Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MGDG-HMPN: Motion-Guided Dynamic Graph Topology and Hierarchical Motion Propagation Network for Skeleton-Based Action Recognition

Tahir Hussain
Central South University
Wang Bin
wb_csut@csu.edu.cn

Central South University
Jinfang Sheng
Central South University
Saeeda Saeed
Central South University
Ikramov Ilham
Central South University
Wajid Ali
Central South University
Sikandar Zulqarnain
Minhaj University Lahore

Research Article

Keywords: Skeleton-based Representation, Graph Convolutional Networks, Motion-guided dynamic graph topology, Adaptive temporal modeling, Hierarchical motion propagation

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11753261/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

MGDG-HMPN: Motion-Guided Dynamic Graph Topology and Hierarchical Motion Propagation Network for Skeleton-Based Action Recognition

Tahir Hussain[1], Wang Bin[1]*, Jinfang Sheng[1], Saeeda Saeed[1], Ikramov Ilham[1], Wajid Ali[1], Sikandar Zulqarnain[2]
[1]School of Computer Science and Engineering, Central South University, Changsha 410017, Hunan, China
[2]Department of Computer Science, Minhaj University Lahore, 54770, Pakistan

*Corresponding author: School of Computer Science and Engineering, Central South University, Changsha 410017, Hunan, China Email address: wb_csut@csu.edu.cn (W. B).

Abstract:

Skeleton-based human action recognition has achieved significant progress with Graph Convolutional Networks (GCNs). However, existing methods construct graph topology using anatomical priors or feature similarity, which fail to explicitly capture motion-driven functional dependencies between joints. Moreover, temporal modeling typically relies on fixed multi-scale convolutions that treat all motion patterns uniformly, limiting adaptability to diverse action dynamics. In this paper, we propose a motion-guided spatial–temporal learning framework that explicitly leverages joint velocity as a supervisory signal for both graph construction and temporal modeling. We introduce a Motion-Guided Dynamic Graph Topology (MGDG), which constructs instance-level graph topology from inter-joint velocity correlations, enabling the modeling of functional co-movement patterns beyond anatomical connectivity. To further address heterogeneous temporal dynamics, we propose a Hierarchical Motion Propagation Network (HMPN), which adaptively selects temporal receptive fields via velocity-driven gating and enhances feature interaction through bidirectional cross-scale propagation. The proposed framework is instantiated within a four-stream architecture over joint, bone, motion, and bone-motion modalities. Extensive experiments across standard benchmarks demonstrate that our method competitive with state-of-the-art approaches, achieving 92.73% (NTU-60 X-Sub), 88.55% (NTU-120 X-Sub) and 96.77% (NW-UCLA), particularly strong gains on actions exhibiting complex and varying motion patterns.

Keywords: Skeleton-based Representation, Graph Convolutional Networks, Motion-guided dynamic graph topology, Adaptive temporal modeling, Hierarchical motion propagation

1. Introduction

Skeleton-based human action recognition has become a fundamental problem in computer vision with applications in surveillance, human–computer interaction, and robotics [1]. By abstracting the human body as a compact set of joint coordinates, skeleton data offers inherent robustness to background clutter, illumination variation, and appearance changes [2]. The availability of large-scale benchmarks such as NTU RGB+D [3] and Kinetics-Skeleton [4] has further accelerated research in this direction.

Graph Convolutional Networks (GCNs) have become the dominant paradigm for modeling skeleton data, treating joints as nodes and bones as edges. ST-GCN [4] pioneered this approach with interleaved spatial and temporal graph convolutions over a fixed anatomical graph. 2s-AGCN [5] introduced learnable adjacency matrices that complement the fixed skeleton graph, and CTR-GCN [6] advanced this with channel-wise topology refinement, learning a shared topology refined by channel-specific correlations. More recently, BlockGCN [7] revealed that jointly optimizing the adjacency matrix with model weights causes a decay of bone-connectivity information, and re-encoded topology using graph distances. MTR-GCN [8] proposed multi-view topology refinement combining dynamic and static joint relations, while several works [9, 10] incorporated transformer-based or high-order feature modeling. Despite this progress, two fundamental limitations persist.

First, graph topology in existing methods is derived from anatomical priors [4] or feature-similarity attention [5, 6], both of which overlook motion-driven functional relationships between joints. During clapping, both hands exhibit synchronized velocity profiles despite being anatomically distant; during kicking, the hip and ankle co-activate with strong velocity correlation. Such co-movement patterns are naturally encoded in inter-joint velocity correlations but remain under-exploited, yielding spatial representations insensitive to instance-specific motion.

Second, temporal modeling relies on fixed multi-scale convolutions [6, 8] applied uniformly across all samples. This assumes homogeneous motion dynamics, which is unrealist. Although recent methods such as AT-GCN [11] explore slow–fast pathways, they apply predefined frame-rate schedules rather than input-adaptive temporal selection. This mismatch between fixed temporal kernels and heterogeneous action dynamics limits discriminative power.

To address these limitations, we propose a unified motion-guided spatial–temporal learning framework built upon a channel-wise topology refinement GCN backbone [6]. It introduces two core components: (1) a MGDG that constructs instance-level graph topology from inter-joint velocity correlations, capturing functional dependencies beyond anatomical connectivity; and (2) HMPN that replaces fixed temporal convolutions with velocity-driven gating and bidirectional cross-scale propagation for adaptive temporal modeling.

The main contributions of this work are as follows:

- **Motion-Guided Dynamic Graph (MGDG):** We introduce a motion-guided learning paradigm that leverages joint velocity as an explicit supervisory signal to construct instance-level graph topology via inter-joint velocity correlations, capturing functional co-movement patterns beyond anatomical connectivity and feature similarity.
- **Hierarchical Motion Propagation Network (HMPN):** We propose a motion-adaptive temporal module with velocity-guided gating and cross-scale propagation to handle diverse action dynamics.
- **Motion-Guided Spatial–Temporal Integration:** We jointly model motion-driven spatial topology and adaptive temporal propagation within a unified framework.
- **Extensive Evaluation:** Extensive experiments on NTU RGB+D, NTU RGB+D 120 and NW-UCLA demonstrate competitive results with state-of-the-art methods.

2. Related Work

2.1 Skeleton-based Human Action Recognition

Early skeleton-based action recognition relied on handcrafted descriptors that encode joint trajectories, relative distances, and geometric transformations of the human body [12, 13]. While interpretable and computationally cheap, these features capture only shallow motion cues and generalize poorly to complex real-world actions [14]. With the rise of deep learning, recurrent architectures such as RNNs and LSTMs were introduced to model the temporal evolution of skeleton sequences [15, 16]. These methods treat skeletons as ordered coordinate vectors and learn temporal dependencies through recurrent structures, but fail to exploit the inherent graph structure of the human body, leading to suboptimal spatial reasoning [14]. More recently, transformer-based formulations [16, 17] capture long-range spatio-temporal dependencies through self-attention, yet still struggle to model structured joint relationships under varying motion conditions, motivating graph-based representations.

2.2 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) generalize convolution to non-Euclidean graph-structured data, making them well suited for relational modeling of skeleton joints. Spectral approaches define convolution in the graph Fourier domain via the Laplacian eigenbasis [18], but are computationally expensive and tied to a fixed graph structure. Spatial approaches instead perform convolution through local neighborhood aggregation [19], offering greater efficiency and flexibility on irregular structures; the first-order approximation of Kipf and Welling [18] further made GCNs scalable and widely applicable. Despite their success, standard GCN formulations assume a fixed or static graph topology, limiting their ability to adapt to the instance-specific structural variations inherent in human motion.

2.3 GCN-based Skeleton Action Recognition

Graph-based methods have become the dominant paradigm in skeleton-based recognition since ST-GCN [4] introduced spatio-temporal graph convolutions over a predefined skeletal structure. Subsequent research has focused on improving graph topology modeling along two primary axes.

Static versus dynamic topology. Static methods rely on predefined anatomical structures or manually designed adjacency matrices fixed during training and inference [4]. While such priors provide structural stability, they cannot capture action-specific or instance-specific joint dependencies. Dynamic methods address this by learning adaptive joint relationships from input features, typically through attention mechanisms or learned embedding that estimate pairwise correlations [5, 20]. Dynamic GCN [20] enriches topology learning by incorporating global contextual features from all joints, while recent attention-based variants [21, 22] dynamically infer per-head or per-sample topologies. Although effective, these methods derive structure primarily from feature similarity, which does not necessarily reflect the true functional relationships between joints during motion. Critically, most dynamic approaches

overlook explicit motion information, a key cue for identifying coordinated joint behavior.

Shared versus channel-wise topology. Another line of research improves graph expressiveness through channel-wise or group-wise topology learning. Early methods share a single adjacency matrix across all feature channels, limiting representational diversity. CTR-GCN [6] introduces channel-wise topology refinement, improving the expressiveness of learned graph structures with minimal additional overhead. BlockGCN [7] improves topology modeling by incorporating graph-distance-based structural encoding, while PCCTR-GCN [23] integrates pose correction with channel topology refinement. Nevertheless, even these approaches derive structure from feature interactions rather than explicit motion dynamics.

2.4 Temporal Modeling in Skeleton-based Recognition

Temporal modeling is crucial for capturing motion dynamics in skeleton-based recognition. Early methods employ standard temporal convolutions or recurrent architectures to model sequence evolution, but handle variable motion speeds poorly. To address this, multi-scale temporal convolutional networks introduce multiple kernel sizes or dilation rates to capture short- and long-range dependencies simultaneously [6, 24]. While effective, these methods apply fixed temporal receptive fields uniformly across all samples, regardless of the underlying motion dynamics. Recent approaches explore attention-based temporal modeling to adaptively focus on discriminative time steps [22, 25], and slow–fast pathways [26] process actions at multiple frame rates. However, these methods still lack explicit modeling of motion speed, which fundamentally determines how temporal information should be aggregated, limiting their ability to handle heterogeneous action dynamics.

In summary, despite significant advances in spatial–temporal modeling, existing methods still lack explicit modeling of motion-driven joint dependencies and adaptive temporal dynamics. Most approaches rely on feature similarity, while temporal modeling commonly uses fixed multi-scale operations that cannot adapt to varying motion patterns. To address these limitations, we propose a motion-guided spatial–temporal framework that incorporates velocity information into both graph construction and temporal modeling for adaptive motion-aware representation learning.

3. Method

3.1 Problem Formulation and Overview

Given a skeleton sequence, we represent it as a tensor $X \in \mathbb{R}^{N\times C\times T\times V\times M}$, where N is the batch size, C the number of input channels, T the number of frames, V the number of body joints and M the maximum number of persons per sample. Skeleton-based action recognition aims to learn a mapping $f_\theta: X \to Y$, where $Y \in \{1, \dots, K\}$ denotes the action label space and θ collects all learnable parameters.

The human skeleton is modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_V\}$ is the joint set and $\mathcal{E}$ the edge set. The graph is encoded by an adjacency matrix $A \in \mathbb{R}^{V\times V}$, whose entry $a_{i,j}$ denotes the connectivity strength between joints v_i and v_j. In conventional GCNs [4, 6], the representation of node v_i is obtained by aggregating features over its neighborhood $\mathcal{N}(v_i)$ using a shared topology:

$$z_i = \sum_{v_j \in \mathcal{N}(v_i)} a_{i,j}\, x_j W \qquad (1)$$

Where W is a learnable transformation matrix, as shown in Eq. (1), the same adjacency A is shared across all samples and channels, preventing the model from adapting to the motion-specific and instance-dependent joint relationships that characterize human actions.

