GCN-Mamba: A Semantic-guided Graph Convolutional Network with Mamba State Space Models for Skeleton-based Action Recognition

Amine Mansouri[1,*], Abdellah Elzaar[2], Toufik Bakir[2], and Smain Femmam[3]
[1] ISAT-DRIVE UR 1859 Laboratory, Université Bourgogne Europe, Nevers, France
[2] ImViA UR 7535 Laboratory, Université Bourgogne Europe, Dijon, France
[3] Networks & Communications Department, Faculty of Sciences, Haute-Alsace University UHA, Mulhouse, France
Email: Amine.Mansouri@ube.fr (A.M.); Abdellah.El-Zaar@ube.fr (A.E.); toufik.bakir@ube.fr (T.B.); smain.femmam@uha.fr (S.F.)
[*]Corresponding author

***Abstract*—Human Action Recognition (HAR) has seen significant advancements with Graph Convolutional Networks (GCNs), which effectively model skeletal motion dynamics. In this work, we propose a novel HAR framework that integrates GCNs with an Adaptive Adjacency Matrix for spatial modeling and the Mamba State Space Model (SSM) for temporal feature extraction. This hybrid approach aims to balance accuracy and efficiency. It does so by leveraging the structural expressiveness of GCNs for spatial modeling, while harnessing the sequential modeling power of Mamba-SSM for temporal dynamics. The central goal of this paper is to evaluate GCN-Mamba against our own previous model, ImpSGN (Improved Semantic-Guided Network), with a specific focus on reducing model complexity while preserving recognition accuracy. GCN-Mamba achieves competitive performance with an approximately 72% reduction in parameter count (from 4.0 M to 1.1 M), making it a lightweight yet effective architecture. Our findings highlight the trade-offs between accuracy and efficiency in HAR models and demonstrate the potential of state-space modeling in skeletal action recognition. The code is publicly available on GitHub[1].**

***Keywords*—deep learning, Human Action Recognition (HAR), Graph Convolutional Networks (GCNs), Mamba State Space Models (Mamba-SSM)**

I. Introduction

Human Action Recognition (HAR) plays a crucial role in various applications, including surveillance, healthcare monitoring, and human-computer interaction [1]. Among the different modalities used for HAR, skeleton-based methods have gained significant attention due to their robustness against background clutter and viewpoint variations.

Graph Convolutional Networks (GCNs) have proven highly effective in modeling skeletal dynamics for Human Action Recognition (HAR) by capturing spatial dependencies between joints [2] [3]. Unlike traditional convolutional or recurrent approaches, GCNs leverage the natural graph structure of the human skeleton, allowing the model to learn meaningful representations of motion patterns and interactions between body parts [4]. However, conventional GCNs often rely on static or predefined adjacency matrices, which may not fully capture the dynamic nature of skeletal movement. To address this limitation, recent advances have explored adaptive adjacency matrices [5], enabling the model to learn and update the connectivity of joints based on the observed motion, thereby enhancing flexibility and robustness in spatial modeling. In GCN-Mamba, this adaptation is computed per frame and per input sequence: the edge weight between any two joints is derived directly from their feature representations at each time step, meaning the graph topology is re-estimated dynamically at inference time rather than being fixed during training or shared across samples.

At the same time, temporal modeling in HAR has evolved beyond recurrent architectures like LSTMs and GRUs, which suffer from vanishing gradients and high computational costs. Transformers have emerged as a powerful alternative, but their quadratic complexity limits efficiency, particularly in real-time applications. Recently, State Space Models (SSMs), such as Mamba-SSM [6] [7], have gained traction due to their ability to model long-range dependencies with linear complexity. Mamba-SSM leverages selective state-space filtering and gating mechanisms, making it highly efficient for sequential tasks like HAR. By integrating GCNs with an adaptive adjacency matrix for spatial learning and Mamba-SSM for temporal sequence modeling, our approach aims to strike a balance between accuracy and efficiency, addressing key challenges in HAR while significantly reducing computational overhead.

In this work, we propose a novel HAR framework that combines GCNs with Adaptive Adjacency Matrices for spatial modeling and Mamba-SSM for temporal modeling. Unlike traditional GCNs, our approach dynamically learns

Manuscript received January 29, 2026; revised February 25, 2026; accepted April 2, 2026; published July 10, 2026.

