Skeleton2Point: Recognizing Skeleton-Based Actions As Point Clouds

Anonymous Authors

ABSTRACT

Skeleton-based action recognition has achieved remarkable results by developing graph convolutional networks (GCNs) and skeleton transformers. However, the existing methods pay much more attention to encoding joints' position with the given time and serial number information, neglecting to model the positional information contained in the 3D coordinate channel itself. To solve these problems, this paper proposes a skeleton-to-point network **(Skeleton2Point)** to model joints' position relationships in three-dimensional space, which is the first to leverage point cloud methods into skeleton-based action recognition in a dual-learner approach. The human skeleton learner feeds compact skeletal representations in the skeleton transformer network, which is composed of a spatial transformer block and a temporal transformer block. In the point cloud learner, skeleton data is transformed into point cloud's form with a proposed **I**nformation **T**ransform **M**odule (ITM), which fills the channel information with the spatial and temporal serial number. Then, several point cloud learning levels are adopted to extract deep position features. The point cloud learning level is made of three key layers: Sampling layer, Grouping layer, and Point cloud extract layer. We also propose a **C**luster-**D**ispatch-based **I**nteraction module (CDI) to enhance the discrimination of local-global information. In comparison with existing methods on NTU-RGB+D 60 and NTU-RGB+D 120 datasets, Skeleton2Point achieves SOTA levels on both joint modality and stream fusion. Especially, on the challenging NTU-RGB+D 120 dataset under the X-Sub and X-Set setting, the accuracies reach 90.63% and 91.86%. Please refer to the supplementary material for our code.

CCS CONCEPTS

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

KEYWORDS

Skeleton-based action recognition, point clouds, point cloud neural network

ACM MM, 2024, Melbourne, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

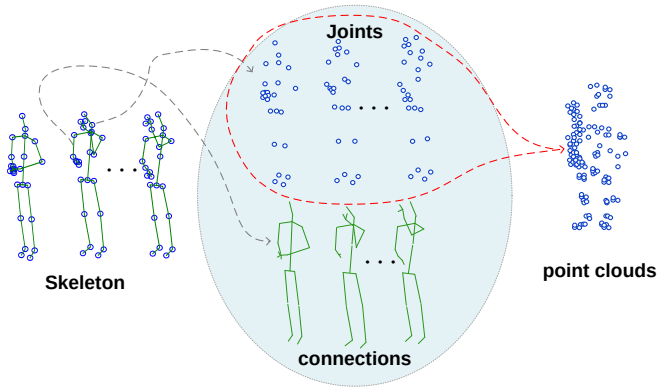

Figure 1: Skeleton data contains 3D joint coordinates and bone orientations. Both containing 3D coordinates, and joints could be regarded as point clouds. So we could leverage point cloud feature extraction method to learn the position information of the joint in a new view.

1 INTRODUCTION

Human action recognition is an important task in the field of computer vision, which also has great research value and broad application prospects in education[18], human-computer interaction[14], and content-based video retrieval[16]. Empowering intelligent machines with the same ability to understand human behaviors is critical for natural human-computer interaction and many other practical applications. For human action videos, various modalities derived from the rich multimedia are beneficial to the recognition task, including RGB, optical flow, and human skeletons. Among them, skeleton-based action recognition algorithms have attracted many researchers to explore due to their robustness against the variation of appearance and background. Skeleton data contain 3D human joint coordinates and their connection matrix, representing information about the joints' position and the connections between them. A typical way to use skeletons for action recognition is to build Graph Convolutional Networks (GCNs). Since [25] proposed STGCN to model skeletal data as a spatiotemporal graph structure, graph convolutional networks have developed rapidly. The joints and bones in the human body naturally form graphs, which makes GCNs a perfect tool for extracting topological features of skeletons. Recently, as Transformer [29] has gradually led in the performance and efficiency of image, natural language processing, and multi-modality, researchers have naturally begun to replace the classical GCN structure. With efficient structures like Positional Encoding (PE), the self-attention mechanism, and utilizing the multi-channel

adjacency matrix obtained by themselves, skeleton transformers have become the dominant method[15][40][30][2].

Containing 3D coordinates, skeleton joints can be naturally viewed as point clouds distributed in three-dimensional space as illustrated in Fig 1. However, while dealing with skeleton joint coordinates in skeleton data, all these methods pay much more attention to the topological relationships that exist between the joints, encoding joints' position with the given time and serial number information, neglecting to model the positional information contained in the 3D coordinates channel itself. In point cloud classification tasks, existing methods like[4][33][39] extract critical points and reduce point numbers by farthest point sampling and kNN algorithm, modeling the positional relationships between points step-by-step using 3D coordinate information, and then extract point cloud's latent feature with MLP or transformer layers.

To solve the above problem, we propose a skeleton-to-point network (Skeleton2Point) that consists of two trunk branches. In the first branch, referred to as the human skeleton branch, skeleton data is encoded with given space-time information and then fed into a graph transformer neural network to obtain predictions. In the second branch, regarded as the point cloud branch, skeleton data is transformed into point cloud's form with an information transform module. Next, FPS and kNN are used to sample and model the position relations between the points, then a point cloud information extractor is leveraged to extract latent features. We also propose a Cluster-Dispatch-based interaction module to enhance the discrimination of local-global information. The results from both branches are integrated to make final predictions. By leveraging the proposed Skeleton2Point, we effectively integrate skeleton information and position information to achieve better human action recognition.

Our main contributions can be summarized as follows:

- To our best knowledge, we are the first to regard skeleton joints as point clouds via incorporating the position information of skeletons into point cloud methods, demonstrating the validity of modeling position relationships with 3D coordinates.
- We devise a novel information transformation module (ITM) to merge the original time and series information and the joint coordinates information. We also propose a Cluster-Dispatch-based interaction module (CDI) to focus on overall movement trends.
- We conduct extensive experiments on NTU-RGB+D 60 and NTU-RGB+D 120 datasets to compare our proposed method with the state-of-the-art models in the joint stream and multi-streams. Experimental results demonstrate the significant improvement of our method. In the most challenging NTU120_XSub&XSet, our method achieves sota by a large margin.

