estimator instantiated with HRNet [19] to extract the 2D poses for each person in each frame, and save the extracted poses as coordinate-triplets: (x, y, score). For generating pseudo heatmaps, we first perform uniform sampling, which will sample T ($T = 32$ or 48 in experiments) frames uniformly from the video and discard the remaining frames. After that, we will find a global cropping box (The red box in Figure 9, same for all T frames) that envelops all persons in the video, and crop all T frames with that box to reduce the spatial size (as illustrated in Figure 9). In `GenPseudoHeatmaps.ipynb`, you can run the entire pipeline to process a video from the NTURGB-D dataset.

3. Detailed Architectures of PoseConv3D and RGBPose-Conv3D

3.1. Different variants of PoseConv3D.

In Table 11, we demonstrate the architectures of the three backbones we adapted from RGB-based action recognition as well as their variants:

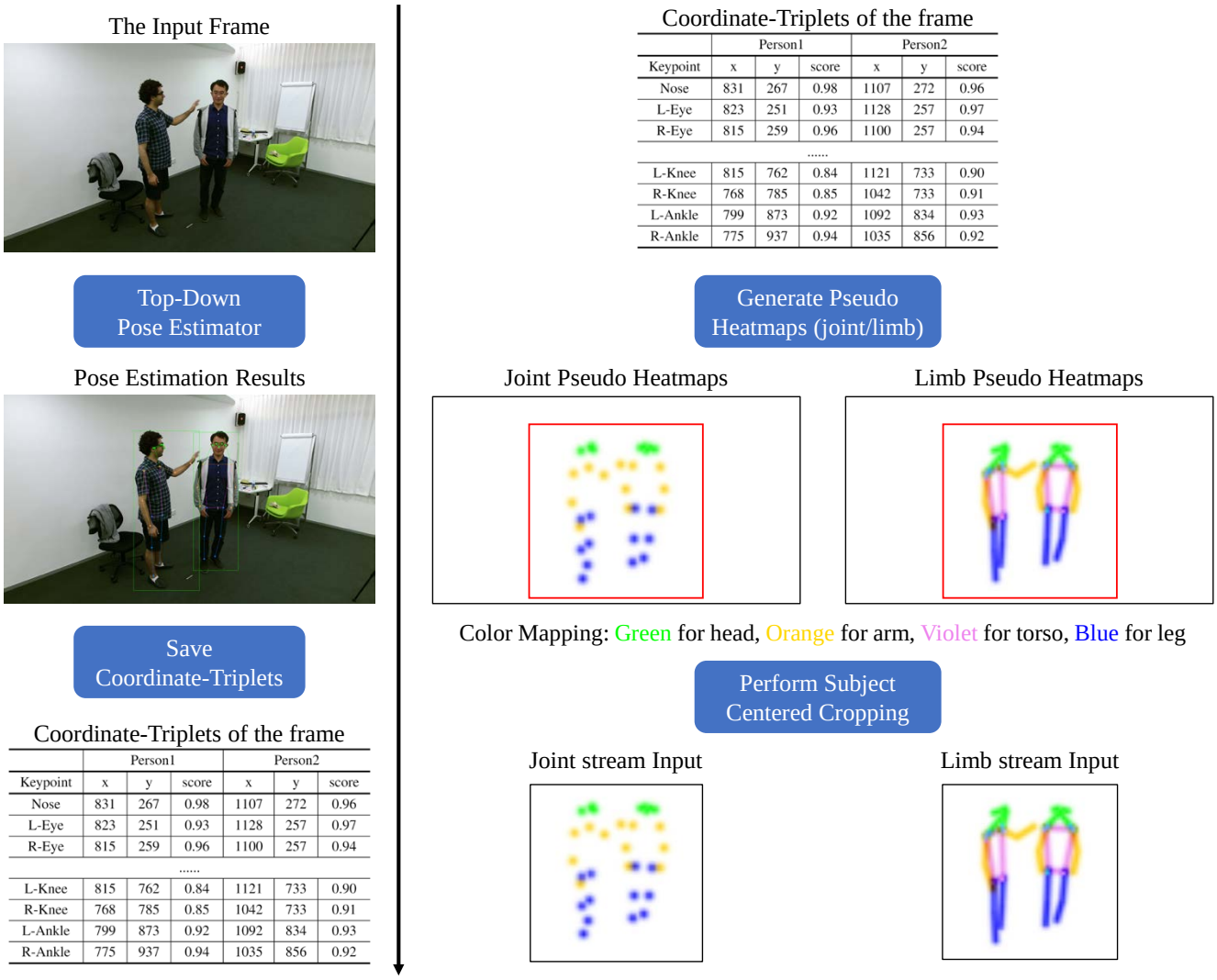
C3D [20]. C3D is one of the earliest 3D-CNN developed for RGB-based action recognition (like AlexNet [9] for image recognition), which consists of eight 3D convolution layers. To adapt C3D for skeleton-based action recognition, we reduce its channel-width to half ($64 \rightarrow 32$) for better efficiency. In addition, for Pose-C3D-s, we remove the last two convolution layers.