We argue that this static formulation is fundamentally misaligned with the dynamic nature of human motion. Our framework, illustrated in Fig. 1 addresses this through two motion-guided components: MGDG that injects velocity-driven functional dependencies into the spatial topology (Sec. 3.4), and HMPN that replaces fixed temporal convolutions with velocity-adaptive temporal modeling (Sec. 3.5). Both components are integrated into a channel-wise topology refinement GCN backbone [6] and applied in the deeper layers of a four-stream architecture. This formulation highlights that conventional GCNs rely on shared topology, which limits their ability to capture motion-dependent and instance-specific joint relationships.

3.2 Input Modalities

Following multi-stream practice [5, 27], our framework processes four complementary skeleton modalities to capture comprehensive spatial–temporal cues. While these modalities are commonly used, we leverage motion not only as an input stream but as a supervisory signal that governs both spatial topology construction and temporal modeling. The joint modality uses the raw 3D coordinates $v_{i,t} = (x_{i,t}, y_{i,t}, z_{i,t})$, encoding the absolute spatial configuration at frame$t$. The joint-motion modality captures short-term

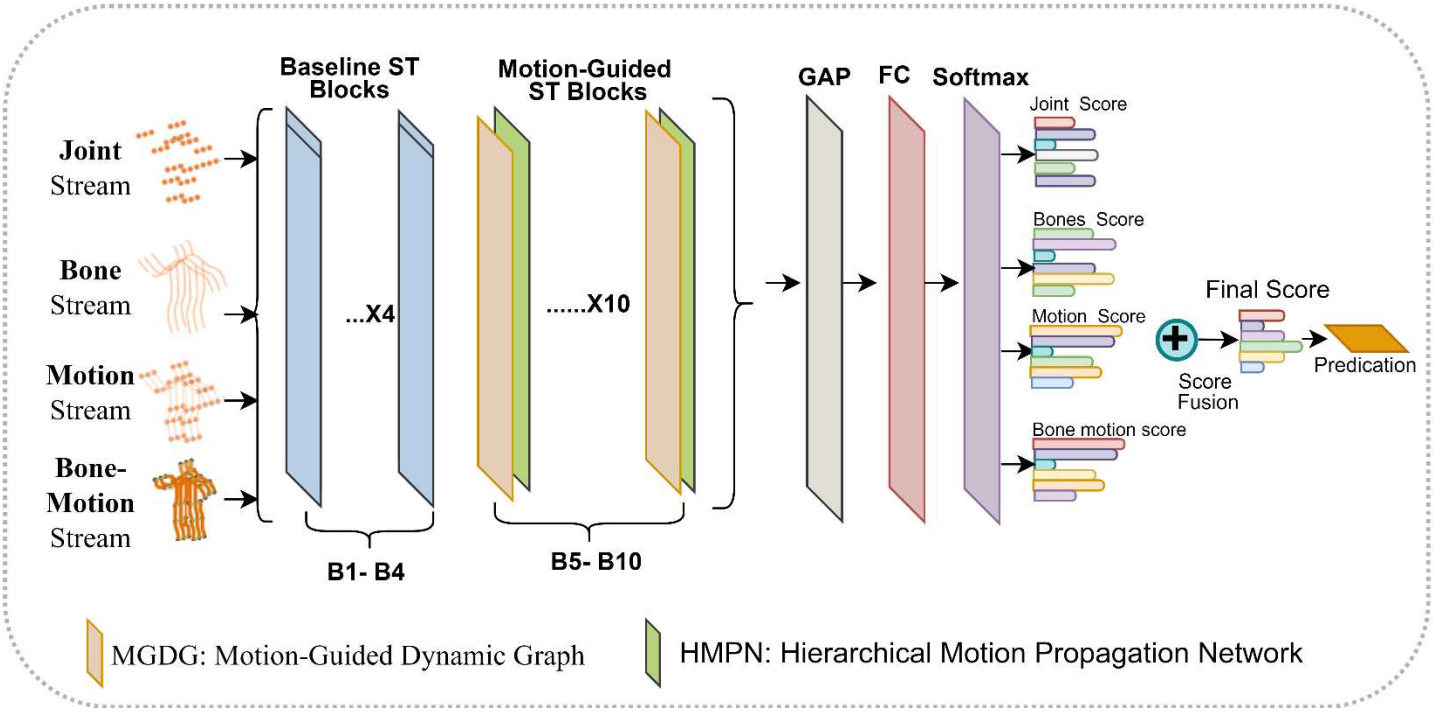


Fig. 1 Overview of the proposed MGDG-HMPN framework. Four input streams are processed through a 10-block spatial–temporal network. Blocks 1–4 are Baseline ST Blocks, while Blocks 5–10 are Motion-Guided ST Blocks incorporating the proposed MGDG and HMPN for adaptive spatial–temporal modeling. Final predictions from all streams are fused at the score level for action recognition.

dynamics through frame-wise displacement $v_{i,t}^{m} = v_{i,t+1} - v_{i,t}$, modeling instantaneous joint velocity. The bone modality represents each bone as a directed vector between physically connected joints, $e_{ij,t} = v_{j,t} - v_{i,t}$ encoding structural geometry while being invariant to global translation. Finally, the bone-motion modality models the temporal evolution of skeletal structure $e_{ij,t}^{m} = e_{ij,t+1} - e_{ij,t}$, capturing higher-order motion of limb segments.

3.3 Channel-Topology Refinement Graph Convolution

We adopt the channel-wise topology refinement GCN [6] as the spatial backbone due to its strong representational capability and low computational overhead.

Given an input feature tensor$X \in \mathbb{R}^{N\times C_{in}\times T\times V}$, three 1×1 convolutional projections produce intermediate embeddings:

$$\Phi_1 = W_1 X,\ \Phi_2 = W_2 X,\quad \Phi_3 = W_3 X \tag{2}$$

Where $\Phi_1, \Phi_2 \in \mathbb{R}^{N\times C_r\times T\times V}$ lie in a reduced relation space with $C_r = C_{in}/r$ $(r = 8)$, and $\Phi_3 \in \mathbb{R}^{N\times C_{out}\times T\times V}$. Global temporal context is obtained by temporal average pooling:

$$\bar{\phi}_1 = \frac{1}{T}\sum_{t=1}^{T} \Phi_1(:,t,:),\ \bar{\phi}_2 = \frac{1}{T}\sum_{t=1}^{T} \Phi_2(:,t,:) \tag{3}$$

Channel-wise pairwise joint correlations are then computed through a non-linear outer difference:

$$M_i = W_4 \cdot \tanh(\bar{\phi}_1^{T} - \bar{\phi}_2) \tag{4}$$

Where $M_i \in \mathbb{R}^{N\times C_{out}\times V\times V}$ encodes channel-specific joint dependencies. The predefined topology is refined as:

$$\widetilde{A}_i = \alpha M_i + PA_i \tag{5}$$

Where $PA_i \in \mathbb{R}^{V\times V}$ a learnable adjacency matrix is initialized from the anatomical graphA_i, and α is a learnable scalar initialized to zero. Spatial aggregation across all K partitions with a residual connection yields:

$$\mathrm{Y} = \mathrm{ReLU}\left(\mathrm{BN}\left(\sum_{i=0}^{K-1} \tilde{A}_i \otimes \Phi_3\right) + f_{down}(X)\right) \tag{6}$$

Where f_{down} is an identity mapping when $C_{in} = C_{out}$, or a 1×1 convolution otherwise. While this backbone provides channel-wise topology refinement based on feature interactions, it remains primarily driven by appearance features and does not explicitly incorporate motion cues. This limitation motivates our MGDG, which introduces velocity-aware topology construction for instance-specific functional dependencies.

3.4 Motion-Guided Dynamic Graph Topology (MGDG)

Anatomical and feature-similarity graphs overlook the velocity-driven functional dependencies that arise naturally during motion. During clapping, both hands exhibit highly synchronized velocity profiles despite being anatomically distant; during kicking, the hip and ankle co-activate with strong temporal correlation. Neither relationship is recoverable from a static skeleton graph or a feature-similarity adjacency. Our key insight is that functional co-movement is explicitly encoded in inter-joint velocity correlations. Based on this, we construct instance-level graph topology from these correlations, enabling explicit modeling of functional dependencies between joints as shown in *Fig. 2*.

Motion velocity and correlation. Given input $X \in \mathbb{R}^{N\times C\times T\times V}$, the per-joint velocity magnitude at frame t is computed as a measure of temporal motion dynamics. Velocity magnitude provides a direction-invariant measure of motion intensity, making it robust to viewpoint variations and sign ambiguity in joint displacement. Formally, it is defined as

$$m_{n,t,v} = \|X_{n,:,t+1,v} - X_{n,:,t,v}\|_2, \quad t = 1, \dots, T-1. \tag{7}$$

To achieve scale invariance across samples and joints, velocity magnitudes are standardized per joint:

$$\hat{m}_{n,t,v} = \frac{m_{n,t,v} - \mu_{n,v}}{\sigma_{n,v} + \epsilon} \tag{8}$$

Where $\mu_{n,v}$ and $\sigma_{n,v}$ are the temporal mean and standard deviation of joint v in sample n. The inter-joint motion correlation matrix is then obtained as the normalized temporal inner product:

$$C_{n,i,j} = \frac{1}{T-1}\sum_{t=1}^{T-1} \hat{m}_{n,t,i} \cdot \hat{m}_{n,t,j} \tag{9}$$

Where $C_{n,i,j} \in [-1,1]$ quantifies the degree of synchronized motion between joints i and$j$. Correlation captures temporal synchronization between joints, reflecting coordinated movement patterns beyond spatial proximity, which directly reflects coordinated movement patterns beyond spatial proximity. Unlike feature-similarity attention, C is computed directly from kinematic velocity and is therefore a physically grounded measure of functional coupling.