2 RELATED WORKS

2.1 Skeleton Based Human Action Recognition

To tackle skeleton-based action recognition, early works treat it as a sequence classification task. [28] design an auto-encoder with RNNs to learn high-level features from the sequence. Another stream converts the skeleton sequence to image-like data using hand-crafted schemes [3, 41]. [6] concatenate an RGB frame with a 2D skeleton heat map and use 3D CNNs to extract features. These works do not explicitly exploit the spatial structure of the human body. Inspired by the development of Graph Convolutional Networks (GCN), Spatial-Temporal GCNs are used to extract high-level features from skeletons since the joints and bones in the human body naturally construct a graph. An early application of GCN on skeleton-based action recognition is STGCN [25], which uses stacked GCN blocks to process skeleton data, while each block consists of a spatial module and a temporal module. [38] proposed a channel-wise topology graph convolution, which models channel-wise topology with a refinement method.

Recently, transformers have also been explored in skeleton-based action recognition. [15] and [19]unified spatial and temporal modeling within the transformer via segment temporal aggregation and physical connectivity constraints in which way the topology of the human body is fully exploited. [30] is proposed with comprehensive high-performance spatiotemporal attention design and topological information fusion. Nevertheless, due to the lack of a module focusing on position information extracting, the methods above couldn't achieve excellent local-global positional relationship capture.

2.2 Multimodality Based Human Action Recognition

Commonly used modalities for multimodality-based action recognition include skeletons [21, 38], color images[42], text[36], human parsing [20] and depth images[32]. For instance, Wu et al.[32] use 3D CNN to effectively fuse and complement the skeleton and depth information for robust multimodal action recognition. Das et al. [24] propose the Video-Pose Network (VPN), which employs both CNN and GCN to model RGB and skeletal modalities, enabling the learning of enhanced spatiotemporal features. Liu et al. [20] utilize both CNN and GCN backbones to process human parsing and human pose modalities separately, which implement a late fusion strategy to combine features from both modalities. Shu et al.[26] propose a novel multimodal fusion network called ESE-FN to aggregate discriminative information of skeletons and color images for better action recognition. These multimodal-based methods take full advantage of the complementarity between modalities. However, unlike the methods mentioned above which utilize two completely different modalities, our proposed Skeleton2Point for the first time treats the skeleton modality as point clouds, which demonstrates the validity of modeling position relationships with 3D coordinates in skeleton-based action recognition.

2.3 Point Cloud Classification

The major method for processing the point clouds is point-based modeling. [4] is a pioneering work that successfully applies deep architecture on raw point sets, with shared multi-layer perceptrons (MLP) used. PointNet++[5] is built on top of PointNet[4], which learns hierarchical point cloud features and can aggregate features in local geometric neighbors. Recently, PointNeXt [22] explored more advanced training and data augmentation strategies with the PointNet++ backbone to further improve accuracy and efficiency. Point Transformer proposes a modified Transformer architecture that aggregates local features with vector attention and relative position encoding [45]. Following them, some works have extended

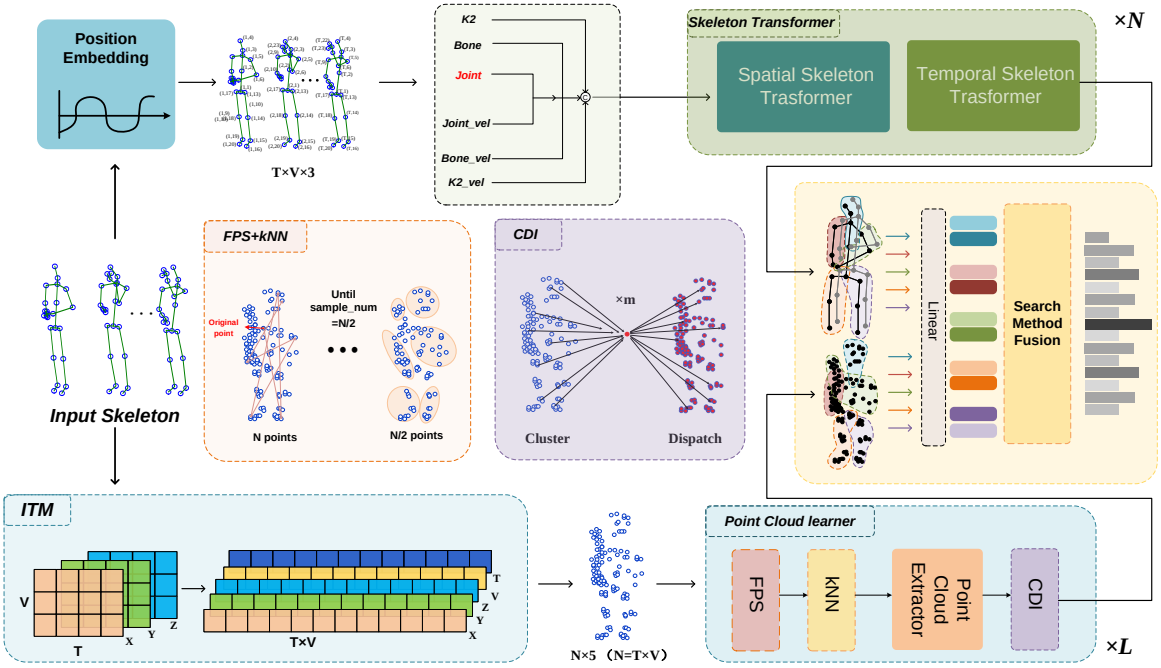

Figure 2: Framework of our proposed Skeleton2Point

the point-based methods to various local aggregation operators such as PointConT[39], achieved a sota of categorization tasks by combining various outstanding modules in point-based modeling. Point cloud models show excellent performance in learning 3D coordinate information, and the joint modality of skeleton data also consists of 3D coordinate information of each joint. So naturally, we think of using point cloud models to deep-dive into the position relationships of skeleton joints. However, there is no similar exploration in existing works, so we propose Skeleton2Point, which learns the skeleton joints' feature along with three-dimensional position information using point cloud methods.

3 METHOD

3.1 Human Skeleton Learning

Skeleton Data. The skeleton data is concise and robust to environmental noise, therefore our skeleton branch harnesses the human skeleton for action recognition. Conceptually the skeleton sequence is a natural topological graph, in which joints are graph vertices and bones are edges. The graph is denoted as $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$, where $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N\}$ is a set of N joints and E is a set of bones in the skeleton. For 3D skeleton data, the joint $\boldsymbol{v}_i$ is denoted as $\boldsymbol{x}_i$, $\boldsymbol{y}_i$, $\boldsymbol{z}_i$, where $\boldsymbol{x}_i$, $\boldsymbol{y}_i$, and$\boldsymbol{z}_i$ locate $\boldsymbol{v}_i$ in three-dimensional Euclidean space.