[1]https://github.com/acvai/GCN-Mamba

doi: 10.12720/jait.17.7.1269-1277

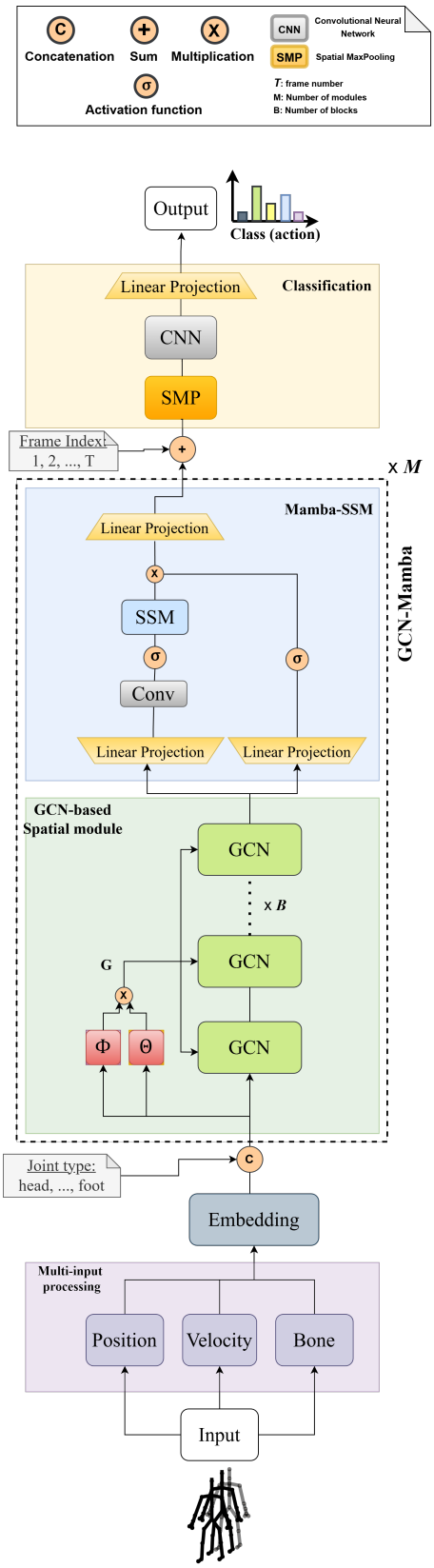

Fig. 1: Architecture of the proposed end-to-end GCN-Mamba model, featuring a single stream composed of multiple spatio-temporal GCN-Mamba blocks. The *Embed* module captures joint dynamics by combining the position, velocity and bone information.

the connectivity between joints, improving adaptability. Furthermore, Mamba-SSM provides an efficient alternative to transformers for capturing long-term temporal dependencies in human motion sequences. The primary objective of this paper is to benchmark GCN-Mamba directly against our prior model, ImpSGN, and to quantify the accuracy–efficiency trade-off. While GCN-Mamba achieves 0.1% lower accuracy on CS and 0.4% lower on CV compared to ImpSGN, it reduces the parameter count by approximately 72% (from 4.0 M to 1.1 M). This trade-off is practically significant: in real-world HAR deployments such as wearable health monitors, embedded surveillance systems, and mobile human-computer interaction, model size and inference latency are often the binding constraints. A model that preserves over 99% of the accuracy of its predecessor while cutting memory requirements by nearly three quarters is far more deployable in such settings, without requiring specialised hardware or model compression post-processing.

The key contributions of this paper are summarized as follows:

- We introduce a novel HAR framework that integrates GCNs with Adaptive Adjacency Matrices for dynamic spatial modeling and Mamba-SSM for efficient temporal modeling.
- We conduct a direct comparative analysis between GCN-Mamba and our previous model ImpSGN, yielding three concrete insights: (i) replacing a CNN-based temporal module with Mamba-SSM achieves approximately 72% parameter reduction (from 4.0 M to 1.1 M) at negligible accuracy cost (-0.1% CS, -0.4% CV); (ii) identical FLOPs and inference latency confirm that this reduction comes from architectural compactness rather than computational shortcuts; and (iii) the performance gap closes further on the larger NTU RGB+D 120 benchmark, suggesting that Mamba-SSM's long-range temporal modeling scales better with dataset complexity than the CNN-based alternative.
- We demonstrate that GCN-Mamba's compact architecture makes it a viable solution for deployment in resource-constrained environments—such as embedded systems, wearable devices, and mobile platforms—where the 4.0 M-parameter ImpSGN would be impractical, without requiring model compression or hardware acceleration.

The remainder of this paper is organized as follows. Section II reviews related work on GCN-based HAR and state-space sequence models. Section III presents the proposed GCN-Mamba architecture in detail, describing the multi-input processing, embedding module, adaptive GCN with semantic integration, and Mamba-SSM temporal module. The experimental setup, datasets, and comparative results on NTU RGB+D 60 and NTU RGB+D 120 are reported in Section IV, including the ablation study and efficiency analysis. Section V concludes the paper and outlines directions for future work.

II. Related Work

Human Action Recognition (HAR) has been extensively studied using both handcrafted feature-based approaches and deep learning techniques. Recent advancements primarily focus on learning spatial and temporal representations from skeletal data, where Graph Convolutional Networks (GCNs) and Sequence Models have played a pivotal role.