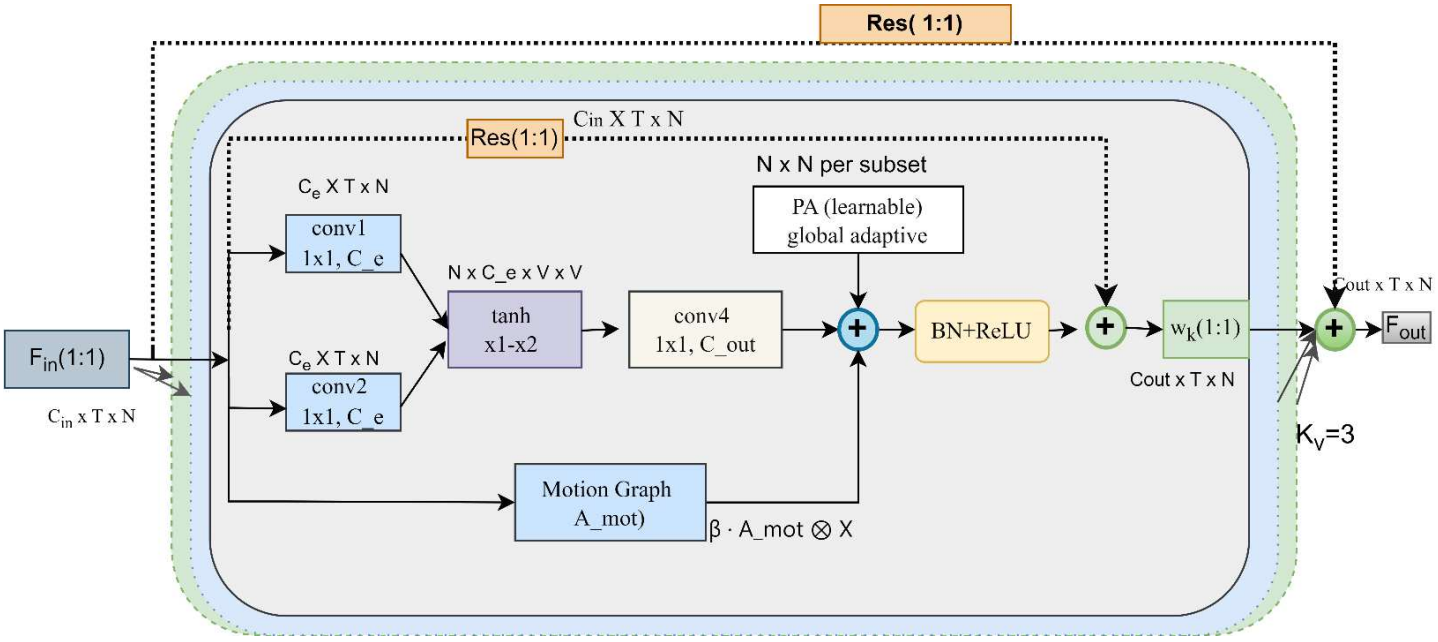


Fig. 2 Architecture of the proposed Adaptive Graph Convolutional Layer (AGCL). The layer integrates three complementary topologies: a learnable global adaptive graphP_A, a channel-refined topology constructed through embedding functions ϕ and θ followed by a tanh nonlinearity, and the proposed instance-specific motion graph A_{mot} derived from inter-joint velocity correlations. The blue blocks denote trainable parameters, $\oplus$ represents element-wise summation and $\otimes$ denotes matrix multiplication. The scalar β (initialized to 0) controls the contribution of the motion-guided topology, while α weights the channel-refined topology. Kv denotes the number of graph partitions, and the dashed residual connection is applied only when $C_{in} \neq C_{ou}$

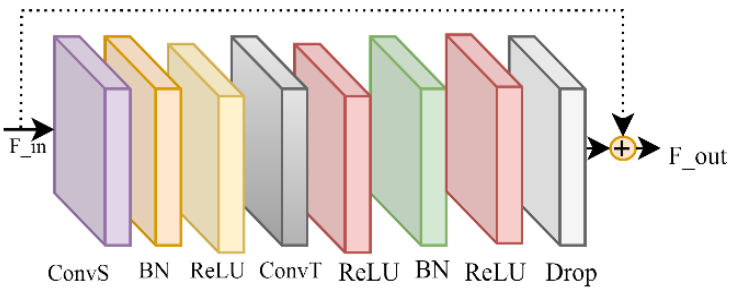


Fig. 1 Basic STC-GC building block of the proposed AGCL framework. Each block consists of a spatial graph convolution module (ConvS_S) followed by a temporal modeling module (ConvT), with BN and ReLU applied after each operation. In the Baseline ST Blocks (B1–B4), ConvT_ uses MS-TCN for temporal modeling, whereas in the Motion-Guided ST Blocks (B5–B10), it is replaced by the proposed HMPN for adaptive temporal modeling. A dropout layer is applied before the residual connection that adds the input features back to the output

Sparsification and normalization. To suppress spurious correlations induced by noise, we apply a hard threshold $\tau = 0.3$, retaining $\tilde{C}_{n,i,j} = C_{n,i,j}$ if $|C_{n,i,j}| > \tau$ and zero otherwise. This operation filters out weak or noisy dependencies caused by minor joint fluctuations, ensuring that only meaningful functional relationships are preserved. The sparse matrix is row-normalized to yield the instance-specific motion adjacency:

$$A^{mot}_{n,i,j} = \frac{\tilde{C}_{n,i,j}}{\sum_{j=1}^{V} |\tilde{C}_{n,i,j}| + \epsilon} \tag{10}$$

The motion graph is injected into the spatial convolution through a learnable scalar β:

$$Z_i = \mathrm{CTRGC}(X, PA_i, \alpha) + \beta \cdot (A^{mot} \otimes X) \tag{11}$$

The scalar β lets the network adaptively determine the contribution of motion-guided topology, and its zero initialization ensures the motion graph is introduced gradually "without disturbing early convergence, a strategy inspired by residual-branch initialization in adaptive graph learning [5, 6]. The MGDG block output then follows Eq. (6) with Z_i replacing the standard CTR output. MGDG is applied only in the deeper blocks (Blocks 5–10). This design is motivated by the observation that deeper layers encode higher-level semantic motion patterns, where functional co-movement becomes more Discriminative than local spatial structure.

3.5. Hierarchical Motion Propagation Network

Unlike conventional multi-scale temporal modeling [6, 24] which applies fixed receptive fields uniformly, our approach explicitly conditions temporal aggregation on motion dynamics, enabling input-adaptive temporal reasoning. Fast explosive actions such as kicking or punching require fine-grained short-range modeling, while slow deliberate actions such as writing or drinking water demand longer temporal context. The HMPN resolves this mismatch through three components: multi-scale temporal branches, a velocity-guided gating mechanism, and bidirectional cross-scale propagation (Fig. 4).

3.5.1 Multi-Scale Temporal Branches

Given spatial features $H \in \mathbb{R}^{N\times C\times T\times V}$, three parallel dilated temporal convolution branches operate at distinct receptive fields. The fast branch uses kernel size $k = 3$ and dilation$d = 1$, giving an effective receptive field $RF = 1 + (k-1)\cdot d = 3$ frames. The medium branch uses $k = 5$, $d = 2$, yielding $RF = 9$ frames. The slow branch uses$k = 7$, $d = 4$, yielding $RF = 19$ frames. Each branch output is:

$$F_s = \mathrm{ReLU}\left(\mathrm{BN}\left(W_s^{temp} * H\right)\right), s \in \{\mathrm{fast,med,slow}\} \tag{12}$$

3.5.2 Velocity-Guided Gating Mechanism Rather than fusing branch outputs with fixed weights, we introduce a Velocity Gate that dynamically assigns branch importance from the global motion characteristics of each input. The gate first applies global spatiotemporal pooling, producing a compact representation that implicitly encodes motion speed and temporal variation, allowing the gating mechanism to adaptively select appropriate temporal scales. $\bar{h} = \frac{1}{TV}\sum_{t,v} H_{(:,:,t,v)} \in \mathbb{R}^{N\times C}$, then a two-layer MLP with hidden dimension $C/2$:

$$w = \mathrm{Softmax}\left(W_2^g \cdot \mathrm{ReLU}\left(W_1^g \bar{h} + b_1^g\right) + b_2^g\right) \in \mathbb{R}^{N\times 3} \tag{13}$$

Where $w = [w_{fast}, w_{med}, w_{slow}]$ satisfies $\sum_s w_s = 1$ by construction.

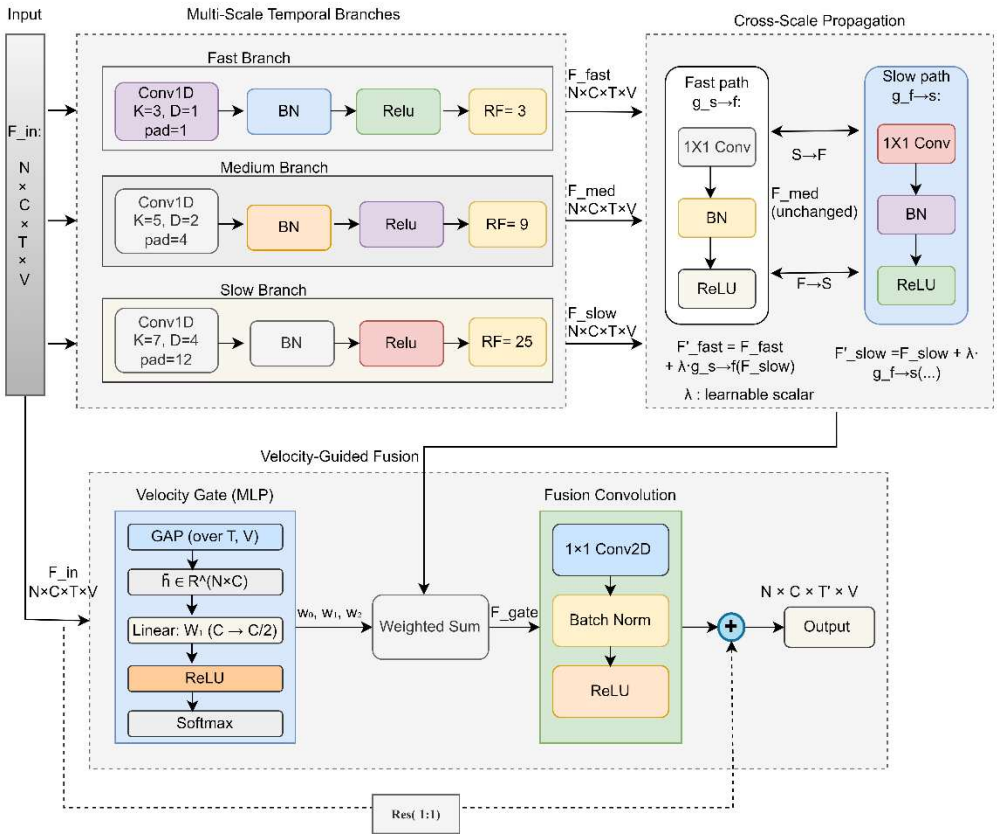


Fig. 4 Overall architecture of the proposed HMPN. Input features are processed through three parallel temporal branches with progressively larger receptive fields fast, medium, and slow capturing high-, mid-, and low-frequency motion patterns, respectively. The fast and slow branches exchange contextual information through bidirectional Cross-Scale Propagation with a learnable coupling scalar λ\, while the medium branch remains unchanged. The refined branch features are then combined through velocity-guided weighted fusion, followed by a 1×11 convolution with BN and ReLU, and finally added to the input via a residual connection.

The gated fusion is:

$$F_{gate} = \sum_{s\in\{fast,med,slow\}} w_s F_s \qquad (14)$$

This makes the temporal receptive field input-adaptive: fast actions receive higher w_{fast}, slow actions higher w_{slow} in contrast to fixed receptive fields in prior work. As a result, the model dynamically emphasizes short-term or long-term temporal dependencies based on the underlying motion characteristics of each input sequence.

3.5.3 Bidirectional Cross-Scale Propagation

To prevent isolated scale processing and enable coherent multi-scale interaction, we introduce Cross-Scale Propagation (CSP), which establishes bidirectional communication between the fast and slow branches before gated fusion. This bidirectional interaction ensures that fine-grained motion details and long-range temporal context are jointly modeled rather than processed independently.

$$F'_{fast} = F_{fast} + \lambda \cdot g_{s\to f}(F_{slow})$$
$$F'_{slow} = F_{slow} + \lambda \cdot g_{f\to s}(F_{fast}) \qquad (15)$$

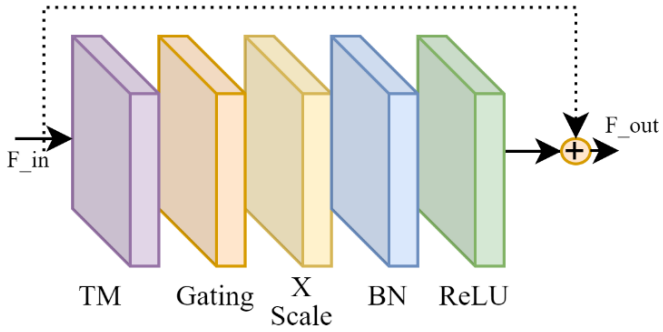


Fig. 5 Basic building block of the proposed HMPN. The block consists of three parallel Temporal Multi-scale (TM) branches (fast, medium, slow), followed by a velocity-guided gating mechanism that adaptively weights branch contributions for each input sample. Bidirectional Cross-Scale (X-Scale) Propagation enables interaction between

the fast and slow branches before feature fusion, BN, and ReLU, with a residual connection adding the input back to the output.