Here we define skeleton data as four different modalities, namely joint (J), bone (B), joint motion (JM), and bone motion (BM). Given two joints data $\boldsymbol{v}_i = \{\boldsymbol{x}_i, \boldsymbol{y}_i, z_i\}$ and $\boldsymbol{v}_j = \{\boldsymbol{x}_j, \boldsymbol{y}_j, z_j\}$, a bone data of the skeleton is defined as a vector $\boldsymbol{e}_{\boldsymbol{v}_i,\boldsymbol{v}_j} = (\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{y}_i - \boldsymbol{y}_j, z_i - z_j)$.

Given two joints data vti, v(t+1)i from two consecutive frames, the data of joint motion is defined as $\boldsymbol{m}_{ti} = \boldsymbol{v}_{(t+1)i} - \boldsymbol{v}_{ti}$i. Similarly, given two bones data $\boldsymbol{e}_{\boldsymbol{v}_{(t+1)i},\boldsymbol{v}_{(t+1)j}}$, $\boldsymbol{e}_{\boldsymbol{v}_{ti},\boldsymbol{v}_{tj}}$ from two consecutive frames, the data of bone motion is defined as $\boldsymbol{m}_{\boldsymbol{v}_{ti},\boldsymbol{v}_{tj}} = \boldsymbol{e}_{\boldsymbol{v}_{(t+1)i},\boldsymbol{v}_{(t+1)j}} - \boldsymbol{e}_{\boldsymbol{v}_{ti},\boldsymbol{v}_{tj}}$.

Backbone. Transformer-based methods have achieved success in skeleton-based action recognition due to their unique advantages in modeling joint relations. Our Skeleton2Point also embraces graph transformer as the backbone to model skeleton features. The input of our model is a sequence of skeletons with a shape of $TV3$, which means T frames of V joints in a 3D space. We build our approach on [30]. The backbone consists of 10 basic graph transformers. A graph transformer is typically composed of a spatial transformer and a temporal transformer. The normal spatial transformer utilizes the attention matrix A_t and adjacency matrix A_i for aggregate features of neighbor vertices to update the features f_i, which can be expressed as follows:

$$f_i^{\text{out}} = f_i^{\text{in}} + V_i A, \tag{1}$$

$$V_i = \text{Conv}_{1\times1}\left(\text{split}_n\left(\text{trans}_v\left(f_i^{\text{in}}\right)\right)\right). \tag{2}$$

where A can be defined as the attention matrix A_t, static (defined manually) adjacency matrix A_I and dynamic adjacency matrix A_d (initialized manually but learnable). The Eq 2 denotes the multi-head mechanism in self-attention that our model used and the attention matrix A_t is calculated as follows:

$$Q, K = \sigma\left(\text{linear}\left(\text{pool}\left(\text{split}\left(f\right)\right)\right)\right), \tag{3}$$

$$A_t = \text{softmax}\left(\text{atten}\left(Q, K\right)\right). \tag{4}$$

Our Skeleton2Point feeds the joint modality skeleton into ten dynamic graph transformer blocks for feature extraction. A Retrospect Model with Multi-stream strategy [30] is specifically used for residual information to extract key information twice for the final classification. The Retrospect Model adopts an adaptive pyramid structure to pass the shallow features back to the final layer, significantly alleviating the key information loss problem due to the small number of joints in the network iteration process.

3.2 Point Cloud Learning

From Skeleton to Point Clouds. In the point cloud branch, we propose the information transform module (ITM) to transform the skeleton data into point cloud's form. Given an input skeleton $\mathbf{S} \in \mathbb{R}^{3\times T\times V}$, we begin by filling the channel information with the spatial and temporal serial numbers of each joint $\mathbf{S}_{i,j}$, where each joint's serial number is presented as $\left[\frac{i}{T} - 0.5, \frac{j}{V} - 0.5\right]$. The improved skeleton is then converted to a collection of point clouds $\mathbf{P} \in \mathbb{R}^{5\times N}$, where $n = T \times V$ is the number of points, and each point contains both position (coordinates) and spatial-temporal serial information.

Point Cloud Set Feature Learning. After modality transformation, several point cloud learning levels are adopted to extract deep position features. The point cloud learning level is made of three key layers: Sampling layer, Grouping layer, and Point cloud extract layer. The Sampling layer selects a set of points from input points, which defines the centroids of local regions. The grouping layer then constructs local region sets by finding "neighboring" points around the centroids. The point cloud extract layer uses [39] as lite-backbone and [33] as heavy-backbone to encode local region patterns into feature vectors.

A point cloud learning level takes an $N \times (d + C)$ matrix as input that is from N points with d-dim coordinates and C-dim point feature. It outputs an $N' \times (d' + C)$ matrix of N' subsampled points with d-dim coordinates and new C'-dim feature vectors summarizing local context. We introduce the layers of a point cloud learning level in the following paragraphs.

- **Sampling layer.** Given input points $\{x_1, x_2, \ldots, x_n\}$,we use iterative farthest point sampling (FPS) to choose a subset of points $\left\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\right\}$, such that x_{i_j} is the most distant point with $\left\{x_{i_1}, x_{i_2}, \ldots, x_{i_j-1}\right\}$regard to the rest points. Compared with random sampling, it has better coverage of the entire point set given the same number of centroids.

- **Grouping layer.** The input to this layer is a point set of size $N \times (d + C)$ and the coordinates of a set of centroids of size $N' \times d$. The output is groups of point sets of size $N' \times K \times (d + C)$, where each group corresponds to a local region and K is the number of points in the neighborhood of centroid points. Note that K varies across groups but the succeeding Point cloud extract layer can convert a flexible number of points into a fixed-length local region feature vector and we set the number K to 24.

  Ball query finds all points that are within a radius of the query point (an upper limit of K is set in implementation). An alternative range query is the K nearest neighbor (kNN) search which finds a fixed number of neighboring points. Compared with kNN, ball query's local neighborhood guarantees a fixed region scale thus making local region features more generalizable across space, which is preferred for tasks requiring local pattern recognition. Ablation experiments are in Table 5. compare different sampling and grouping strategies.