Graph Convolutional Networks in HAR: GCNs have demonstrated significant success in HAR by efficiently modeling the skeletal structure as a graph, where joints act as nodes and bones as edges. Early works, such as ST-GCN [4], [8], leveraged predefined adjacency matrices to encode spatial relationships. However, fixed topologies limit adaptability to dynamic movements. To overcome this, adaptive adjacency matrices have been introduced, allowing the model to learn spatial connections dynamically based on movement patterns [9]. Recent approaches, including EfficientGCN [10] and MSG3D [11], further optimized spatial modeling by incorporating multi-scale graph operations. Our prior work, ImpSGN, improved upon these strategies by integrating semantic priors into GCN-based spatial modeling, enhancing interpretability and robustness.

Advances in Sequence Modeling for HAR: Traditional temporal modeling in HAR primarily relied on Recurrent Neural Networks (RNNs), including LSTMs and GRUs, to capture motion dynamics [12]. However, these models suffer from gradient-related issues and inefficiencies when handling long-range dependencies. Transformers emerged as an alternative, offering superior temporal modeling via self-attention mechanisms [13]. Nonetheless, their quadratic complexity in sequence length remains a bottleneck for real-time applications. Recently, State Space Models (SSMs), such as Mamba-SSM [6], have gained attention due to their ability to model long-range dependencies with linear complexity. Unlike self-attention, SSMs rely on selective gating and efficient recurrence, making them computationally attractive for sequence-based tasks. Recent works have further explored the landscape of HAR methods from both application and survey perspectives, providing broader context on the evolution of deep learning techniques in this field [14], [15].

Integrating GCNs with SSMs for HAR: While previous research has explored GCNs for spatial modeling and sequence models for temporal learning [16], limited work has investigated the synergy between GCNs with adaptive adjacency matrices and Mamba-SSM for HAR. Our proposed approach bridges this gap by combining the strengths of learnable graph structures with efficient sequence modeling. This hybrid framework aims to balance accuracy and efficiency, reducing parameter overhead while maintaining competitive performance.

III. Semantic-Guided Network with Mamba State Space Models

In this section, we present GCN-Mamba, a semantic-guided Graph Convolutional Network (GCN) enhanced with Mamba State Space Models (SSM), as illustrated in Fig. 1. This model integrates semantic information with spatial-temporal modules to improve the understanding of human body dynamics. As an extension of ImpSGN, GCN-Mamba refines spatio-temporal feature extraction through specialized GCN-Mamba blocks, enhancing action recognition performance.

The proposed architecture processes multi-modal skeletal inputs (including position, velocity, and bone data) offering a more comprehensive representation of motion patterns. The backbone consists of a sequence of GCN-Mamba blocks that selectively emphasize spatial and temporal features, refining them before passing the extracted representations to a classification module for final predictions.

Trained in an end-to-end manner, GCN-Mamba achieves performance comparable to ImpSGN while significantly reducing model complexity. With a lightweight design and efficient computation, it delivers strong accuracy while being well-suited for deployment in resource-constrained environments.

A. Multi-input Processing

Previous studies [5], [17] emphasize that preprocessing is crucial for skeleton-based action recognition. In this work, we group the input features into three main types: joint positions, motion velocities, and finally the bone features.

A 3D action sequence is represented as $P = p \in \mathbb{R}^{J_{in} \times T_{in} \times K_{in}}$, where K_{in} is the coordinate dimension (x, y, z), J_{in} is the number of joints (e.g., foot, head), and T_{in} is the number of frames. Relative joint positions, denoted by P_{rel}, are calculated with respect to the center spine joint (c):

$$p_{rel_i} = s[i, :, :] - s[c, :, :], \tag{1}$$

where $s[i, :, :]$ and $s[c, :, :]$ are the coordinates of the i-th joint and the center spine joint. The joint position input is then obtained by combining P and P_{rel}.

Motion velocities describe how joints move between frames and are obtained by:

$$\begin{aligned} f_{\text{vel},t} &= s[:, t+2, :] - s[:, t, :], \\ v_{\text{vel},t} &= s[:, t+1, :] - s[:, t, :], \\ f_{\text{pvel},t} &= f_{\text{vel},t} \odot \text{PrimeVec}, \\ v_{\text{pvel},t} &= v_{\text{vel},t} \odot \text{PrimeVec}. \end{aligned} \tag{2}$$

Here, f_{vel} and v_{vel} denote fast and slow velocities, while f_{pvel} and v_{pvel} are the same velocities scaled element-wise ($\odot$) by PrimeVec, a vector of the first J_{in} prime numbers.