X3D [4]. X3D is a recent state-of-the-art 3D-CNN for action recognition. Replacing vanilla convolutions with depth-wise convolutions, X3D achieves competitive recognition performance with tiny amounts of parameters and FLOPs. The architecture of the adapted Pose-X3D is almost unchanged compared to the original X3D-S, except that we remove the original first stage. For Pose-X3D-s, we remove convolution layers from each stage uniformly by changing the hyper-parameter γ_d from 2.2 to 1.

SlowOnly [5]. SlowOnly is a popular 3D-CNN used for RGB-based action recognition. It is obtained by inflating the ResNet layers in the last two stages from 2D to 3D. To adapt SlowOnly for skeleton-based action recognition, we reduce its channel-width to half ($64 \rightarrow 32$) as well as remove the original first stage in the network. We also have conducted experiments with Pose-SlowOnly-wd (with channel-width 64) and Pose-SlowOnly-HR (with 2x larger input and deeper network). There is no performance improvement despite the much heavier backbone.

3.2. RGBPose-Conv3D instantiated with SlowOnly.

RGBPose-Conv3D is a general framework for RGB-Pose dual-modality action recognition, which can be instantiated with various 3D-CNN backbones. In this work, we instantiate both pathways with the SlowOnly network. As shown in Table 12, the RGB pathway has a smaller frame rate and a larger channel width since RGB frames are low-level features. On the contrary, the Pose pathway has a larger frame rate and a smaller channel width. Time stride convolutions are used as bi-directional lateral connections between the two pathways (after res_3 and res_4) so that semantics of different modalities can sufficiently interact. Besides lateral connections, the predictions of two pathways are also combined in a late fusion manner, which leads to further improvements in our empirical study. RGBPose-Conv3D is trained with two individual losses respectively for each pathway, as a single loss that jointly learns from two modalities leads to severe overfitting.



	Person1			Person2		
Keypoint	x	y	score	x	y	score
Nose	831	267	0.98	1107	272	0.96
L-Eye	823	251	0.93	1128	257	0.97
R-Eye	815	259	0.96	1100	257	0.94
			.....			
L-Knee	815	762	0.84	1121	733	0.90
R-Knee	768	785	0.85	1042	733	0.91
L-Ankle	799	873	0.92	1092	834	0.93
R-Ankle	775	937	0.94	1035	856	0.92

Figure 9. **The pipeline of generating the input of PoseConv3D. Left, Pose Extraction:** We perform Top-Down pose estimation for each single frame. The estimated 2D poses are saved as coordinate-triplets: (x, y, score). **Right, Generating Pseudo Heatmap Volumes:** Based on the coordinate-triplets, we generate pseudo heatmaps for joints and limbs using Eq 1, 2 in the main paper. We perform subjects-centered cropping and uniform sampling to make the heatmap volumes compact.

4. Supplementary Experiments

4.1. Ablation Study on Pose Extraction

This section discusses different alternatives that can be adopted in pose extraction to validate our choice. The input size for all 3D-CNN experiments is $T \times H \times W = 48 \times 56 \times 56$.