- **Point cloud extract layer.** In this layer, the input are N' local regions of points with data size $N' \times K \times (d + C)$. Each local region in the output is abstracted by its centroid and local feature that encodes the centroid's neighborhood. Output data size is $N' \times (d + C')$. We choose PointConT[39] as the lite building block and PointMLP[33] as the heavy building block for local pattern extracting. By using relative coordinates together with point features we can capture point-to-point relations in the local region. It is noted that the detailed implementation of the block is not the main concern of our method. The implementation of the point cloud extract unit can be replaced by any other point cloud network module.
  In the lite block[39], there are two branches: the high-frequency aggregation branch and the low-frequency aggregation branch. The high-frequency branch can be defined as

$$f_h = \text{ResMLP}\left(\text{MaxPool}\left(f_g\right)\right), \quad f_h \in \mathbb{R} \tag{5}$$

  where MaxPool and ResMLP denote max-pooling operation and residual MLP block, respectively. The low-frequency branch simply utilizes an average pooling layer (AvgPool) before the Transformer, and this design allows the Transformer to focus on embedding low-frequency information. This branch can be defined as:

$$f_l = \text{Trans}\left(\text{AvgPool}\left(f_g\right)\right), \quad f_l \in \mathbb{R} \tag{6}$$

  In the end, we concatenate the features from the high-frequency aggregation branch and the low-frequency aggregation branch and then feed them to an MLP block as the Inception aggregator output features f'.

$$f' = \text{MLP}\left(\|f_h, f_l\|\right), \quad f' \in \mathbb{R} \tag{7}$$

  In the heavy block[39], the key operation in one stage can be formulated as:

$$g_i = \Phi_{\text{pos}}\left(\mathcal{A}\left(\Phi_{\text{pre}}\ \left(f_{i,j}\right), |\ j = 1, \cdots, K\right)\right), \tag{8}$$

  where Φ_{pre} and Φ_{pos} are residual point MLP blocks: the shared Φ_{pre} is designed to learn shared weights from a local region while the Φ_{pos} is leveraged to extract deep aggregated features. MLP is a small network composed of a Fully-connected(FC) layer, Batch Normalization layer, and activation function. In detail, the mapping function can be written as a series of homogeneous residual MLP blocks, $MLP(x) + x$, in which MLP is combined by FC, normalization, and activation layers (repeated two times).

The framework of PointMLP is succinct for extracting point clouds transformed from human skeleton joints, it exhibits some prominent merits. 1) Since PointMLP only leverages MLPs, it is naturally invariant to permutation, which perfectly fits the characteristic of point clouds. 2) By incorporating residual connections, PointMLP can be easily extended to dozens of layers, resulting in deep feature representations. 3) In addition, since there are no sophisticated extractors included and the main operation is only highly optimized feed-forward MLPs, even if we introduce more layers, our PointMLP still performs efficiently.

Cluster-Dispatch based interaction module. Inspired by the [34], we also propose a Cluster-Dispatch-based interaction module (CDI) to make the point cloud backbone focus on the overall movement trends. Suppose an action sample contains m points and the center is c, all m points in the sample are aggregated by global averaging to get the center point. Assuming the similarity between the m points and the center is $s \in \mathbb{R}^m$, we obtain $p_v \in \mathbb{R}^{m \times d'}$ by mapping these m points to the value space, where d' is the value dimension. Similarly, there is a clustering center c_v in the value space and the clustering feature $f \in \mathbb{R}^{d'}$ can be written as:

$$f = \frac{1}{\gamma}\left(\sigma\left(\sum_{i=1}^{m} \alpha s_i + \beta\right) \cdot p_v + c_v\right), \quad \text{s.t.}, \gamma = 1 + \sum_{i=1}^{m} \sigma\left(\alpha s_i + \beta\right). \tag{9}$$

Here α and β are learnable scalars to scale and shift the similarity. $\sigma()$ is a sigmoid function to re-scale the similarity to (0, 1) and s_i denotes the similarity between the i-th point and the center. To dispatch the feature, the aggregated features f are adaptively assigned to each point in the action sample, allowing the points to communicate with each other and share features from all points by:

$$p_i' = p_i + \mathrm{FC}(\mathrm{sig}(\alpha s_i + \beta) \cdot f). \tag{10}$$

In the original space, p_i denotes the i-th point and p_i' denotes the point after reallocation. Ablation experiments in Table 3. demonstrate the effectiveness of the proposed modules.

3.3 Gaussian Search Method based Fusion

We validate the model under the joint modality only and 6-stream fusion (6s), respectively. The joint modality only is used because joints contain 3D coordinates same as point clouds, so we want to verify that the deep position feature that point cloud learning had extracted would help the skeleton backbone learn the joint position complementarily. Following prior work [7], the input of multiple streams refers to $\tilde{\mathcal{X}}_k = \left(I - P^k\right)\mathcal{X}$, where $k = 1, 2, \ldots, K$, and K is set to 2 in NTU-RGB+D 60 dataset.

In a fusion of skeleton and point clouds, fixed parameters limit the performance of each model and make it difficult to achieve optimal results. We follow the weight search algorithm[37] based on Gaussian Process Bayesian Optimization. This algorithm effectively finds the best solution by constructing a probabilistic model of the objective function. Given an objective function $f(x)$ and initial samples $\mathcal{P} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i represents the input and $y_i = f(x_i)$ represents the observed output. The goal is to find the global optimum x of $f(x)$ within the search space $\mathcal{X}$. The iteration continues until convergence, determined by a predefined number of iterations or convergence criteria. Ablation experiments are in Table 5. compare the different fusion results between fixed parameters and our search method. Please refer to our code for details.

Table 1: Ablation studies of point cloud branch on the NTU-RGB+D 60 and NTU-RGB+D 120 datasets with the joint input modality in point cloud branch.

Point Extractor	ITM	CDI (parallel)	CDI (cascade)	NTU-RGB+D 60 60C-Sub	NTU-RGB+D 60 120C-Sub
Lite	✗	✗	✗	81,22	69.35
Lite	✓	✗	✗	84.11	79.14
Lite	✓	✓	✗	84,62	79.67
Lite	✓	✗	✓	**84.75**	**80.17**

Table 2: Ablation studies of point cloud branch on the NTU-RGB+D 60 dataset with the joint input modality in point cloud branch.