Bone features are represented by lengths $Len = len_i | i = 1, \cdots, J_{in}$ and angles $Ang = ang_i | i = 1, \cdots, J_{in}$, computed as:

$$\begin{aligned} len_i &= s[i, :, :] - s[i_{adj}, :, :], \\ ang_{i,w} &= \arccos\left(\frac{len_{i,w}}{\sqrt{len_{i,x}^2 + len_{i,y}^2 + len_{i,z}^2}}\right). \end{aligned} \tag{3}$$

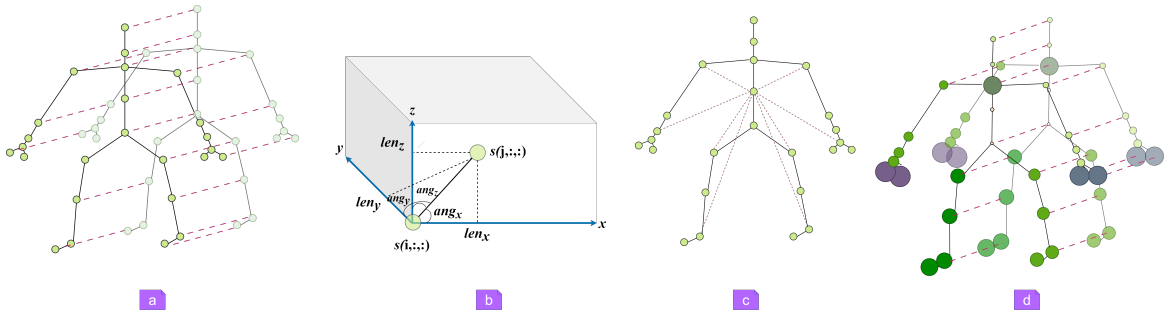

Fig. 2: The input data in this visualization illustrates four feature types: (a) the motion velocities, (b) the 3D bone angles and lengths, (c) the relative joint positions, and (d) the prime-scaled velocities. (Best viewed in color.)

where i_{adj} is the neighboring joint of the i-th joint, and w corresponds to one of the 3D coordinates (x, y, z). An illustration is provided in Fig. 2.

B. Embedding

In a 3D coordinate system, a joint is described by three main inputs: its position P (original and relative), its velocities (including fast f_{vel}, slow v_{vel}, and their prime-scaled versions f_{pvel}, v_{pvel}), and its bone features B (lengths Len and angles Ang). These inputs are encoded into a shared high-dimensional space H_{j,t,k_1}, where the embeddings of position $\tilde{P}$, velocity $\tilde{Vel}$, and bone features $\tilde{B}$ are combined by summation:

$$H_{j,t,k_1} = \tilde{P}_{j,t,k_1} + \tilde{Vel}_{j,t,k_1} + \tilde{B}_{j,t,k_1}, \tag{4}$$

where, k_1 is the dimension of the joint representation in this high-dimensional space. For example, the position embedding $\tilde{P}$ is produced using two fully connected (FC) layers:

$$\begin{aligned} P_{inner_{j,t,k'}} &= \text{ReLU}(W_1 P_{j,t,k} + b_1), \\ \tilde{P}_{j,t,k_1} &= \text{ReLU}(W_2 P_{inner_{j,t,k'}} + b_2). \end{aligned} \tag{5}$$

here $W_1 \in \mathbb{R}^{k' \times (3\times 2)}$ and $W_2 \in \mathbb{R}^{k_1 \times k'}$ are the weight matrices that map the input through an intermediate size k' and then to the target dimension k_1. Bias vectors b_1, b_2 are added, and $\text{ReLU}(\cdot)$ is the rectified linear activation. Velocity and bone features are embedded in the same way.

C. Adaptive GCN with Semantic Integration

To effectively capture the correlations among joints in skeletal data, it is essential to employ Graph Convolutional Networks (GCNs). Traditional GCN-based approaches define joints as nodes and pre-establish graph connections (edges) using prior knowledge [4]. More recently, adaptive graph methods have emerged, allowing content-driven learning of graph structures [9]. In this work, we adopt the latter approach while integrating joint-type semantics, similar to [5], to enhance feature learning. We represent a skeleton frame as a graph with V nodes, corresponding to V joints. Each joint type j at frame t is expressed as a joint representation $Z_{j,t,k}$, encapsulating both dynamic and semantic characteristics. The joint representation consists of a series of elements, denoted as $z_{1,t,k}$ through $z_{V,t,k}$, formulated as $\{Z_{j,t,k} = (z_{1,t,k}, ..., z_{V,t,k})\}$.

Following previous works [9], [18], [19], in the embedded space, the edge weight between the i^{th} and j^{th} joints within the same frame is computed based on their feature similarity:

$$A_t(i,j) = \theta(z_{i,t})^T \phi(z_{j,t}), \tag{6}$$

where θ and ϕ are transformation functions implemented using fully connected (FC) layers: $\theta(x) = W_3 x + b_3$ and $\phi(x) = W_4 x + b_4$.