Where $g_{a\to b}(\cdot) = \mathrm{ReLU}\big(\mathrm{BN}(W_{a\to b} *\cdot)\big)$ is a 1×1 projection and λ is a learnable coupling scalar initialized to 0.1. This allows slow-branch long-range context to sharpen fast-branch local features, and vice versa. The propagated features F'_{fast} and F'_{slow} replace F_{fast} and F_{slow} in Eq. (14). The complete HMPN Block output, with residual connection, is:

$$Y_{HMPN} = \mathrm{ReLU}\big(\mathrm{BN}\big(W_{fuse} * F_{gate}\big) + H\big) \tag{16}$$

Where $W_{fuse} \in \mathbb{R}^{C\times C\times 1\times 1}$ is a 1×1 fusion convolution. The full HMPN module stacks B such blocks sequentially, $H^{(b+1)} = \mathrm{HMPNBlock}\big(H^{(b)}\big)$ for $b = 0, \dots, B-1$.

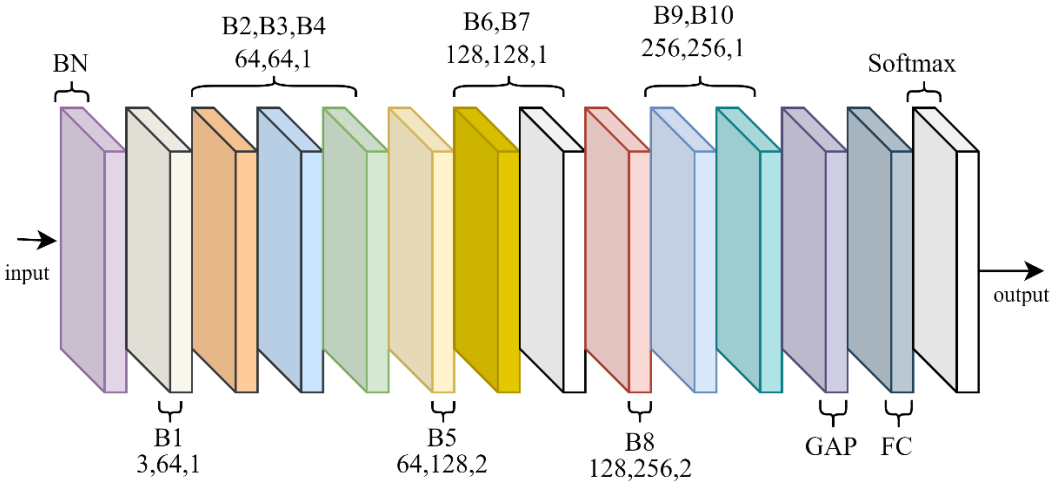


Fig. 6 Overall network architecture. The network consists of ten spatial–temporal blocks (B1–B10), where B1–B4 are Baseline ST Blocks and B5–B10 are Motion-Guided ST Blocks. The three integers in each block denote the input channels, output channels, and stride, respectively. GAP denotes the global average pooling layer

3.6 Overall Architecture Block

The complete network (Fig. 6) processes the input through batch normalization on the reshaped tensor of shape $N\times MVC\times T$, followed by ten stacked spatial–temporal blocks. Blocks 1–4 are Baseline ST Blocks combining CTRGC and MS-TCN, with channel dimensions $3\to 64\to 64\to 64\to 64\to 128$ and stride-2 temporal downsampling at Block 5 (basic block shown in (Fig. 3). Blocks 5–10 are Motion-Guided ST Blocks integrating MGDG and HMPN, with channel dimensions $128\to 128\to 128\to 256\to 256\to 256$ and stride-2 down sampling at Block 8. This design appliesmotion-guided topology modeling and adaptive temporal modeling at deeper semantic levels, where motion patterns become more informative and discriminative. Global average pooling over the temporal and spatial dimensions yields a feature vector $z\in\mathbb{R}^{N\times 256}$, passed through dropout ($p = 0.3$) and a linear classifier:

$$\hat{y} = \mathrm{Softmax}\big(W_{fc}z + b_{fc}\big) \tag{17}$$

where j, b, jm *and* bm denote the joint, bone, joint-motion, and bone-motion streams, respectively. The fusion weights w_s are determined by grid search on the validation set of each benchmark. Each stream is trained end-to-end using the cross-entropy loss.

Multi-stream fusion. The final prediction is obtained by weighted score-level averaging over the four streams:

$$S_{final} = \sum_{s\in\{j,b,jm,bm\}} w_s\cdot S_s, \quad \sum_s w_s = 1 \tag{18}$$

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_{n,k}\log\hat{y}_{n,k} \tag{19}$$

Where $y_{n,k}\in\{0,1\}$the one-hot ground-truth is label and $\hat{y}_{n,k}$ is the predicted class probability.

4. Experiments

1. Datasets

NTU RGB+D 60 [3]. This dataset contains 56,880 skeleton sequences spanning 60 action classes, performed by 40 subjects and captured from 3 camera views. Each skeleton comprises 25 body joints with 3D coordinates, with up to two persons per sample. We follow two standard protocols:

(1) Cross-Subject (X-Sub): 40,320 training / 16,560 testing sequences with disjoint subjects; (2) Cross-View (X-View): 37,920 training (cameras 2, 3) / 18,960 testing (camera 1) sequences. It is widely used as a standard benchmark for skeleton-based action recognition.

NTU RGB+D 120 [29]. An extension containing 114,480 sequences covering 120 action classes, performed by 106 subjects across 32 setups. We follow two standard evaluation protocols: (1) Cross-Subject (X-Sub) and (2) Cross-Setup (X-Set) even-numbered setups for training, odd-numbered for testing. It is a more challenging benchmark due to increased action diversity and larger subject variability. The dataset introduces greater intra-class variation and inter-class similarity, making it suitable for evaluating robust large-scale skeleton-based models.

Northwestern-UCLA dataset [30] is a multi-view skeleton-based action recognition dataset captured using three Kinect cameras. It contains 1,494 video clips across 10 action Categories, performed by 10 subjects from different Viewpoints. We [31] follow the standard evaluation protocol, where data from the first two camera views are used for training and the remaining view is used for testing.

4.2. Implementation Details

All experiments are implemented in PyTorch [28] and conducted on one NVIDIA RTX 3090 GPUs. Each of the four streams (joint, bone, joint-motion, bone-motion) is trained independently with identical architecture and hyperparameters.

We optimize the network using stochastic gradient descent (SGD) with Nesterov momentum of 0.9 and weight decay of 4×10^{-4}. The base learning rate is set to 0.1, with a 5-epoch linear warm-up to stabilize early training, followed by step decay that scales the learning rate by 0.1 at epochs 35 and 55. Training runs for a total of 65 epochs with a batch size of 64. Cross-entropy is used as the loss function (Eq. 19).

For the NTU RGB+D 60 and NTU RGB+D 120 datasets, we follow the preprocessing pipeline of [6], including raw skeleton parsing, denoising with body-energy selection and coordinate normalization, and temporal resampling of each sequence to 64 frames via valid crop resize. During training, random rotation is applied as the only data augmentation; during testing, center-crop sampling is used. For the NW-UCLA dataset, owing to its small scale, we adopt a batch size of 16, a temporal window of 52 frames, random view transformation as augmentation, and a repeat factor of 5 per epoch.

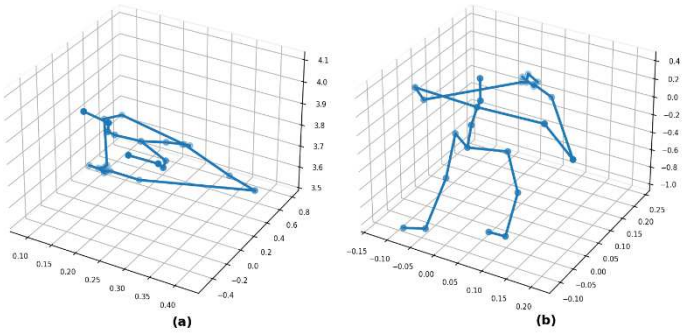


Fig. 7. Example of data preprocessing for the (drinking water) action on the NTU RGB+D dataset. (a) Original skeleton (b) Skeleton after preprocessing.

4.3. Comparison with State-of-the-Art Methods

We compare our method against representative state-of-the-art skeleton-based action recognition approaches on all four benchmarks. Following standard practice [4, 5, 6, 7], we report multi-stream fusion results obtained by simple score fusion techniques for fair comparison.

Table 1 compares the top 1 accuracy of the proposed methods with top state-of-the-art methods on the NTU RGB+D 60 dataset under both X-Sub and X-View protocols. Our four-stream model achieves 92.73% on X-Sub and 96.88% on X-View, outperforming all compared methods on the cross-subject split. As shown in the table, we also report different stream ablation studies (joint, bone, motion, and joint+bone), confirming the complementarity of the input streams. The full four-stream fusion further improves accuracy, demonstrating the effectiveness of the proposed motion-guided framework.

Table 1. Comparison of Top-1 Accuracy with state-of-the-art methods on NTU RGB+D 60 dataset.

Methods	Pub/Year	X-Sub (%)	X-View (%)
ST-GCN [4]	AAAI 2018	81.50	88.30
2s-AGCN [5]	CVPR 2019	88.50	95.10
MS-G3D [24]	CVPR 2020	91.52	96.20
Dynamic GCN[20]	ACM 2020	91.53	96.10
Shift-GCN [32]	CVPR 2020	90.72	96.50
DDGCN [33]	CVPR 2020	91.10	97.10
TA-GCN [34]	AAAI 2021	89.91	95.80
CTR-GCN [6]	CVPR 2021	92.40	96.80
STF-Ne't [35]	PR 2022	91.90	96.00
ML-STGNet [46]	TPAMI 2022	91.90	96.10
Efficient GCN[36]	TPMI 2023	91.70	95.70
LKA-GCN[37]	IEE 2023	90.70	96.10
GSTLN[38]	PR 2023	91.90	96.60

EMS-TAGCN [39]	CHB 2024	91.30	97.50
4s-MADT-GCN[45]	EAAI 2024	90.40	96.50
CTA-MSFA[40]	Multimedia Systems 2025	92.00	97.30
MTGCN[41]	Neurocomputing 2025	90.52	95.15
4s-DAMD-GCN [42]	Neurocomputing 2025	92.20	96.50
IGPN-Light [43]	Neural Networks 2025	92.20	96.50
TE-STGCN [44]	Applied Soft Computing 2026	92.30	96.71
MGDG-HMPN (J only)		91.12	95.15
MGDG-HMPN (B only)		90.28	94.25
MGDG-HMPN (M only)		87.79	93.81
MGDG-HMPN (J+B)		92.01	95.97
MGDG-HMPN Ours	**---**	**92.73**	**96.88**

Table 2 compares Top 1 accuracy of the proposed method with top state-of-the-art methods on the NTU RGB+D 120 dataset under both X-Sub and X-Set protocols. Our four-stream model achieves 88.10% on X-Sub and 89.21% on X-Set, outperforming all compared methods. These consistent gains confirm that the proposed motion-guided framework generalizes well to the larger and more challenging NTU-120 benchmark.