Point Extractor	ITM	CDI	NTU-RGB+D 60 X-Sub
Heavy	✗	✗	81.39
Heavy	✓	✗	88.17
Heavy	✓	✓	**88.56**

4 EXPERIMENTS

4.1 Datasets

NTU-RGB+D 60 is a large-scale human action recognition dataset collected in an indoor environment, containing 56,880 skeleton action sequences. The action samples are performed by 40 volunteers and categorized into 60 classes. Each sample contains an action and is guaranteed to have at most 2 subjects, which are captured by three Microsoft Kinect v2 cameras from different views concurrently. The authors of this dataset recommend two benchmarks: (1) cross-subject (Xsub): training data comes from 20 subjects, and testing data comes from the other 20 subjects. (2) cross-view (X-view): training data comes from camera views 2 and 3, and testing data comes from camera view 1.

NTU-RGB+D 120 is currently the largest dataset with 3D joint annotations for human action recognition, which extends the NTU-RGB+D 60 dataset with an additional 57,367 skeleton sequences over 60 extra action classes. Totally 113,945 samples over 120 classes were performed by 106 volunteers, captured with three camera views. This dataset contains 32 setups, each denoting a specific location and background. The authors of this dataset recommend two benchmarks: (1) cross-subject (X-sub): 53 of the 106 subjects' actions are used for training, and the remaining 53 are used for validation. (2) cross-setup (X-setup): Of the 32 setups, data with even setup IDs are used for training, and the remaining data with odd IDs are used for validation.

Table 3: Recognition accuracy comparison against state-of-the-art methods on NTU-RGB+D 60 and NTU-RGB+D 120 datasets under the joint and multi-stream modality. Bold text denotes optimal performance, dagger marks † indicate second-best.

Type	Methods	Publisher	NTU-RGB+D 60 Joint		NTU-RGB+D 120 Joint		NTU-RGB+D 60		NTU-RGB+D 120	
			X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)	X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)
GCN	Shift-GCN[13]	*CVPR'20*	87.80	95.10	80.90	83.20	90.70	96.50	85.90	87.60
	MS-G3D[44]	*CVPR'20*	88.77	94.88	82.35	84.14	91.50	96.20	86.90	88.40
	CTR-GCN[38]	*ICCV'21*	88.95	90.23	84.95	86.68	92.40	96.80	88.90	90.60
	MSTGCN[43]	*AAAI'21*	89.00	95.10	82.80	84.50	92.3	91.5	87.5	88.8
	EfficientGCN[27]	*TPAMI'22*	-	-	-	-	92.10	96.10	88.70	88.90
	Info-GCN[7]	*CVPR'22*	89.80	95.20	85.10	86.30	93.00	97.10	89.80	91.20
	DD-GCN[1]	*ICME'23*	90.50	**96.90**	86.10	87.60	92.6	96.9	88.9	90.2
	HD-GCN[12]	*ICCV'23*	90.60	95.70	85.70	87.30	93.4†	97.2	90.1	91.6
	FR-Head[9]	*CVPR'23*	90.33	95.26	85.51	87.32	92.8	96.8	89.5	90.9
	Sym-CNN[17]	*TPAMI'22*	-	-	-	-	90.1	96.4	-	-
Hypergraph	Hyper-GNN[10]	*TIP'21*	-	-	-	-	89.5	95.7	-	-
	DHGNN[31]	*CoRR'21*	-	-	-	-	90.7	96.0	86.0	87.9
	Selective-HCN[46]	*ICMR'21*	-	-	-	-	90.8	96.6	-	-
	SD-HGCN[11]	*ICONIP'21*	-	-	-	-	90.9	96.7	87.0	88.2
Transformer	HyperFormer[40]	*arXiv'22*	90.70	95.10	86.60†		92.9	96.5	89.9	91.3
	STF[15]	*AAAI'22*	91.34†	96.46	85.06	86.40	92.47	96.86	88.85	89.92
	ST&ST[2]	*ACMMM'23*	90.90	95.40	85.80	87.90	93.1	96.7	89.8	91.2
	TranSkeleton[19]	*TCSVT'23*	-	-	-	-	92.8	97.0	89.4	90.5
	Baseline	*2023*	90.73	95.76	85.82	87.64	93.20	97.20	90.20	91.50
	Skeleton2Point(Lite)	*2024*	91.50	96.28	88.62	89.92	93.32	97.35†	90.50†	91.76†
	Skeleton2Point(Heavy)	*2024*	**92.07**	96.70†	**88.93**	**90.34**	**93.41**	**97.49**	**90.63**	**91.86**

Table 4: Comparison of parameter, computation cost when training&inferring and accuracy on the NTU-RGB+D 60 and NTU-RGB+D 120 cross-subject Protocols.

Methods	Publisher	Modality	Param	Flops	X-Sub	X-Set
DRDIS[8]	*TCSVT'21*	Ske+RGB	58.49M	18.67G	91.10	81.30
VPN[24]	*ECCV'20*	Ske+RGB	24.00M	-	93.50	86.30
MMNet[35]	*TPAMI'22*	Ske+RGB	14.40M	19.2G	86.60	88.70
IPPNet[23]	*arxiv'23*	Ske+RGB	25.27M	7.84G	85.00	86.70
Skeleton2Point(Lite)	*2024*	Ske+PC	16.98M	5.78G	**93.32**	**90.20**
Skeleton2Point(Heavy)	*2024*	Ske+PC	25.87M	10.27G	**93.41**	**90.63**

Table 5: Comparison of different combinations of α and β when ensembling skeleton learner and lite point cloud learner[39] in joint modality.

$[\alpha, \beta]$	XS60(%)	XV60(%)	X-Sub120(%)	X-Set120(%)
$[0.9, 0.1]$	91.09	96.03	86.57	88.33
$[0.85, 0.15]$	91.29	96.12	86.86	88.60
$[0.8, 0.2]$	91.43	96.19	87.03	88.77
$[0.75, 0.25]$	91.46	96.22	87.06	88.83
$[0.7, 0.3]$	91.31	96.18	87.09	88.81
$[0.6, 0.4]$	90.97	95.96	86.87	88.59
Search Method	**91.50**	**96.28**	**87.14**	**88.91**

Table 6: Comparison of different sampling and grouping methods in NTU-RGB+D 60 under the X-Sub setting.