By computing affinities across all joint pairs, we derive an adjacency matrix $A_t \in \mathbb{R}^{V\times V}$. To ensure stability and numerical robustness, we apply row-wise SoftMax normalization, following prior works [19], [20], yielding the normalized adjacency matrix G_t. This ensures that the sum of all edge weights connected to any given node equals 1. Message passing is then conducted via a residual graph convolutional layer:

$$\begin{aligned} Y_t &= G_t Z_t W_y, \\ Z_t^* &= Y_t + Z_t W_z. \end{aligned} \tag{7}$$

Here, W_y and W_z are trainable transformation matrices, shared across temporal frames to maintain parameter efficiency. The output representation Z_t^* can be further refined by stacking multiple residual GCN layers while preserving the adjacency structure G_t.

To further enhance feature expressiveness, we incorporate semantic information related to the frame index. Using the same encoding approach as in Equation 5, we derive the frame index embedding $\tilde{f}_{j,t,k_2} \in \mathbb{R}^{k_2}$. The joint representation is then enriched by integrating this semantic information:

$$\hat{z}_{j,t} = z^*_{j,t} + \tilde{f}_{j,t,k_2}, \quad \hat{z}_{j,t} \in \mathbb{R}^{k_2}. \tag{8}$$

where $\hat{z}_{j,t}$ denotes the final semantically enriched joint representation.

This semantic augmentation allows the model to distinguish temporal variations more effectively while maintaining a structured spatial representation. Finally, we can visualize the skeleton-based action recognition in the form of attention maps, as shown in Fig. 3.

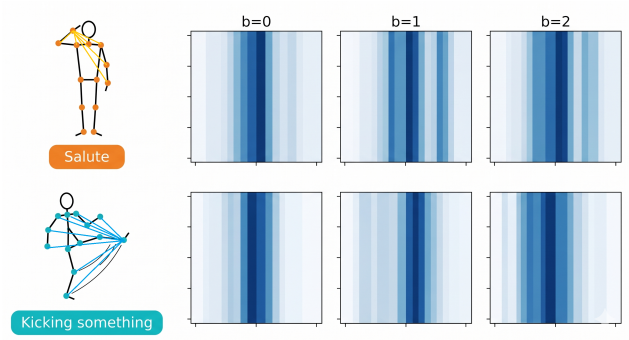

Fig. 3: Evolution of joint-specific attention across successive GCN-Mamba blocks. The visualization shows how the model's focus shifts and sharpens on different skeleton joints. Attention magnitude is depicted by circle size, and a continuous color map (light to dark) encodes the precise attention weights.

D. Mamba State Space Model for Temporal Modeling

To enhance temporal modeling in Human Action Recognition (HAR), we integrate the Mamba State Space Model (SSM) into our framework. Traditional recurrent architectures like LSTMs and GRUs suffer from vanishing gradients and inefficient long-range dependencies, while Transformer-based approaches incur quadratic complexity due to self-attention. To address these limitations, we adopt Mamba-SSM, which achieves linear complexity while effectively capturing long-range dependencies using state-space transitions.

The Mamba model propagates temporal information through the following discrete-time state-space formulation:

$$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t, \quad y_t = \mathbf{C}x_t, \tag{9}$$

where x_t represents the latent state, u_t is the input skeletal feature, and y_t is the output representation. The learned matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ govern state transitions and feature transformations, allowing for an efficient and compact representation of temporal dependencies.

To further refine sequence modeling, Mamba incorporates selective gating, dynamically filtering relevant features through the update rule:

$$h_t = \sigma(\mathbf{W}_f u_t) \odot (\mathbf{A}h_{t-1} + \mathbf{W}_u u_t), \tag{10}$$

where $\mathbf{W}_f$ and $\mathbf{W}_u$ are learnable weight matrices, and σ is the sigmoid gating function (distinct from the ReLU activation used in the embedding layers). This mechanism enables Mamba to adaptively suppress redundant information, improving robustness against noise in motion sequences.

By replacing traditional attention mechanisms with efficient state-space modeling, Mamba-SSM offers several advantages:

- **Linear time complexity** ($\mathcal{O}(T)$) instead of quadratic ($\mathcal{O}(T^2)$).
- **Stronger long-range dependencies** with optimized state propagation.
- **Lower memory usage** and improved scalability for variable-length skeletal motion data.

In our framework, GCNs first extract spatial dependencies using an adaptive adjacency matrix, and the resulting joint features are processed by Mamba-SSM for temporal modeling. This hybrid approach, combining *graph-based spatial learning* with *state-space-driven temporal modeling*, ensures efficient and accurate action recognition while significantly reducing parameter count.