Table 2. Comparison of Top-1 with state-of-the-art methods on NTU RGB+D 120 dataset.

Method	Year	X-Sub (%)	X-Set (%)
ST-GCN [4]	AAAI 2018	70.71	73.21
2s-AGCN [5]	CVPR 2019	82.92	84.90
MS-G3D [24]	CVPR 2020	86.90	88.40
Shift-GCN [32]	CVPR 2020	85.90	87.60
CTR-GCN [6]	CVPR 2021	88.11	89.20
AdaSGN [56]	ICCV 2021	85.90	86.80
FR-AGCN[57]	Neurocomputing 2022	86.60	87.80
KA-AGTN (2s) [58]	Knowledge-Based 2022	86.1	87.00
CDGC [59]	IEE TCSVT 2021	86.3	87.81
LKA-GCN[37]	IEEE TVCG 2023	86.3	87.80
4sTF-Net [335]	Pattern Recognitio 2023	86.5	88.22
FRF-GCN [60]	AAI 2024	87.1	88.41
MaskCLR [61]	CVPR 2024	87.4	89.50
MTGCN[41]	Neurocomputing 2025	80.83	81.70
TE-STGCN [44]	Applied Soft Computing 2026	88.3	89.50
MGDG-HMPN Ours	**-----**	**88.55**	**89.67**

Table 3 compares the proposed MGDG-HMPN with top state-of-the-art methods on the Northwestern-UCLA dataset. Our model achieves 96.77%, performing on par with the strongest recent methods. The competitive result on this smaller, multi-view dataset demonstrates that the proposed motion-guided framework generalizes beyond the large-scale NTU benchmarks.

Table 3. Comparison of Top1 accuracy with state-of-the-art methods on the Northwestern-UCLA dataset.

Methods	Year	Acc (%)
2s-AGCN [5]	CVPR 2019	93.30
MS-G3D [24]	CVPR 2020	95.31
Shift –GCN[32]	CVPR 2020	94.60
CTR-GCN [6]	ICCV 2021	96.51
InfoGCN [47]	CVPR 2022	96.60
GSTLN[38]	Pattern Recognition 2023	94.81
DD-GCN[55]	ICME 2023	96.70
FR Head [48]	CVPR 2023	96.80
MSS-GCN[50]	Expert Systems 2024	96.80
JPFormer(4s) [49]	Signal Processing 2024	96.71
TE-STGCN [44]	Applied Soft Computing 2026	96.80
MGDG-HMPN Ours	**----**	**96.77**

Table 4 compares the proposed MGDG-HMPN with top state-of-the-art methods in terms of model parameters, FLOPs, and accuracy. The proposed model achieves 96.77% with 5.45M parameters and 6.40 GFLOPs, providing a balanced trade-off between accuracy and computational cost outperforming lightweight baselines in accuracy while remaining substantially more efficient than heavier models such as EMS-TAGCN (24.3M / 97.5%).

Table 4. Efficiency Comparison of model via parameters, FLOPS, and accuracy.

Method	Params (M)	FLOPs (G)	Acc (%)
ST-GCN [4]	3.1	---	88.30
2s-AGCN [5]	6.94	---	95.11
MS-G3D [24]	2.8	5.22	96.20
ASE-GCN[54]	6.00	---	96.20

MST-GCN [53]	12.00	---	96.61
4sSTF-net [52]	2.00	15.2	95.70
CTR-GCN [6]	1.5	---	96.80
EfficientGCN-B4 [51]	1.1	8.36	95.70
TE-STGCN [43]	5.66	7.31	96.60
EMS-TAGCN [39]	24.3	1.9	97.51
MGDG-HMPN (Ours)	**5.45**	**6.40**	**96.78**

Table 5 shows the CTR-GCN baseline reaches 89.71%, adding MGDG alone improves it to 90.26%, while HMPN alone yields only 89.70%, its velocity-driven gating depends on motion-aware spatial features that MGDG provides. Combining both achieves 91.12%, confirming the two components are complementary.

Table 5. Component-wise ablation study of MGDG and HMPN on NTU-60 X-Sub (Joint stream)

Model	S. Module	T. Module	MGDG	HMPN	Accu %
CTR-GCN Baseline	CTRGC	MS-TCN	✗	✗	89.71
+ MGDG	MGDG	MS-TCN	✓	✗	90.26
+ HMPN	CTRGC	HMPN	✗	✓	89.70
Full MGDG-HMPN	**MGDG**	**HMPN**	✓	✓	**91.12**

Table 6 shows the multi-branch design lifts the MS-TCN baseline (89.71%) to 89.98%, the velocity gate further raises it to 90.48%, and adding CSP achieves the best 91.12%, confirming that all three sub-components contribute progressively.

Table 6. Internal ablation study of HMPN sub-components on NTU-60 X-Sub (Joint stream

Variant	Multi-Branch	Velocity Gate	CSP	Accu%
MS-TCN baseline	✗	✗	✗	89.71
Multi-branch only	✓	✗	✗	89.98
+ velocity gate	✓	✓	✗	90.48
Full HMPN	✓	✓	✓	**91.12**

5. Qualitative Analysis and Discussion

We validate proposed mechanisms through qualitative analysis examining topology evolution, sample adaptation, anatomical alignment, confusion patterns, per-action gains, velocity-aware gating, and feature visualization.

5.1 Layer-wise Evolution of the Topology

CTR-GCN's topology P_A and our motion-guided topology A_{mot} both evolve through depth but via fundamentally different mechanisms. P_A is a shared parameter matrix, input-independent during inference; A_{mot} is dynamically recomputed from inter-joint velocity correlations per sample.

Fig. 8 compares both topologies for "rub two hands" at blocks 6, 8, and 9. P_A develops scattered arm-region cells at block 6, becomes denser at block 8, and settles into a column-dominant pattern at block 9 reflecting learned global priors independent of input. In contrast, A_{mot} preserves action-dependent structure at every depth, including strong negative leg–arm correlations from anti-phase leg motion during bimanual hand action. As depth increases, A_{mot} becomes denser while maintaining motion-specific relationships, indicating progressive temporal motion context aggregation. Negative correlations are structurally impossible for softmax-normalized topologies a direct consequence of velocity-correlation construction.

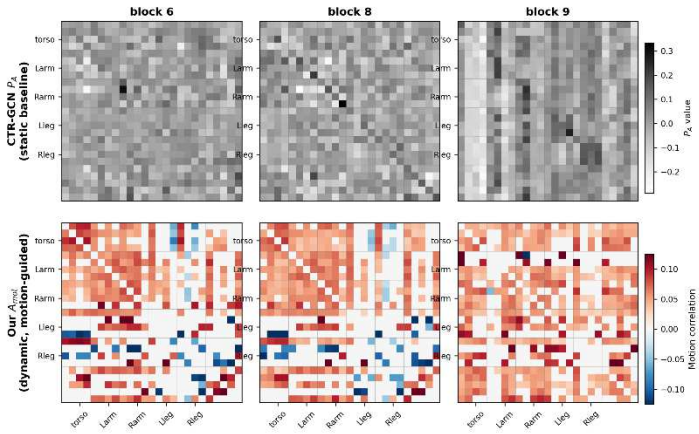


Fig. 8 Depth-wise evolution of CTR-GCN's shared topology P A (top) and the proposed motion-guided topology A mot (bottom) at blocks 6, 8, and 9 for rub two hands. Unlike the input-independent P_A , A_{mot} preserves action-specific motion patterns, including negative arm–leg correlations.

5.2 Sample-specificity of the Motion-guided Topology

A key property of MGDG is that the graph topology adapts to each input action. Fig. 9 fixes the network depth at block 6 and compares topology representations across three actions: clapping, rub two hands, and punching. CTR-GCN's topology P_A is shown once because it is shared across all samples and remains unchanged during inference.

In contrast, the proposed A_{mot} exhibits distinct action-dependent structures for each sample. Clapping produces strong positive correlations within the bilateral arm regions,

reflecting synchronized hand motion. Rub two hands generates dense arm correlations together with strong negative responses in the leg rows, corresponding to anti-phase lower-body stabilization during the action. Punching shows asymmetric arm activation accompanied by counter-stabilizing leg correlations. These motion-specific structures are absent in P_A, which cannot adapt to input dynamics. The comparison highlights that the proposed MGDG introduces input-conditional topology construction beyond the shared topology refinement used in CTR-GCN.

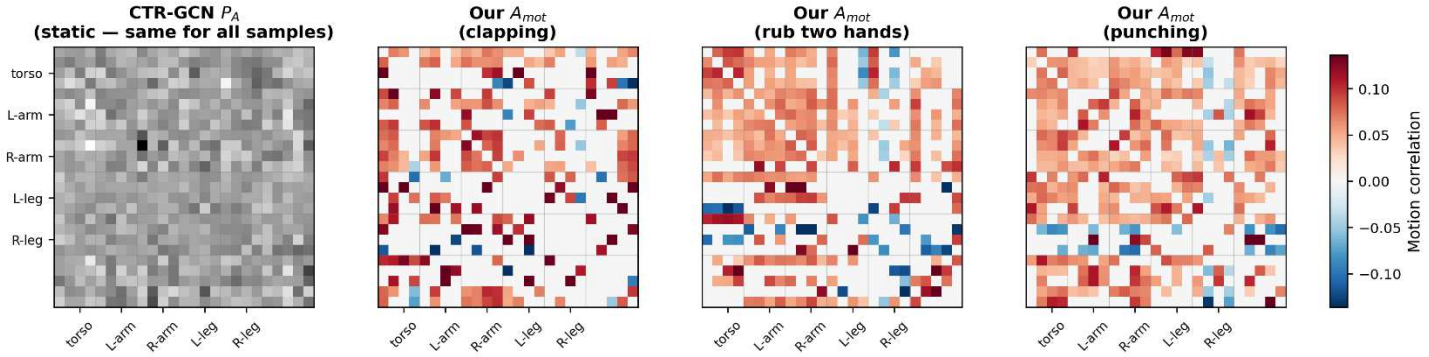


Fig. 9 Comparison of CTR-GCN's shared topology P_A and the proposed motion-guided topology A_{mot} at block 6 for three action samples: clapping, rub two hands, and punching. Unlike the sample-invariantP_A, A_{mot} adapts to each action and captures distinct motion-dependent correlations.

5.3 Anatomical Interpretation on Real-pose Skeletons

While heatmaps reveal topology differences, they do not show which body parts connect. Fig. 10 projects the strongest topology connections onto real skeleton poses for three top-improved actions.

CTR-GCN's shared topology (left) produces similar edge patterns across all actions, emphasizing global cross-body connections between head, spine, and hands. Being input-independent, the connectivity remains unchanged regardless of the underlying motion. In contrast, our motion-guided topology (right) generates distinct action-specific connections following the active anatomical regions of each action: hand-region actions produce dense hand-centered connectivity, while bilateral whole-body actions introduce coordinated hand–foot correlations spanning the full skeleton. This direct contrast demonstrates that the velocity-correlation formulation dynamically adapts graph connectivity to motion semantics rather than relying on a fixed adjacency prior.