Sampling	Grouping	Acc(%)
Random sample	kNN	82.43
FPS	Ball(R=0.1)	80.12
FPS	Ball(R=0.2)	81.94
FPS	kNN	**84.75**

4.2 Implementation Details

We implement the proposed method with the PyTorch deep learning framework. All experiments are conducted on 8 GeForce RTX 3070 GPUs. We follow previous work [38] to process the two datasets and all skeleton sequences are padded to 64 frames. We used Stochastic Gradient Descent (SGD) as the optimizer and cross-entropy as the loss function. On the skeleton branch, we adopt [30] as the human skeleton backbone. In the first 5 epochs, we apply a warmup strategy for stable training. The initial learning rate is set to 0.1 and we decrease it at epoch 35 and 55 with a factor of 0.1. We train all models with 90 epochs and select the best performance with a batch size of 128. On the point cloud branch, we adopt [39] as the lite point cloud information extractor and [30] as the heavy point cloud information extractor. The training epoch and learning rate are set to 250 and 0.01 respectively, while the batch size is also set to 128. In the skeleton branch, the base channel C is set to 80, and the hidden channel C_h is set to 320. In the point cloud branch, the base channel C is set to 64, and the hidden channel C_h is set to 1024. Please refer to our code for more details.

4.3 Ablation Study

Ablation experiments of different components. We experimentally validate the importance of the components in the point cloud branch, where CDI (parallel) denotes the CDI module is paralleled with the MLP in point cloud information extractor and CDI (cascade) denotes the CDI module is cascaded with the MLP in point cloud information extractor. Table 2 shows that the branch without ITM has a performance drop of 2.9% and 9.8% on the NTU-RGB+D 60 dataset under the X-Sub setting and NTU-RGB+D 120 dataset under the X-Sub setting respectively. The result drops by 0.64% without using the CDI, which shows that the position modeling needs to interact with global-local information and focus on overall movement trends. In addition, the result of using cascade CDI outcompeting using parallel CDI by a margin of 0.3% reflects the

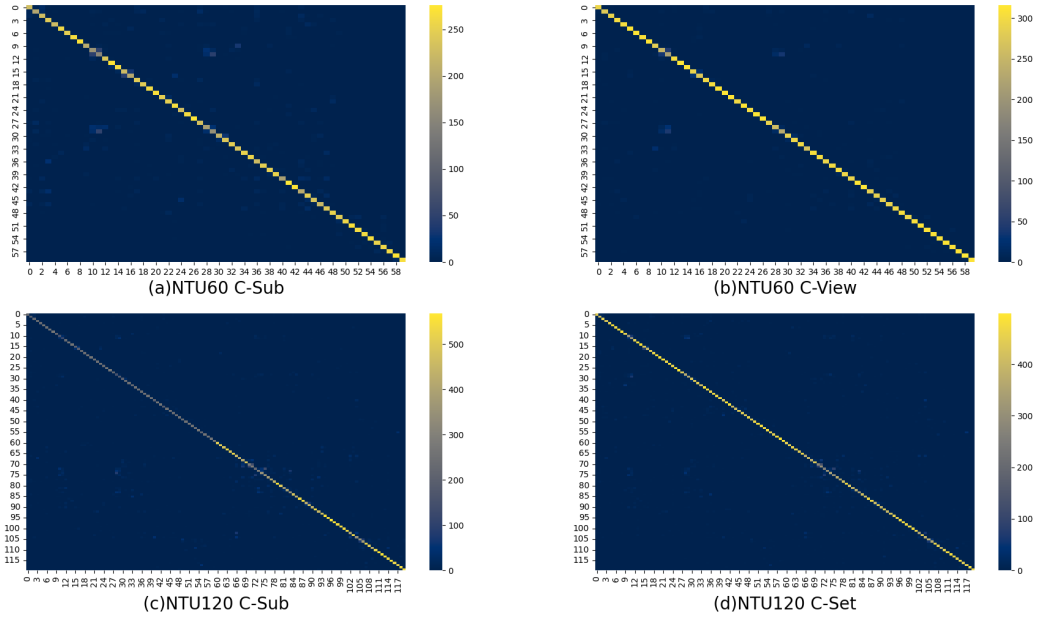

Figure 3: The confusion matrix of NTU-RGB+D 60 and NTU-RGB+D 120 datasets. The more yellow squares on the diagonal, the more accurate the recognition. (a) NTU-RGB+D 60 dataset on the benchmark of X-Sub. (b) NTU-RGB+D 60 dataset on the benchmark of X-View. (c) NTU-RGB+D 120 dataset on the benchmark of X-Sub. (d) NTU-RGB+D 120 dataset on the benchmark of X-Set.

cascade structure's advantage and is more beneficial for supporting position modeling with point cloud methods.

Parameter and computation cost. Additionally, we showcase the parameter and computation cost required for the proposed Skeleton2Point in Table 4. Compared with other multi-branch methods using different modalities like RGB, our Skeleton2Point achieves better performance and lower cost using skeleton and point clouds, demonstrating the validity and efficiency of modeling position relationships with point cloud methods in skeleton-based action recognition.

Ablation Study on the Hyper-parameters. To compare with the weight search algorithm, we analyze the configurations on the hyper-parameters of our method, and the results are available in Table 5. We try many combinations of α and β to balance the importance of the skeleton branch and point cloud branch separately. From the results, we can observe that a bigger percentage of the point cloud branch may hurt the performance while too small values only provide a little improvement. It is concluded that the refinement of the semantic features from the skeleton branch plays a major role and the three-dimension position features provide the auxiliary effects.

Ablation Study on Sampling and Grouping methods. To validate that the sampling and grouping methods (FPS+kNN) are suitable for the skeleton action recognition tasks, we also evaluate different sampling methods like random sample and grouping methods like ball query on NTU-RGB+D 60 dataset under the X-Sub setting with the joint input modality in Table 6.

4.4 Comparison with Related Methods

Our proposed Skeleton2Point method is compared with the state-of-the-art methods on two different datasets: NTU-RGB+D 60 and NTU-RGB+D 120 datasets with the joint input modality to verify the competitive performance. Especially, only the joint stream dataset is used for a fair comparison because joints contain 3D coordinates the same as point clouds. If the bone or motion stream datasets were used, point cloud methods couldn't be well-compatible. Both the best results of all the methods shown in Table 4 and Table 5 are reported in the original papers on the joint stream dataset. The quantitative results are displayed in Table 3. The comparison methods include GCN-based and transformer-based methods. Popular GCN-based methods (the upper 8 solutions in Table 3) improve ST-GCN by constructing skeleton graphs with dynamic topologies to increase the receptive field of graph convolution. Recently proposed transformer-based methods (the last 3 solutions in Table 3) treat each joint of the human skeleton as a token and use spatial and