2D v.s. 3D Skeletons. We first compare the recognition performance of using 2D and 3D skeletons for action recognition. The 3D skeletons are either collected by sensors (NTU-60) or estimated with state-of-the-art 3D pose estimators based on RGB inputs [8, 21] (FineGYM). For a fair comparison, we use MS-G3D [14] (the current state-of-the-art GCN for skeleton-based action recognition) with the same configuration and training schedule for 2D and 3D keypoints and list the results in Table 13a. The estimated 2D keypoints (even low-quality ones) consistently outperform 3D keypoints (sensor collected or estimated) in action recognition. Besides RGB-based 3D-pose estimators, we also consider the 'lifting' approaches [15, 16], which directly 'lift' 2D-pose (sequences) to 3D-pose (sequences). We regress the 3D poses based on 2D poses extracted with HRNet, use the lifted 3D poses for action recognition. The results in Table 13b indicate that such lifted 3D poses do not provide any additional information, performs even worse than the original 2D poses in action recognition.

Bottom-Up v.s. Top-Down. To compare the pose estimation quality of Bottom-Up and Top-Down approaches,

Table 11. **The architecture of PoseConv3D instantiated with three backbones: C3D, X3D, SlowOnly.** The dimensions of kernels are denoted by $T \times S^2, C$ for temporal, spatial, channel sizes. Strides are denoted with T, S^2 for temporal and spatial strides. GAP denotes global average pooling.

stage	C3D-s	C3D	X3D-s	X3D	SlowOnly	SlowOnly-wd	SlowOnly-HR
data layer	Uniform 48, 56 ×56						Uniform 48, 112 ×112
stem layer	conv 3×3^2, 32		conv 1×3^2, 24 stride 1, 2^2 conv 5×1^2, 24		conv 1×7^2, 32	conv 1×7^2, 64	conv 1×7^2, 32
stage1	maxpool 1×2^2 $[3\times3^2, 64]\times1$		$\begin{bmatrix}1\times1^2, 54\\3\times3^2, 54\\1\times1^2, 24\end{bmatrix}\times2$	$\begin{bmatrix}1\times1^2, 54\\3\times3^2, 54\\1\times1^2, 24\end{bmatrix}\times5$	None		$\begin{bmatrix}1\times1^2, 32\\1\times3^2, 32\\1\times1^2, 128\end{bmatrix}\times3$
stage2	maxpool 1×2^2 $[3\times3^2, 128]\times2$		$\begin{bmatrix}1\times1^2, 108\\3\times3^2, 108\\1\times1^2, 48\end{bmatrix}\times5$	$\begin{bmatrix}1\times1^2, 108\\3\times3^2, 108\\1\times1^2, 48\end{bmatrix}\times11$	$\begin{bmatrix}1\times1^2, 32\\1\times3^2, 32\\1\times1^2, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1^2, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times4$	
stage3	maxpool 1×2^2 $[3\times3^2, 256]\times2$		$\begin{bmatrix}1\times1^2, 216\\3\times3^2, 216\\1\times1^2, 96\end{bmatrix}\times3$	$\begin{bmatrix}1\times1^2, 216\\3\times3^2, 216\\1\times1^2, 96\end{bmatrix}\times7$	$\begin{bmatrix}\underline{3\times1^2}, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times6$	$\begin{bmatrix}\underline{3\times1^2}, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times6$	
stage4	None	$[3\times3^2, 256]\times2$	conv 1×1^2, 216		$\begin{bmatrix}\underline{3\times1^2}, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times3$	$\begin{bmatrix}\underline{3\times1^2}, 256\\1\times3^2, 256\\1\times1^2, 1024\end{bmatrix}\times3$	
	GAP, fc						

Table 12. **RGBPose-Conv3D instantiated with the SlowOnly backbone.** The dimensions of kernels are denoted by $T \times S^2, C$ for temporal, spatial, channel sizes. Strides are denoted with T, S^2 for temporal and spatial strides. The backbone we use is ResNet50. GAP denotes global average pooling.