5.4 Confusion Structure of the Corrections

Fig. 12 (a) compares confusion matrices of CTR-GCN and MGDG-HMPN on a representative subset of NTU-60 classes, along with their element-wise difference while Fig. 12 (b) shows the overall confusion matrix of NTU-60 X-view evaluation. The diagonal in Fig. 12 (a) entries remain nearly unchanged for saturated classes (writing, put on shoe, brush teeth), while most corrections occur off-diagonally.

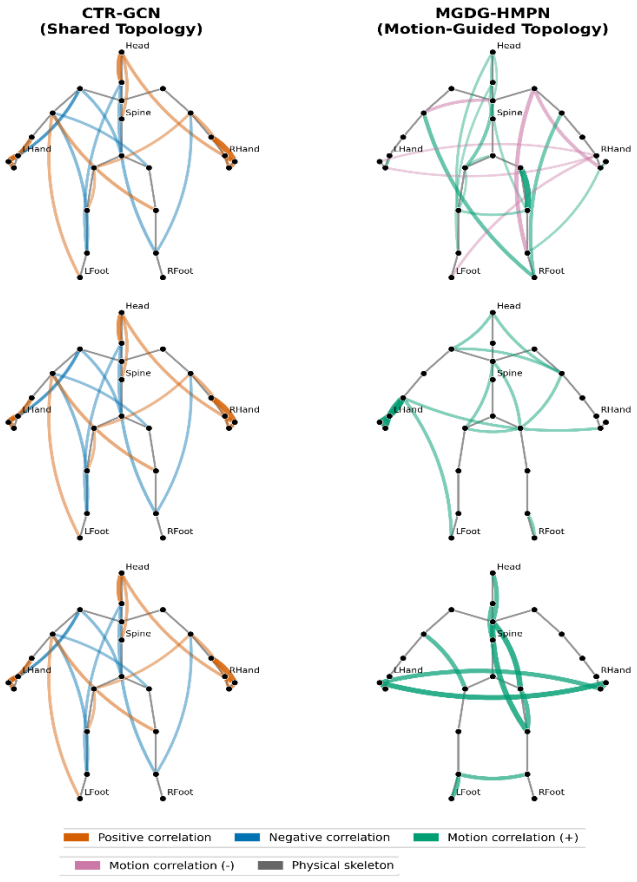


Fig. 10 Topology connections projected onto real skeleton poses for three top-improved actions. Left: CTR-GCN's shared topology P_A Right: the proposed motion-guided topology Amot, showing distinct action-specific connections that adapt to the underlying motion.

MGDG-HMPN reduces confusion between motion-similar pairs by reallocating confidence toward the correct class for instance, “drink water” recovers +1.5% while reducing confusion with “throw” by −1.0%, and “play phone” recovers +1.1% from samples previously misclassified as “drink water”. These structured corrections confirm that the proposed motion-guided topology improves fine-grained discrimination among actions with similar joint configurations but different motion dynamics.

5.5 Per-class Improvement Analysis

Fig. 11 shows the per-class accuracy difference (MGDG-HMPN − CTR-GCN) across all 60 NTU-60 X-Sub classes. The model improves 32 of 60 classes with an average gain of +0.62%, with the largest improvement (+8.36% on type on keyboard) on fine-grained, motion-intensive actions. Degradations concentrate on near-static actions (sneeze/cough, put on shoe, brush hair) where motion cues are weak; the asymmetry between gains (+8.36%) and the largest regression (−3.99%) confirms that MGDG-HMPN provides targeted improvements driven by motion content rather than uniform capacity expansion.

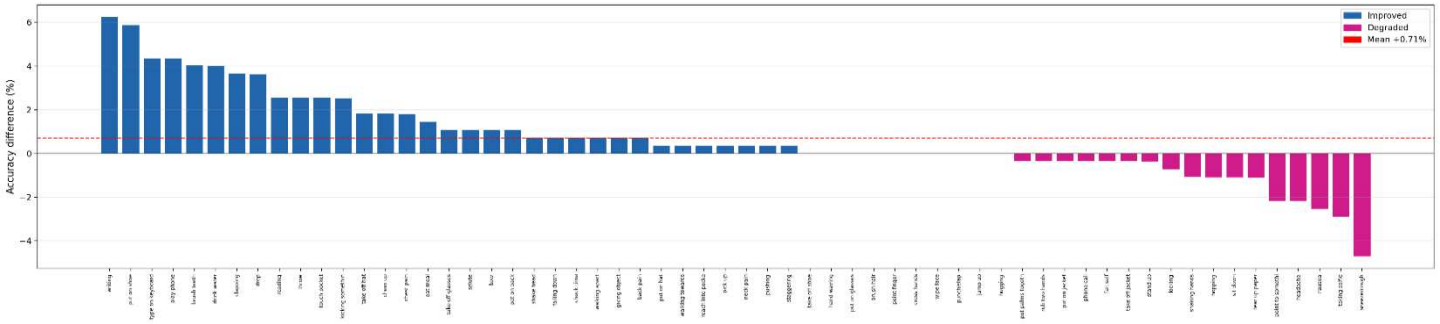


Fig. 11 Per-class accuracy difference (MGDG-HMPN − CTR-GCN) on NTU-60 X-Sub. Blue bars indicate gains, pink bars indicate regressions. The red dashed line shows the mean improvement.

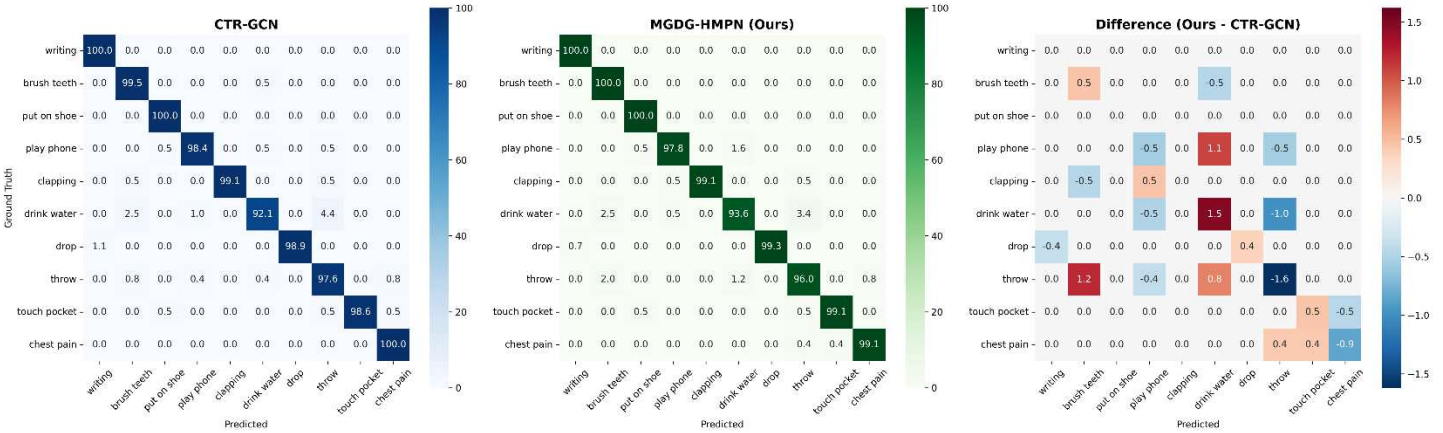


Fig. 12 (a). Confusion matrices on representative NTU-60 X-Sub classes. Left: CTR-GCN. Middle: MGDG-HMPN. Right: element-wise difference (Ours Vs CTR-GCN), highlighting corrected inter-class confusion

5.6 Top-improved and Top-degraded Actions

Fig. 13 highlights the ten most improved and ten most degraded classes from Fig. 12. The largest gains occur on fine-grained, motion-intensive actions such as type on keyboard (+8.36%), reading (+5.13%), and play phone (+4.00%), where temporal motion cues are crucial for discrimination. These results align with the proposed motion-guided design, which better captures dynamic hand interactions than CTR-GCN's static topology. In contrast, the degraded classes mainly involve near-static or weak-motion actions such as sneeze/cough, put on shoe, and brush hair. The performance drops are small and bounded in the

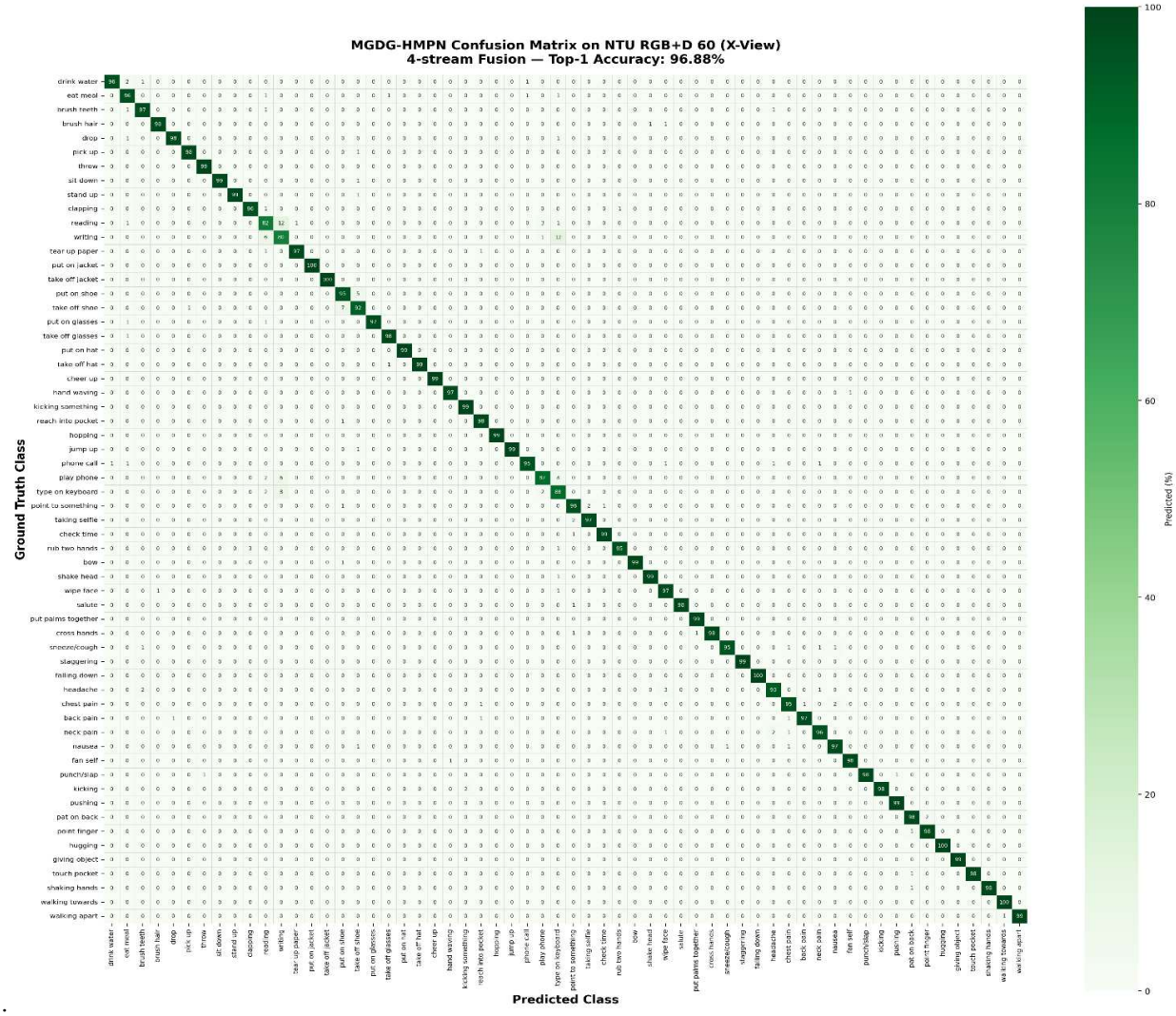


Fig. 12(b). Confusion matrix on the NTU RGB + D dataset with the X-View evaluation benchmark