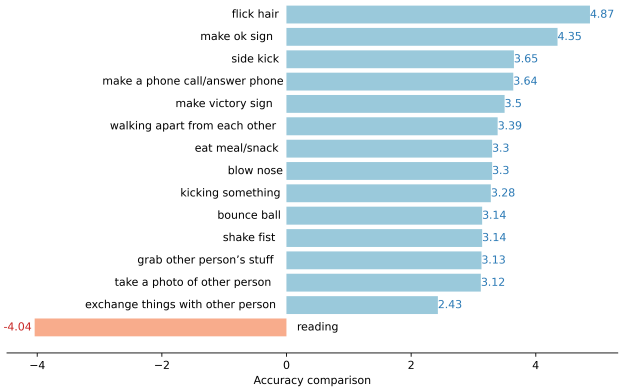

Figure 4: Visualization of the top-15 actions with the highest change when the human skeleton branch is integrated with the point cloud branch for NTU-RGB+D 120 dataset under the X-Sub setting with the joint input modality.

temporal self-attention operators to capture feature dependencies. Using a much more compact branch to model three-dimensional position relation and transformer-based network, our Skeleton2Point achieves comparable performance with the existing methods. Especially, in the most challenging NTU-RGB+D 120 under the X-Sub setting, Skeleton2Point has reached state-of-the-art accuracy, which is extremely promising.

4.5 Visualization

Top-15 change analysis. To better illustrate the fusion effect of the human skeleton branch and point cloud branch, we visualize the top-15 actions with the highest change when the human skeleton branch is integrated with the point cloud branch on the most challenging benchmark, NTU-RGB+D 120 dataset under the X-Sub setting in Fig 4. SkeletonPoint(Ours) shows a substantial improvement in accuracy over the Skeleton branch baseline for all actions except the "reading" action. This observation proves that the introduction of point cloud information as an aid is beneficial for the model to further learn human actions. Secondly, point cloud information performs better on actions with larger motion amplitudes, where the accuracy of actions with more significant motions such as "flick hair" and "make ok sign" improves significantly, while the accuracy of actions with smaller motions such as "reading" decreases. An important reason for this change is that the point cloud information is sampled with FPS, making it easier for the point cloud model to capture the details of the motion and the overall trajectory. For motions with larger motion amplitude that have more dispersed feature points in the point cloud data, FPS covers a wider range of motion trajectories, thus improving the accuracy for motions with larger motion amplitude; In contrast, for actions with smaller motion amplitude, the point cloud density is relatively low, resulting in relatively fewer feature points sampled, which may then be overlooked.

Confusion matrices analysis. Fig 3 shows the confusion matrix of our Skeleton2Point on NTU-RGB+D 60 and NTU-RGB+D 120 datasets. In the confusion matrix of the NTU-RGB+D 120 dataset under the X-Sub setting, a total of 62 samples achieved an accuracy exceeding 95%, accounting for approximately 51.67% of the total samples. Meanwhile, there are 98 action samples with an accuracy exceeding 85%, accounting for approximately 81.67%. Among 120 action samples, the 'staggering', 'jump up', 'arm circles', 'take off jacket', 'walking towards' and 'cheers and drink' action samples have the highest recognition accuracy, reaching 100%, while the 'staple book' action sample has the lowest recognition accuracy, just reaching 44.66%.

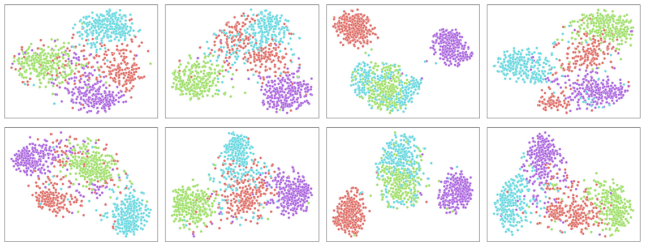

Figure 5: Visualization of latent representation by t-SNE for ambiguous groups from NTU-RGB+D 120 dataset. Different colors indicate different classes. The upper one is from the backbone[30], while the bottom one is from our Skeleton2Point.

Ambiguous groups analysis. In addition, in order to better demonstrate the modeling ability and effectiveness of our Sidea to ragard skeleton as point clouds on skeleton data, we randomly pick some ambiguous groups and visualize the distribution of them in the feature space using tSNE, which is shown in Fig 5. Each ambiguous group contains four classes, including an anchor class and three ambiguous classes. We compare our method with the backbone[33]. From Fig 5 we can see that our model obtains a different and more discriminative representation resulting in a compact clustering. We also observed that the skeleton features of the much different categories are far apart in the feature space. Meanwhile, action samples of different categories but with similarities are more close in the feature space.

5 CONCLUSIONS

This paper proposes Skeleton2Point, a new representation learning framework for improved skeleton-based action recognition, which explores the learning mechanism of joints' position information by regarding joints as point clouds. Different from previous methods that neglect to model the positional information, our Skeleton2Point is the first to leverage point cloud methods into skeleton-based action recognition in a dual-branch approach. The method simultaneously demonstrates the validity of modeling position relationships with 3D coordinates in skeleton-based action recognition. Finally, the efficacy of our Skeleton2Point is well validated by thorough experiments on the NTU-RGB+D 60 and NTU-RGB+D 120 datasets, where Skeleton2Point outperforms most existing methods.

Discussion. Despite the performance of our proposed Skeleton2Point on the NTU-RGD+D datasets, how to extract temporal information better after the skeleton is transformed into point clouds remains to be explored. We will concentrate on it in our future work. In addition, there are some potential negative societal impacts to be considered, like applying our module will introduce extra training costs, which should be thought in the carbon emission problem.

REFERENCES

[1] Li C., Huang Q., and Mao Y. [n. d.]. DD-GCN: directed diffusion graph convolutional network for skeleton-based human action recognition. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*.

[2] Zhang C., Hu Y., L M., Yangand C., and Hu X. 2023. Skeletal Spatial-Temporal Semantics Guided Homogeneous-Heterogeneous Multimodal Network for Action Recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3657–3666.

[3] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. 2018. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7024–7033.

[4] Qi C.R., Su H., Mo K., and Guibas L.J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[5] Qi C.R., Rui Z., Yi L., Su H., and Guibas L.J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).

[6] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. 2022. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2969–2978.

[7] Chi H., Han M., Chi S., Lee S., Huang Q., and Ramani K. 2022. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20186–20196.

[8] Wu H., Ma X., and Li Y. 2021. Spatiotemporal multimodal learning with 3D CNNs for video action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 3 (2021), 1250–1261.