stage	*RGB* Pathway	*Pose* Pathway	output sizes T×S^2
data layer	uniform 8,1^2	uniform 32,4^2	*RGB*: 8×224^2 *Pose*: 32×56^2
stem layer	conv 1×7^2, 64 stride 1, 2^2 maxpool 1×3^2 stride 1, 2^2	conv 1×7^2, 32 stride 1, 1^2	*RGB*: 8×56^2 *Pose*: 32×56^2
res_2	$\begin{bmatrix}1\times1^2, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times3$	N.A.	*RGB*: 8×56^2 *Pose*: 32×56^2
res_3	$\begin{bmatrix}1\times1^2, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1^2, 32\\1\times3^2, 32\\1\times1^2, 128\end{bmatrix}\times4$	*RGB*: 8×28^2 *Pose*: 32×28^2
res_4	$\begin{bmatrix}\underline{3\times1^2}, 256\\1\times3^2, 256\\1\times1^2, 1024\end{bmatrix}\times6$	$\begin{bmatrix}\underline{3\times1^2}, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times6$	*RGB*: 8×14^2 *Pose*: 32×14^2
res_5	$\begin{bmatrix}\underline{3\times1^2}, 512\\1\times3^2, 512\\1\times1^2, 2048\end{bmatrix}\times3$	$\begin{bmatrix}\underline{3\times1^2}, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times3$	*RGB*: 8×7^2 *Pose*: 32×7^2
	GAP, fc	GAP, fc	# classes

we instantiate the two approaches with the same backbone (HRNet-w32). Besides, we also instantiate the Top-Down approach with the MobileNet-v2 backbone for comparison, which has a similar performance to HRNet (Bottom-Up) on COCO-validation. We use extracted 2D poses to train a *Pose-SlowOnly* on NTU-60. Table 13c shows that the performance of HRNet (Bottom-Up) on COCO-val is much worse than HRNet (Top-Down) and close to MobileNet (Top-Down). However, the Top-1 accuracy of HRNet (Bottom-Up) is much higher than MobileNet (Top-Down) and close to HRNet (Top-Down). Although the potential of Bottom-Up should not be neglected, considering the better performance and faster inference speed (Top-Down runs faster when there aren't many persons in a frame), we use Top-Down for pose extraction in this work.

Interested Person v.s. All Persons. Many people may exist in a video, but not all of them are related to the interested action. For example, in FineGYM, only the pose of the athlete is helpful, while other persons like the audience or referee are unrelated. We compare using 3 kinds of person bounding boxes for pose extraction: **Detection**, **Tracking**(with Siamese-RPN [10]) and **GT** (with increasing prior about the athlete). In Table 13d, we see that the prior of the interested person is extremely important: even weak prior knowledge (1 GT box per video) can improve the performance by a large margin.

Coordinates v.s. Heatmaps. Storing 3D heatmap volumes may take vast amounts of disk space. To be more efficient, we can save the 2D poses as coordinate-triplets (x, y, score) and restore them to 3D heatmap volumes following the methods we introduced in Sec **??**. We conduct

Table 13. **Ablation study on Pose Extraction.**

Input	GYM	NTU-60
Kinect-3D [24]	N.A.	89.4
DOPE-3D [21]	76.3	N.A.
VIBE-3D [8]	87.0	N.A.
HRNet-2D [19]	**92.0**	**91.9**
MobileNet-2D [6]	**89.0**	**90.2**

(a) **2D skeleton v.s. 3D skeleton.**

Input	GYM
DOPE [21]	76.3
VIBE [8]	87.0
FrameLift [15]	90.0
VideoLift [16]	90.2
HRNet-2D [19]	92.0

(b) **Lifted 3D-pose doesn't help in recognition.**

Pose Estimator	COCO AP	NTU-60
HRNet (Top-Down)	**0.746**	**93.6**
HRNet (Bottom-Up)	0.654	93.0
Mobile (Top-Down)	0.646	92.0

(c) **Pose Estimation: Top-Down v.s. Bottom-Up.**

Human Proposals	GYM Mean-Top1
Detection	75.8
Tracking	85.3
GT	**92.0**

(d) **Pose extracted with different boxes.**

Input Format	GYM Mean-Top1
Coordinate-MobileNet	90.7
Coordinate-HRNet	93.2
Heatmap-MobileNet	92.7
Heatmap-HRNet	93.6

(e) **Coordinate v.s. Heatmap.**

Table 14. **Transferring Ability.** Skeleton representations learned on the large-scale Kinetics400 can transfer to downstream datasets well. Backbone parameters are frozen for the 'Linear' setting.