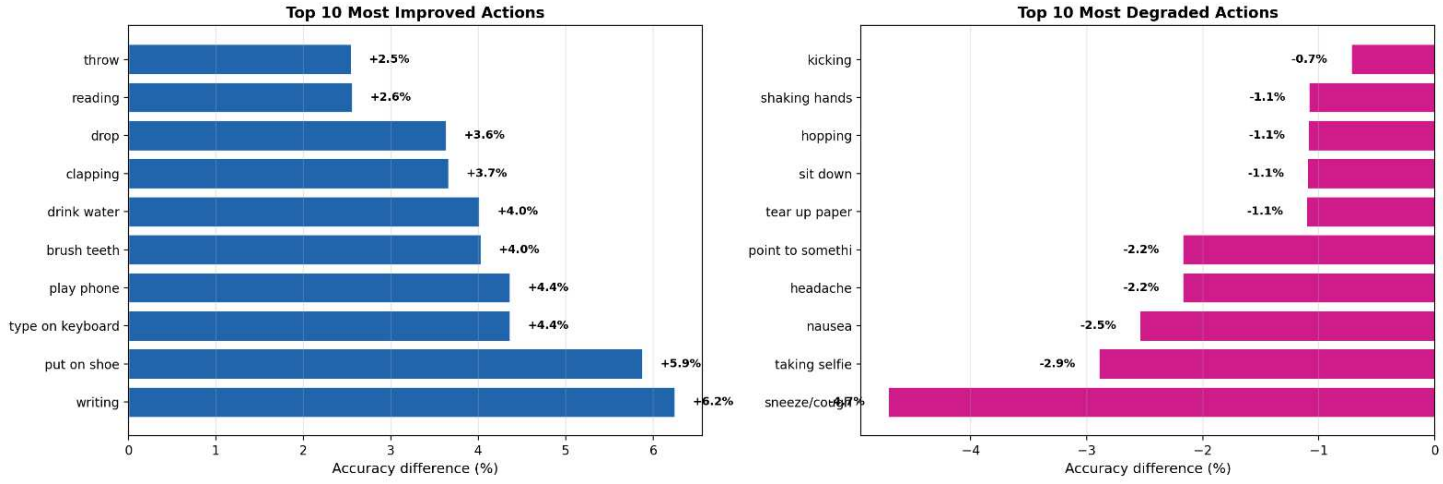


Fig. 13 Top-10 most improved (blue) and top-10 most degraded (pink) actions on NTU-60 X-Sub. Improvements concentrate on motion-intensive actions, while degradations occur in near-static classes

absence of strong motion signals, the velocity-correlation graph contributes little additional information, while the static prior in P_A continues to handle these classes effectively. Overall, MGDG-HMPN primarily benefits motion-dominant actions while remaining stable on low-motion categories.

5.7 Velocity-awareness of the HMPN Gate

We next analyze the second component, HMPN, which replaces the standard MS-TCN with three parallel temporal branches and a velocity-driven softmax gating mechanism. The key question is whether the gate meaningfully adapts to action dynamics.

Fig. 14 (A) shows that the learned gate is far from uniform across eight representative actions. Fine-motion actions such as writing and rub two hands favor the fast branch, while more sustained or coordinated motions such as clapping and point finger assign higher weight to the slow branch. This indicates that the gating mechanism is action-dependent rather than collapsed.

More importantly, Fig. 14 (B) compares the fast-branch weight with the mean joint velocity computed directly from

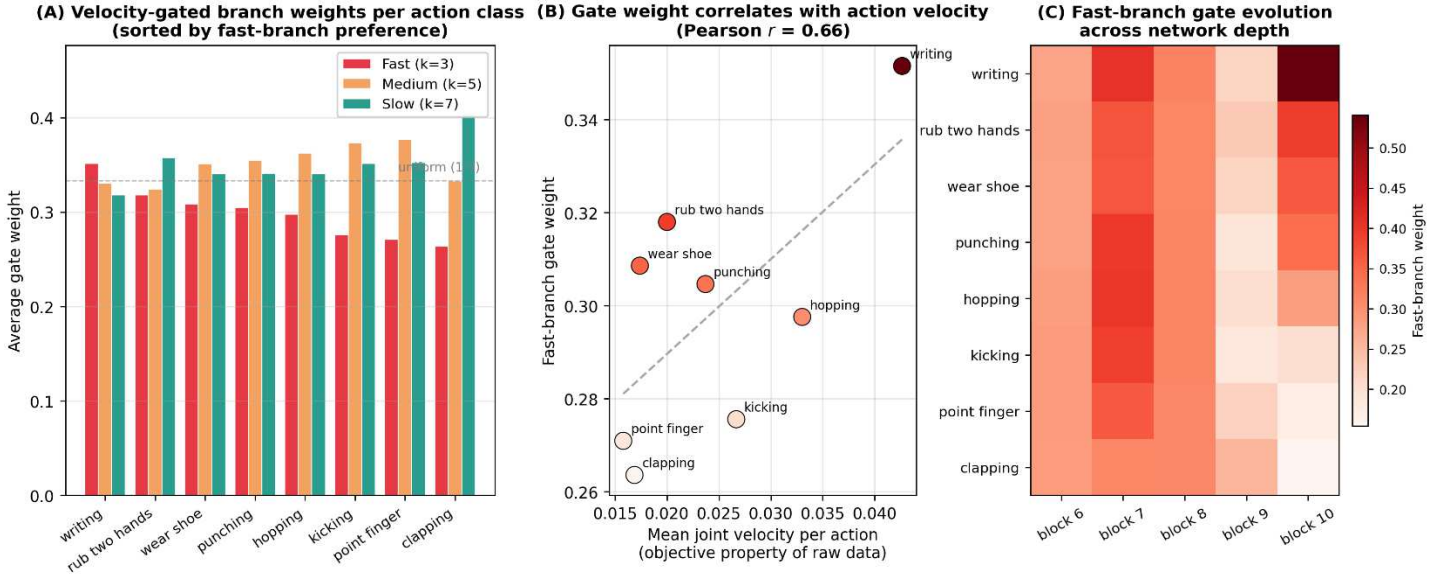


Fig. 14 HMPN velocity-gated multi-scale temporal selection. (A) Mean gate weights over three temporal branches for eight actions. (B) Correlation between fast-branch weight and mean joint velocity (Pearson$r = 0.66$). (C) Evolution of fast-branch weight across blocks 5–10, showing increasing class-specific specialization.

the raw skeleton sequence an objective property of the input that the gate does not observe directly. Despite this, the gate shows a clear positive correlation (Pearson$r = 0.66$) with motion magnitude, indicating that HMPN implicitly aligns its temporal-scale selection with the actual motion energy of the input. Fig. 14 (C) further shows that this behavior evolves with depth: early layers distribute weights more uniformly, while deeper layers produce sharper, class-specific preferences (e.g., stronger fast-branch activation for writing). This suggests a hierarchical specialization of temporal scale selection across network depth.

5. 8 t-SNE Visualization Comparison

Fig. 15. t-SNE visualization of multi-stream feature representations on NTU-60 X-Sub for six representative actions from the top-improved classes (Section 5.6). (a) CTR-GCN produces fragmented clusters with several classes splitting into multiple disconnected sub-regions and exhibiting visible inter-class overlap. (b) MGDG-HMPN yields tighter, well-separated clusters for the same six classes, indicating that the motion-guided topology and velocity-gated temporal module learn more discriminative and unified feature representations. Background hexagons (gray) show the global feature density across all 60 classes

5.9 Discussion

The eight figures together establish that the accuracy gains in Section 4 follow from the mechanisms we introduced, not from incidental capacity. MGDG produces topologies that (i) evolve distinctly from P_A through depth (Fig. 8), (ii) are genuinely sample-specific rather than collapsed (Fig. 9), and (iii) align with the anatomical structure of the underlying

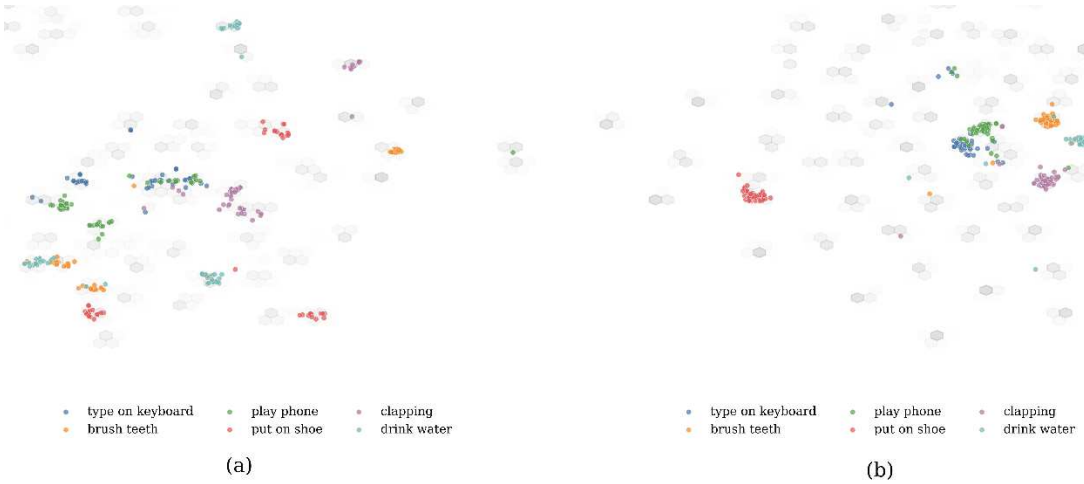


Fig. 15. *Visualization of t-SNE result (a) CTR-GCN Representation (b) Our model MGDG-HMPN*

motion on real-pose skeletons (Fig. 10). At the prediction level, the gains (iv) take the form of targeted disambiguation between motion-confusable classes rather than uniform score inflation (Fig. 11), and (v–vi) cluster on fine-motion, hand-region actions while regressions are confined to near-static actions (Fig. 12–13). HMPN's gate (vii) is action-dependent and quantitatively correlated with raw input velocity at $r = 0.66$ (Fig. 7). The absence of any of these properties a collapsed gate, a sample-invariantA_{mot}, gains evenly distributed across all classes, or anatomically random edges would have refuted our design claims; their joint presence supports them. Finally, t-SNE visualization of multi-stream features (Fig. 8) shows that MGDG-HMPN produces tighter, better-separated class clusters than CTR-GCN on the most-improved actions, confirming that the learned representations are not only more accurate but also more discriminative in the embedding space.

Our approach has two limitations. First, the velocity-correlation formulation of MGDG is sensitive to noisy or occluded joint trajectories, since A_{mot} is computed from inter-frame differences. Second, the three temporal scales in HMPN ($k = 3,5,7$) and the block range for MGDG (blocks 5–10) are fixed hyperparameters rather than learned, leaving room for adaptive-kernel or learnable-depth extensions.