Advantages Beyond Computational Efficiency: Although Mamba-SSM is often highlighted for its linear computational complexity, its advantages in our framework extend significantly beyond efficiency considerations. First, the selective state-space mechanism enables adaptive propagation of information across the entire temporal sequence, effectively providing a global receptive field. This allows the model to capture long-range temporal dependencies that are difficult to model with fixed-kernel temporal convolutions. Second, unlike attention-based mechanisms that incur quadratic memory and computational growth with sequence length, Mamba preserves linear complexity while still maintaining global context modeling capability. This makes it particularly suitable for long skeleton sequences without the instability or memory overhead associated with self-attention. Finally, the continuous-time state-space formulation underlying Mamba provides a principled and stable mechanism for long-horizon sequence modeling. Such stability is crucial in skeleton-based action recognition, where discriminative motion patterns may span extended temporal intervals. Therefore, the integration of Mamba-SSM in GCN-Mamba is motivated not only by efficiency, but also by its superior capability to model global, long-range, and stable temporal dynamics.

Comparison with Transformer-Based Temporal Models: To put these advantages in concrete terms, consider a sequence of T frames processed by a self-attention Transformer versus Mamba-SSM. A Transformer computes a full $T \times T$ attention matrix, resulting in $\mathcal{O}(T^2)$ time complexity and $\mathcal{O}(T^2)$ memory growth. For $T = 20$ frames (as used in our setting), this remains manageable; however, the quadratic growth becomes a hard constraint in longer sequences or higher-frequency inputs encountered in real-time deployment. Mamba-SSM, by contrast, processes the sequence recurrently through a fixed-size latent state x_t, maintaining $\mathcal{O}(T)$ time and $\mathcal{O}(1)$ memory with respect to sequence length — the state size is constant regardless of how many frames have been processed. In practice, this translates to lower peak memory consumption during inference, which is critical for deployment on memory-constrained hardware such as embedded processors or wearable devices. Furthermore, because Mamba-SSM does not require the full sequence to be held in memory simultaneously (unlike attention, which must attend over all past tokens), it supports streaming inference naturally — processing each new frame as it ar-

TABLE I: Performance comparison of GCN-Mamba with state-of-the-art methods on NTU 60 (CS, CV benchmarks).

Method	CS (%)	CV (%)
HBRNN-L	59.1	64.0
Part-Aware LSTM	62.9	70.3
ST-LSTM + Trust Gate	69.2	77.7
STA-LSTM	73.4	81.2
GCA-LSTM	74.4	82.8
Clips+CNN+MTLN	79.6	84.8
VA-LSTM	79.4	87.6
ElAtt-GRU	80.7	88.4
ST-GCN	81.5	88.3
DPRL+GCNN	83.5	89.8
SR-TSL	84.8	92.4
HCN	86.5	91.1
AGC-LSTM (joint)	87.5	93.5
AS-GCN	86.8	94.2
GR-GCN	87.5	94.3
1s Shift-GCN	87.8	**95.1**
2s-AGCN	88.5	**95.1**
VA-CNN	88.7	94.3
SGN	89.0	94.5
SAGN	89.2	94.2
ImpSGN (previous work)	**89.7**	95.0
GCN-Mamba (ours)	89.6	94.6

TABLE II: Performance comparison of GCN-Mamba with state-of-the-art methods NTU 120 (C-Subject and C-Setup).

Method	C-Subject (%)	C-Setup (%)
Part-Aware LSTM	25.5	26.3
ST-LSTM + Trust Gate	55.7	57.9
GCA-LSTM	58.3	59.2
Clips+CNN+MTLN	58.4	57.9
Two-Stream GCA-LSTM	61.2	63.3
RotClips+MTCNN	62.2	61.8
Body Pose Evolution Map	64.6	66.9
SGN	79.2	81.5
1s Shift-GCN	80.9	83.2
SAGN	**82.1**	**83.8**
GCN-Mamba (ours)	**82.1**	**83.8**

rives without recomputation. This property is particularly valuable for real-time HAR applications where frames are received continuously and low-latency response is required. The 1.1ms inference latency reported in Table III reflects these properties: GCN-Mamba matches the latency of ImpSGN despite replacing its temporal module with one capable of modeling longer temporal dependencies.

IV. Experiments

In this section, we assess the performance of our proposed model through extensive evaluations on the large-scale NTU RGB+D 60 and NTU RGB+D 120 datasets [21], [22]. The results, presented in Tables I and II, compare our approach against a range of existing methods, including CNN-based, LSTM-based, and GCN-based architectures, highlighting the advancements brought by our model.

A. Datasets

1) NTU RGB+D 60: In this work, we employ the NTU RGB+D 60 dataset, a large indoor collection introduced by [21]. The recordings were made using Microsoft Kinect v2 sensors, which capture four modalities: RGB videos, depth maps, infrared (IR), and 3D skeletons. For our experiments, only the skeleton data are used. The dataset contains 56,880 action sequences performed by 40 participants, covering 60 action categories. These categories are organized into three groups: health-related actions, daily activities, and interactive actions. Each skeleton is represented by 25 joints, described by their 3D coordinates (x, y, z). To evaluate models, two benchmark splits are provided:

a) Cross-Subject (CS): The 40 actors are divided into training and testing groups, resulting in 40,091 training samples and 16,487 test samples.

b) Cross-View (CV): Sequences from cameras 2 and 3 (37,646 samples) are used for training, while those from camera 1 (18,932 samples) form the test set.