Policy	HMDB51	UCF101
Scratch	58.6	79.1
Linear	64.9	83.1
Finetune	**69.3**	**87.0**

Table 15. **Comparison with state-of-the-art multi-modality action recognition approaches.**

Method	HMDB51	UCF101
I3D [1]	80.7	98.0
PoTion [2]	43.7	65.2
PoTion + I3D	80.9	98.2
PA3D [22]	55.3	-
PA3D + I3D	82.1	-
PoseConv3D	**69.3**	**87.0**
PoseConv3D + I3D	**82.7**	**98.4**

experiments on FineGYM to explore how much information is lost during the heatmap $\rightarrow$ coordinate compression. In Table 13e, we see that for low-quality pose estimators, it leads to a 2% drop in Mean-Top1. For high-quality ones, the degradation is more moderate (only a 0.4% Mean-Top1 drop). Thus we choose to store coordinates instead of 3D heatmap volumes.

4.2. Multi-Modality Results Action Recognition on UCF101 and HMDB51

In main paper Table 5, we train different PoseConv3D on UCF101 and HMDB51 from scratch. In this section, we demonstrate that PoseConv3D can also take advantage of pretraining on large-scale datasets. We adopt weights pretrained on Kinetics400 to initialize the PoseConv3D. Pretraining with skeleton data from the large-scale Kinetics400 benefits the downstream recognition tasks on smaller datasets, under both 'Linear' and 'Finetune' paradigms (Table 14).

We further compare PoseConv3D with previous state-of-the-arts of skeleton-based action recognition on UCF101 and HMDB51: PoTion [2] and PA3D [22]. For a fair comparison, we fuse the skeleton-based predictions with I3D [1] predictions, instead of predictions from the more advanced OmniSource [3]. Table 15 shows that PoseConv3D not only outperforms other approaches by a large margin on skeleton-based action recognition, but also leads to better overall performance after fusing with predictions based on other modalities.

4.3. Using 3D Skeletons in PoseConv3D

PoseConv3D takes stacked 2D skeleton keypoint heatmaps as input. Assume only 3D skeletons are available for a target dataset, one can also use the 3D skeletons in PoseConv3D by projecting them to a 2D plane. The NTURGB+D dataset [17] provides 3D skeleton sequences collected by Microsoft Kinect v2 sensors [24]. Besides, the dataset also includes the projection of 3D joints onto the 2D image coordinate systems. We use the projected 2D skeletons of NTU-60 as the input for PoseConv3D and study the effect.

Table 16 demonstrates the recognition performance of using projected 2D skeletons in PoseConv3D. Using the projected 2D skeletons as inputs instead of the original 3D skeletons, there is a 2% Top-1 accuracy drop for MS-G3D due to the information lost in 3D $\rightarrow$ 2D compression. If both use 2D skeletons as input, PoseConv3D outperforms

Table 16. **PoseConv3D with projected 2D poses.** We report the recognition performance of the joint model.

Input + Method	Top-1
2D-projection + MS-G3D [14]	86.8
3D-skeleton + MS-G3D [14]	88.8[1]
2D-projection + PoseConv3D	**89.2**

the GCN-based counterpart by 2.4%, even surpasses the MS-G3D with 3D skeletons as input by 0.4%, which indicates the great spatiotemporal modeling capability of 3D-CNN can compensate for the information lost in 3D $\rightarrow$ 2D projection.