6. Conclusion

In this work, we proposed MGDG-HMPN, a motion-guided spatial–temporal framework for skeleton-based action recognition. The model introduces two complementary components built on a CTR backbone. MGDG replaces the input-independent static topology with a dynamic graph constructed from inter-joint velocity correlations, enabling sample-specific and direction-sensitive connectivity that captures both synchronized and anti-phase joint interactions. HMPN replaces conventional multi-scale temporal convolution with a velocity-gated multi-branch design, where a learned softmax gate dynamically selects temporal receptive fields based on motion characteristics. Extensive experiments on NTU RGB+D 60, NTU RGB+D 120, and NW-UCLA demonstrate state-of-the-art performance, and detailed visualizations confirm that the gains arise from the proposed motion-guided mechanisms rather than incidental capacity.

7. Declarations

- **Funding Declaration:** This research received no external funding.
- **Conflict of interest:** The authors declare that they have no conflicts of interest or competing interests.
- **Author Contributions:** H.T. was the principal author responsible for the research and manuscript preparation; B.W. and F.J. supervised the research and provided guidance, while the remaining authors contributed to manuscript review and revision.

Reference:

[1] Chen, D., Chen, M., Wu, P., et al., 2025. Two-stream spatio-temporal GCN-transformer networks for skeleton-based action recognition. Scientific Reports, 15.

[2] Lee, J., Lee, M., Lee, D., Lee, S., 2023. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In: ICCV. pp. 10444–10453.

[3] Shahroudy, A., Liu, J., Ng, T.-T., Wang, G., 2016. NTU RGB+D: A large scale dataset for 3D human activity analysis. In: CVPR. pp. 1010–1019.

[4] Yan, S., Xiong, Y., Lin, D., 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI. pp. 7444–7452.

[5] Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: CVPR. pp. 12026–12035.

[6] Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W., 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: ICCV. pp. 13359–13368.

[7] Zhou, Y., Yan, X., Cheng, Z.-Q., Yan, Y., Dai, Q., Hua, X.-S., 2024. BlockGCN: Redefine topology awareness for skeleton-based action recognition. In: CVPR. pp. 2049–2058.

[8] Graph convolutional network with multi-view topology for lightweight skeleton-based action recognition (MTR-GCN), 2025. Symmetry, 17(8).

[9] Dirgová Luptáková, Kubovčík, M., Pospíchal, J., 2022. Wearable sensor-based human activity recognition with transformer model. Sensors, 22(5), p. 1911.

[10] An enhanced spatial-temporal graph convolution network with high-order features for skeleton-based action recognition, 2025. Scientific Reports.

[11] Auxiliary Task Graph Convolution Network: A skeleton-based action recognition for practical use (AT-GCN), 2025. Applied Sciences, 15(1), p. 198.

[12] Vemulapalli, R., Arrate, F., Chellappa, R., 2014. Human action recognition by representing 3D skeletons as points in a Lie group. In: CVPR. pp. 588–595.

[13] Fernando, B., Gavves, E., Oramas, J.M., Ghodrati, A., Tuytelaars, T., 2015. Modeling video evolution for action recognition. In: CVPR. pp. 5378–5387.

[14] Wang, L., et al., 2022. Skeleton graph-neural-network-based human action recognition: A survey. Sensors, 22(6).

[15] Du, Y., Wang, W., Wang, L., 2015. Hierarchical recurrent neural network for skeleton based action recognition. In: CVPR. pp. 1110–1118.

[16] A survey on 3D skeleton-based action recognition using learning methods, 2024. Sensors.

[17] Plizzari, C., Cannici, M., Matteucci, M., 2021. Skeleton-based action recognition via spatial and temporal transformer networks. Computer Vision and Image Understanding, 208.

[18] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: ICLR.

[19] Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. In: ICML. pp. 2014–2023.

[20] Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H., 2020. Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition. In: ACM MM. pp. 55–63.

[21] Nakamura, I., 2024. Multi-scale spatial-temporal self-attention graph convolutional networks for skeleton-based action recognition. arXiv preprint arXiv:2404.02624.

[22] Dynamic graph attention network for skeleton-based action recognition (DGAN), 2025. Applied Sciences, 15(9), p. 4929.

[23] Gao, Y., Duan, X., Dai, Q., 2025. Skeleton-based action recognition using graph convolutional network with pose correction and channel topology refinement (PCCTR-GCN). Computers, Materials & Continua, 83(1), pp. 701–718.

[24] Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W., 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In: CVPR. pp. 143–152.

[25] Skeleton action recognition via graph convolutional network with self-attention module (SA-GCN), 2024. Electronic Research Archive.

[26] Shuvo, M.R., Mekala, M.S., Elyan, E., 2025. Deep learning and attention-based methods for human activity recognition and anticipation: A comprehensive review. Cognitive Computation, 17(6), pp. 1–28.

[27] Shi, L., Zhang, Y., Cheng, J., Lu, H., 2020. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Transactions on Image Processing, 29, pp. 9532–9545.

[28] Chen, E., Bai, X., Gao, L., Tinega, H.C., Ding, Y., 2019. A spatiotemporal heterogeneous two-stream network for action recognition. IEEE Access, 7, pp. 57267–57275.

[29] Liu, J., Shahroudy, A., Lisboa Perez, M., Wang, G., Duan, L.-Y., Wong, A.K.C., 2019. NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[30] Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.-C., 2014. Cross-view action modeling, learning and recognition. In: CVPR. pp. 2649–2656.

[31] Liu, X., Li, Y., Xia, R., 2021. Adaptive multi-view graph convolutional networks for skeleton-based action recognition. Neurocomputing, 444, pp. 288–300.

[32] Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H., 2020. Skeleton-based action recognition with Shift Graph Convolutional Network. In: CVPR. pp. 183–192.

[33] Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H., 2020. Skeleton-based action recognition with Shift Graph Convolutional Network. In: CVPR. pp. 183–192.

[34] Xu, K., et al., 2022. Topology-aware convolutional neural network for efficient skeleton-based action recognition. In: AAAI, 36(3).

[35] Wu, L., Zhang, C., Zou, Y., 2022. SpatioTemporal focus for skeleton-based action recognition. arXiv preprint arXiv:2203.16767.

[36] Song, Y.-F., Zhang, Z., Shan, C., Wang, L., 2023. Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), pp. 1474–1488.

[37] Liu, Y., Zhang, H., Li, Y., He, K., Xu, D., 2023. Skeleton-based human action recognition via large-kernel attention graph convolutional network. IEEE Transactions on Visualization and Computer Graphics, 29(12), pp. 2575–2585.

[38] Yang, Z., Wang, Y., Liu, J., Wang, Z., 2023. Global spatio-temporal synergistic topology learning for skeleton-based action recognition. Pattern Recognition, 140, p. 109540.

[39] Mehmood, F., Guo, X., Chen, E., Akbar, M.A., Khan, A.A., Ullah, S., 2025. Extended multi-stream temporal-attention module for skeleton-based human action recognition. Computers in Human Behavior, 163, p. 108482.

[40] Chen, H., et al., 2026. Improving skeleton action recognition with channel-temporal attention and multi-stream feature aggregation. Multimedia Systems, 32(1), p. 8.

[41] Chen, H., Shen, Y., Zhang, Y., Pan, X., Yang, Q., 2025. Skeleton-based action recognition through dual-granularity feature fusion with self-adapting graph convolution and multi-scale temporal convolution. Neurocomputing, 639, p. 130261.

[42] Li, J., Ye, P., Xia, Y., Wang, Y., Cao, Y., 2025. Disentangled adaptive multi-dimensional dynamic graph convolutional network for skeleton-based action recognition. Neurocomputing, Art. no. 131693.

[43] Wu, C., Wu, X.-J., Xu, T., Kittler, J., 2025. Adaptive pooling with dual-stage fusion for skeleton-based action recognition. Neural Networks, Art. no. 107615.

[44] Li, C., Wen, C., Sun, P., Cui, Y., Hong, R., Niu, Z., Su, H., 2026. TE-STGCN: Topology enhanced spatio-temporal graph convolutional network for skeleton-based action recognition. Applied Soft Computing, Art. no. 114835.

[45] Xia, Y., Gao, Q., Wu, W., Cao, Y., 2024. Skeleton-based action recognition based on multidimensional adaptive dynamic temporal graph convolutional network. Engineering Applications of Artificial Intelligence, 127, p. 107210.

[46] Zhu, Y., Shuai, H., Liu, G., Liu, Q., 2022. Multilevel spatial–temporal excited graph network for skeleton-based action recognition. IEEE Transactions on Image Processing, 32, pp. 496–508.

[47] Chi, H.-G., Ha, M.H., Chi, S., Lee, S.W., Huang, Q., Ramani, K., 2022. InfoGCN: Representation learning for human skeleton-based action recognition. In: CVPR. pp. 20186–20196.

[48] Zhou, H., Liu, Q., Wang, Y., 2023. Learning discriminative representations for skeleton based action recognition. In: CVPR. pp. 10608–10617.

[49] Cui, H., Hayama, T., 2024. Joint-partition group attention for skeleton-based action recognition. Signal Processing, 224, p. 109592.

[50] Li, C., Huang, Q., Mao, Y., Li, X., Wu, J., 2024. Multi-granular spatial-temporal synchronous graph convolutional

network for robust action recognition. Expert Systems with Applications, 257, p. 124980.

[51] Song, Y.-F., Zhang, Z., Shan, C., Wang, L., 2022. Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), pp. 1474–1488.

[52] Wu, L., Zhang, C., Zou, Y., 2023. Spatiotemporal focus for skeleton-based action recognition. Pattern Recognition, 136, p. 109231.

[53] Chen, Z., Li, S., Yang, B., Li, Q., Liu, H., 2021. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In: AAAI, 35, pp. 1113–1122.

[54] Xiong, X., Min, W., Wang, Q., Zha, C., 2022. Human skeleton feature optimizer and adaptive structure enhancement graph convolution network for action recognition. IEEE Transactions on Circuits and Systems for Video Technology, 33(1), pp. 342–353.

[55] Li, C., Huang, Q., Mao, Y., 2023. DD-GCN: Directed diffusion graph convolutional network for skeleton-based human action recognition. In: ICME. pp. 786–791.

[56] Shi, L., Zhang, Y., Cheng, J., Lu, H., 2021. AdaSGN: Adapting joint number and model size for efficient skeleton-based action recognition. In: ICCV. pp. 13413–13422.

[57] Hu, Z., Pan, Z., Wang, Q., Yu, L., Fei, S., 2022. Forward-reverse adaptive graph convolutional networks for skeleton-based action recognition. Neurocomputing, 492, pp. 624–636.

[58] Liu, Y., Zhang, H., Xu, D., He, K., 2022. Graph transformer network with temporal kernel attention for skeleton-based action recognition. Knowledge-Based Systems, 240, p. 108146.

[59] Miao, S., Hou, Y., Gao, Z., Xu, M., Li, W., 2021. A central difference graph convolutional operator for skeleton-based action recognition. IEEE Transactions on Circuits and Systems for Video Technology, 32(7), pp. 4893–4899.

[60] Yun, X., Xu, C., Riou, K., Dong, K., Sun, Y., Li, S., et al., 2024. Behavioral recognition of skeletal data based on targeted dual fusion strategy. In: AAAI, 38, pp. 6917–6925.

[61] Abdelfattah, M., Hassan, M., Alahi, A., 2024. MaskCLR: Attention-guided contrastive learning for robust action representation learning. In: CVPR. pp. 18678–18868.