2) NTU RGB+D 120: The second dataset used in this study is NTU RGB+D 120, which is currently the largest indoor dataset for skeleton-based human action recognition (HAR) [22]. It extends NTU 60, increasing the total number of videos to 114,480, performed by 106 subjects. These subjects are evenly split into training and testing groups (53 in each). The action categories have also been expanded from 60 to 120. Two official evaluation protocols are provided:

a) Cross-Subject (CSub): The participants are split into two groups, one for training (63,026 sequences) and the other for testing (50,919 sequences).

b) Cross-Setup (CSet): The division is based on different camera viewpoints (distance and height), resulting in 54,468 training samples and 59,477 test samples.

It should be noted, following [22], that 535 sequences are missing in the dataset and must be excluded from all experiments.

TABLE III: Model Complexity and Efficiency Comparison.

Method	CS (%)	CV (%)	Params (M)	FLOPs (G)	Inference (ms)
ImpSGN (previous work)	89.7	95.0	4.0	1.2	1.1
GCN-Mamba (ours)	89.6	94.6	1.1	1.2	1.1

B. Implementation Details

1) Network settings: In our design, we use an embedding dimension of 64, with independent weights assigned for velocity, position, and bone distance features (i.e., no weight sharing). The skeleton input has the shape *N*: batch size, *C*: number of channels, *V (or J)*: 25 joints, *T*: 20 frames. Initially, the input channels are *C*=3, corresponding to the x, y, and z coordinates. After applying multi-input processing, the number of channels increases to $C = 3 \times 2$, since two sets are generated for each feature type. The adaptive adjacency matrix G_t is defined with size 25×25. The GCN-Mamba block is configured with an input dimension of 128 and an output dimension of 512. For the classification stage, the first convolutional (CNN) layer is set with 512 filters and a temporal kernel size of 3, while the second CNN layer uses 512 filters with a kernel size of 1. Throughout the network, we employ batch normalization, dropout, and nonlinear activations including Softmax, ReLU, and Hardswish.

2) Training settings: The model was trained using the PyTorch framework on a single Tesla V100s GPU provided by the university cluster (CCUB). We adopted the Adam optimizer with an initial learning rate of 0.001. The learning rate was reduced by a factor of 10 at epochs 60, 100, and 120, and training was completed after 140 epochs. To maintain compatibility across GPUs, the batch size was fixed at 64. For classification, we used cross-entropy as the loss function. In addition, label smoothing with a factor of 0.1 was applied to improve generalization during training.

C. Comparison with Existing Methods

In this section, we compare GCN-Mamba with published methods on the NTU RGB+D 60 and NTU RGB+D 120 datasets. Results are summarised in Tables I and II. The primary axis of comparison is our own prior model, ImpSGN; results against the broader literature serve to contextualise GCN-Mamba within its architectural family.

Table I shows that GCN-Mamba improves over the original SGN model [5] by 0.6% on CS and 0.1% on CV. This improvement stems directly from replacing the fixed skeleton graph of SGN with an adaptive adjacency matrix, which allows the model to learn data-driven joint correlations rather than relying on predefined anatomical connections, and from substituting the CNN-based temporal module with Mamba-SSM, which captures longer-range dependencies at linear rather than quadratic complexity. When compared to prominent CNN-based and RNN-based methods such as *Clips+CNN+MTLN* [23] and *ElAtt-GRU* [24], GCN-Mamba achieves 10% and 8.9% higher accuracy on the CS benchmark, respectively. These gains reflect the fundamental limitation of CNN and RNN architectures in jointly modeling spatial graph structure and long-range temporal dynamics — a limitation that the GCN-Mamba spatio-temporal block is specifically designed to address.

GCN-Mamba demonstrates superior performance over approaches such as *SR-TSL* [17] and *ST-GCN* [4], both of which rely on fixed graph topologies for spatial modeling. This advantage is consistently observed across both the CS and CV benchmarks, highlighting the benefit of adaptive graph topology learning. Compared to ImpSGN, GCN-Mamba achieves 0.1% lower CS accuracy and 0.4% lower CV accuracy, a marginal deficit that must be read alongside Table III: GCN-Mamba reduces the parameter count by approximately 72% (from 4.0M to 1.1M) while maintaining identical FLOPs (1.2G) and inference latency (1.1ms). This confirms that the Mamba-SSM temporal module replaces a heavier architecture at no computational cost, with only negligible impact on accuracy.