4.4. Ablation on the Practice for Group Activity Recognition

In experiments, we find that representing all people with a single heatmap volume is the best practice for group activity recognition with PoseConv3D. On the Volleyball dataset, we have also explored three alternatives that process different persons' heatmaps separately: **A.** For each joint, we allocate N channels for N persons. The PoseConv3D input then has $N \times K$ channels (instead of K); **B.** We generate a 3D heatmap volume ($K \times T \times H \times W$) for each person and use PoseConv3D (weights shared among N persons) to extract the skeleton feature separately. We use average pooling to aggregate N persons' features to a single feature vector; **C.** On top of **B**, we insert several (1 to 3) encoder layers (from scratch or with **B** pre-training) before the average pooling for inter-person modeling. Figure 10 provides an illustration of three alternatives. For **A**, the high dimensional input leads to severe overfitting. The Top-1 accuracy is only 75.3%. For **B, C**, despite the great amounts of computation ($> 13\times$) consumed, the recognition performance is not satisfying. At best, **B**, **C** achieves 85.7% and 87.9% Top-1 on Volleyball, still much inferior to accumulating heatmaps (91.3%). Accumulating heatmaps is a simple and relatively good solution for balancing complexity and effectiveness. More complex designs may lead to further improvements, which is left to future work.

4.5. Uniform Sampling for RGB-based recognition

Based on the outstanding improvement by uniform sampling on skeleton-based action recognition, we wonder if this sampling strategy also works for RGB-based action recognition. Thus we apply uniform sampling to RGB-based action recognition on NTU-60 [17] and GYM [18]. We use SlowOnly-R50 [5] as the backbone and set the input length as 16 frames. From Table 17, we see that uniform sampling also outperforms fix-stride sampling by a large

[1]We rerun the official code of MS-G3D to get this accuracy.

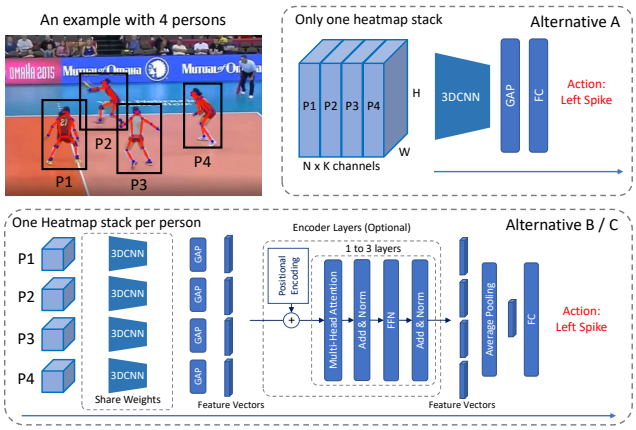


Figure 10. An illustration (with 4 persons) of the proposed three alternatives for group activity recognition. Best viewed with 4x zoom in.

Table 17. **Uniform sampling also works for RGB-based action recognition.** Alls results are for 10-clip testing, except the 'uniform-16 (1c)', which uses 1-clip testing.

(a) **FineGYM.**

Sampling	Mean-Top1
16x2	87.9
16x4	88.7
uniform-16 (1c)	**91.1**
uniform-16	**91.6**

(b) **NTU-60 (X-Sub)**

Sampling	Top1
16x2	94.9
16x4	95.1
uniform-16 (1c)	**95.7**
uniform-16	**96.1**

margin in RGB-based recognition on these two datasets: the accuracy of uniform sampling with 1-clip testing is better than the accuracy of fix-stride sampling with 10-clip testing. We mainly attribute the advantage of uniform sampling to the highly variable video lengths in these two datasets. On the contrary, we observe a slight accuracy drop on Kinetics400[2] when applying uniform sampling: for SlowOnly-R50 with input length 8, the Top-1 accuracy drops from 75.6% to 75.2%.