[9] Zhou H., Liu Q., and Wang Y. 2023. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[10] Xiaoke Hao, Jie Li, Yingchun Guo, Tao Jiang, and Ming Yu. 2021. Hypergraph neural network for skeleton-based action recognition. *IEEE Transactions on image processing* 30 (2021), 2263–2275.

[11] Changxiang He, Chen Xiao, Shuting Liu, Xiaofei Qin, Ying Zhao, and Xuedian Zhang. 2021. Single-skeleton and dual-skeleton hypergraph convolution neural networks for skeleton-based action recognition. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II 28*. Springer, 15–27.

[12] Lee J., Lee M., Lee D., and Lee S. 2023. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10444–10453.

[13] Cheng K., Zhang Y., He X., Chen W., Cheng J., and Lu H. 2020. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 183–192.

[14] Hema S Koppula and Ashutosh Saxena. 2015. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence* 38, 1 (2015), 14–29.

[15] Ke L., Peng K., and Lyu S. 2022. Towards to-at spatio-temporal focus for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 1131–1139.

[16] Chao Li, Zi Huang, Yang Yang, Jiewei Cao, Xiaoshuai Sun, and Heng Tao Shen. 2017. Hierarchical latent concept discovery for video event detection. *IEEE Transactions on Image Processing* 26, 5 (2017), 2149–2162.

[17] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE transactions on pattern analysis and machine intelligence* 44, 6 (2021), 3316–3333.

[18] Min Li, Zhenjiang Miao, Xiao-Ping Zhang, Wanru Xu, Cong Ma, and Ningwei Xie. 2021. Rhythm-aware sequence-to-sequence learning for labanotation generation with gesture-sensitive graph convolutional encoding. *IEEE Transactions on Multimedia* 24 (2021), 1488–1502.

[19] Haowei Liu, Yongcheng Liu, Yuxin Chen, Chunfeng Yuan, Bing Li, and Weiming Hu. 2023. TranSkeleton: Hierarchical Spatial-Temporal Transformer for Skeleton-Based Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2023).

[20] Jinfu Liu, Runwei Ding, Yuhang Wen, Nan Dai, Fanyang Meng, Shen Zhao, and Mengyuan Liu. 2024. Explore Human Parsing Modality for Action Recognition. *CAAI Transactions on Intelligence Technology (CAAI TIT)* (2024).

[21] Jinfu Liu, Xinshun Wang, Can Wang, Yuan Gao, and Mengyuan Liu. 2023. Temporal Decoupling Graph Convolutional Network for Skeleton-based Gesture Recognition. *IEEE Transactions on Multimedia* 26 (2023), 811–823.

[22] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems* 35 (2022), 23192–23204.

[23] Ding R., Wen Y., Liu J., Dai N., Meng F., and Liu M. 2023. Integrating Human Parsing and Pose Network for Human Action Recognition. *arXiv preprint arXiv:2307.07977* (2023).

[24] Das S., Dai S., Sharmaand R., Bremond F., and Thonnat M. [n. d.]. Vpn: Learning video-pose embedding for activities of daily living. In *16th European Conference , 2020, Proceedings, Part IX 16*. 72–90.

[25] Yan S., Xiong Y., and Lin D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[26] Xiangbo Shu, Jiawen Yang, Rui Yan, and Yan Song. 2022. Expansion-Squeeze-Excitation Fusion Network for Elderly Activity Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 8 (2022), 5281–5292.

[27] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. 2023. Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2023), 1474–1488. https://doi.org/10.1109/TPAMI.2022.3157033

[28] Kun Su, Xiulong Liu, and Eli Shlizerman. 2020. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9631–9640.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[30] Xin W., Miao Q., Liu Y., Liu R., Pun C., and Shi C. 2023. Skeleton MixFormer: Multivariate Topology Representation for Skeleton-based Action Recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*. 2211–2220.

[31] Jinfeng Wei, Yunxin Wang, Mengli Guo, Pei Lv, Xiaoshan Yang, and Mingliang Xu. 2021. Dynamic hypergraph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2112.10570* (2021).

[32] Hanbo Wu, Xin Ma, and Yibin Li. 2022. Spatiotemporal Multimodal Learning With 3D CNNs for Video Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 3 (2022), 1250–1261.

[33] Ma X., Qin C., You H., Ran H., and Fu Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123* (2022).

[34] Ma X., Zhou Y., Wang H., Qin C., Sun B., Liu C., and Fu Y. 2023. Image as Set of Points. *arXiv preprint arXiv:2303.01494* (2023).

[35] Bruce XB., Liu Y., Zhang X., Zhong S., and K.CC. Chan. 2022. Mmnet: A model-based multimodal network for human action recognition in rgb-d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 3522–3538.

[36] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. 2023. Generative Action Description Prompts for Skeleton-based Action Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 10276–10285.

[37] Wentian Xin, Yi Liu, Ruyi Liu, Qiguang Miao, Cheng Shi, and Chi-Man Pun. 2023. Auto-Learning-GCN: An Ingenious Framework for Skeleton-Based Action Recognition. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 29–42.

[38] Chen Y., Zhang Z., Yuan C., Li B., Deng Y., and Hu W. 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 13359–13368.

[39] Liu Y., Tian B., Lv Y., Li L., and Wang F. 2023. Point cloud classification using content-based transformer via clustering in feature space. *IEEE/CAA Journal of Automatica Sinica* (2023).

[40] Zhou Y., Cheng Z., Li C., Fang Y., Geng Y., Xie X., and Keuper M. 2022. Hypergraph transformer for skeleton-based action recognition. *arXiv preprint arXiv:2211.09590* (2022).

[41] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. 2019. PA3D: Pose-action 3D machine for video recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 7922–7931.

[42] Bruce X.B. Yu, Yan Liu, Xiang Zhang, Sheng-hua Zhong, and Keith C.C. Chan. 2023. MMNet: A Model-Based Multimodal Network for Human Action Recognition in RGB-D Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2023), 3522–3538.

[43] Chen Z., Li S., Yang B., Li Q., and Liu H. 2021. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*.

[44] Liu Z., Zhang H., Chen Z., Wang Z., and Ouyang W. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 143–152.

[45] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 16259–16268.

[46] Yiran Zhu, Guangji Huang, Xing Xu, Yanli Ji, and Fumin Shen. 2022. Selective hypergraph convolutional networks for skeleton-based action recognition. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*. 518–526.