4.6. NTU-60 Error Analysis

On NTU-60 X-Sub split, we achieve 94.1% Top-1 accuracy with skeleton-based action recognition, which outperforms the current state-of-the-art result by 2.6%. To further study the failure cases, we first define the confusion score S of a pair of the action classes i, j as:

$$S = n_{ij} + n_{ji} \tag{1}$$

n_{ij} indicates the number of videos belong to the class i but recognized as class j. In NTU-60, there are 1770 pairs of action classes in total, while we list the five most confus-

[2]In Kinetics400, most video clips are of the same temporal length: 10 seconds.

Table 18. **Top 5 confusion pairs of skeleton-based action recognition on NTU-60 X-Sub.** Multi-modality fusion with *RGBPose*-Conv3D improves the recognition performance on confusion pairs by a lot.

Action1	Action2	S_{Pose}	$S_{\text{RGB+Pose}}$
Read	Play with phone/tablet	67	13
Write	Type on a keyboard	57	20
Write	Play with phone/tablet	50	5
Take a selfie	Point to sth. with finger	48	10
Read	Write	44	24

Figure 11. **Problems in Kinetics400 Pose Extraction.** Left: Human missing in action 'kayaking'. Middle: Human skeleton is too small to be recognized in action 'diving cliff'. Right: Only human parts appear, the pose estimator fails ('washing feet').

Table 19. **Mean class accuracy on the Kinetics-Motion subset.**

Method	Top1 Acc
Swin-L [13]	92.7
ST-GCN [23]	72.0
PoseConv3D	81.9
Swin-L + PoseConv3D	**94.7**

ing pairs in Table 18. Most failure cases are of these top-confusing pairs, *e.g.*, over 27% failure cases are of the top 5 confusion pairs. It is hard to distinguish these pairs of actions with human skeletons only.

Some confusing pairs can be resolved by exploiting other modalities such as RGB appearance. If the model successfully recognizes the keyboard, then it can distinguish typing from writing. Table 18 shows that, with multi-modality fusion in *RGBPose*-Conv3D, the recognition performance on those confusing pairs improves a lot.

4.7. Why skeleton-based pose estimation performs poorly on Kinetics400

PoseConv3D with high-quality 2D skeletons improves the Top-1 accuracy of skeleton-based action recognition on Kinetics400 from 38.0% to 47.7%. However, the accuracy on Kinetics400 is still far below the accuracies on other datasets. Besides the difficulties mentioned in Sec 1, two more problems will degrade the quality of extracted skeleton sequences (Figure 11): 1. Since Kinetics400 is not human-centric, human skeletons are missing or hard to recognize in many frames. 2. For the same reason, only small parts of humans appear in many frames, while the pose estimators are easy to fail in this scenario.

We also report the mean class accuracy on Kinetics-Motion [23] in Table 19, which contains 30 action classes in Kinetics that are strongly related to body motions. The accuracy of skeleton-based action recognition is much higher on this subset, increasing from 47.7% to 81.9%. When combined with the state-of-the-art RGB predictions, the improvement is much more significant, increasing from 0.6% to 2.0%. However, the skeleton-based performance is still far behind the state-of-the-art RGB-based action recognition method [13], which achieves 92.7% mean class accuracy on Kinetics-Motion. The inferior recognition performance indicates that there still needs more future work for skeleton-based action recognition in the wild.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1, 5

[2] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR*, pages 7024–7033, 2018. 5

[3] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, pages 670–688. Springer, 2020. 5

[4] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 2

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 2, 6

[6] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017. 5

[7] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, 2016. 2

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

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012. 2

[10] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 4

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[12] 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. *TPAMI*, 2019. 1

[13] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv:2106.13230*, 2021. 7

[14] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, 2020. 3, 6

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

[16] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019. 3, 5

[17] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, June 2016. 1, 5, 6

[18] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, pages 2616–2625, 2020. 1, 6

[19] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 2, 5

[20] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[21] Philippe Weinzaepfel, Romain Brégier, Hadrien Combaluzier, Vincent Leroy, and Grégory Rogez. Dope: Distillation of part experts for whole-body 3d pose estimation in the wild. In *ECCV*, 2020. 3, 5

[22] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *CVPR*, pages 7922–7931, 2019. 5

[23] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, volume 32, 2018. 7

[24] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 5