On the NTU RGB+D 120 dataset, GCN-Mamba matches the best results in our comparison, tying with SAGN at 82.1% (C-Subject) and 83.8% (C-Setup), as shown in Table II. Relative to the original SGN model [5], GCN-Mamba achieves improvements of $+2.9\%$ on C-Subject (79.2% $\rightarrow$ 82.1%) and $+2.3\%$ on C-Setup (81.5% $\rightarrow$ 83.8%). This result is particularly meaningful: it demonstrates that the accuracy–efficiency trade-off observed on NTU RGB+D 60 generalises to a dataset with twice the number of action categories and a significantly larger subject pool, suggesting that the adaptive graph and Mamba-SSM components scale well with dataset complexity.

D. Ablation Study

To quantify the individual contributions of the two major components of GCN-Mamba — the adaptive adjacency matrix and the Mamba-SSM temporal module — we conduct a systematic ablation study on the NTU RGB+D 60 dataset under the Cross-Subject (CS) and Cross-View (CV) protocols. We define five model configurations, summarized in Table IV, by selectively enabling or disabling each module.

Impact of the adjacency matrix strategy: Comparing States 1 and 2 isolates the effect of replacing the predefined skeleton graph (ST-GCN style [4]) with our learned adaptive adjacency matrix. The adaptive variant achieves gains of $+1.19\%$ on CS and $+0.79\%$ on CV, confirming that data-driven topology learning captures joint relationships more effectively than a fixed anatomical graph. This improvement is consistent across both single-module (States 1 vs. 2) and full-model (States 4 vs. 5) settings, where the adaptive adjacency matrix provides a $+1.15\%$ CS and $+0.43\%$ CV advantage when combined with Mamba-SSM.

Impact of the Mamba-SSM temporal module: State 3, which uses only the Mamba-SSM module without any GCN spatial processing, yields significantly lower accuracy (CS: 76.36%, CV: 79.63%). This result demonstrates that Mamba-SSM alone, operating on raw joint sequences, lacks the structured spatial inductive bias needed for skeleton-based HAR, and therefore requires GCN-extracted spatial features as input to be effective. When Mamba-SSM is added to the predefined-adjacency GCN (State $1 \rightarrow 4$), accuracy increases by $+0.64\%$ CS and $+0.46\%$ CV. Similarly, adding Mamba-SSM to the adaptive GCN (State $2 \rightarrow 5$) yields a further gain of $+0.6\%$ CS and $+0.1\%$ CV, establishing the full GCN-Mamba model (State 5) as the best-performing configuration.

Complementarity of both components: The ablation results reveal a clear complementarity between spatial and temporal modeling. The adaptive adjacency matrix is the dominant contributor to spatial accuracy, while Mamba-SSM provides consistent, additive temporal gains on top of it. Crucially, neither component alone reaches the accuracy of their combination: State 5 outperforms all partial configurations, confirming that the synergy between adaptive graph convolution and state-space temporal modeling is the key driver of GCN-Mamba's performance.

E. Comparison with ImpSGN and Discussion of Novelty

ImpSGN achieved strong performance on the NTU RGB+D benchmarks, reporting 89.7% (CS) and 95.0% (CV) on NTU 60. These results established ImpSGN as a

TABLE IV: Ablation study: contribution of each major component on NTU RGB+D 60. CS: Cross-Subject; CV: Cross-View. ✓indicates the component is active.

State	Predefined Adj.	Adaptive Adj.	SSM-Mamba	CS (%)	CV (%)
1	✓			87.81	93.71
2		✓		89.0	94.5
3			✓	76.36	79.63
4	✓		✓	88.45	94.17
5 (GCN-Mamba)		✓	✓	**89.6**	**94.6**

competitive and efficient skeleton-based action recognition model.

Our proposed GCN-Mamba further advances this line of research by introducing a Mamba-based sequential modeling mechanism within the graph convolutional framework. This design enhances long-range temporal dependency modeling while maintaining a lightweight architecture.

Performance on NTU 60: On NTU 60, GCN-Mamba achieves 89.6% (CS) and 94.6% (CV), which remains competitive with ImpSGN and other state-of-the-art graph-based methods. Although the accuracy is comparable, the key advancement of GCN-Mamba lies in its architectural efficiency and modeling capability rather than marginal accuracy gains. **Model Efficiency and Compactness:** A major contribution of GCN-Mamba is its reduced model complexity. Compared to ImpSGN (4.0M parameters), GCN-Mamba requires only 1.1M parameters (see Table III), representing a reduction of approximately 72% in model size. Despite this substantial compression, GCN-Mamba maintains competitive recognition performance.

Furthermore, the computational cost (FLOPs) remains comparable, while inference latency is preserved, demonstrating that the proposed Mamba-enhanced graph modeling improves parameter efficiency without sacrificing effectiveness.

Novelty and Contribution: Unlike ImpSGN, which relies on structured graph convolution and spatial grouping mechanisms, GCN-Mamba integrates selective state-space modeling to better capture long-range temporal dynamics. This hybrid design enables:

- More compact parameterization,
- Improved temporal dependency modeling,
- Competitive accuracy with significantly fewer parameters.

These results demonstrate that GCN-Mamba achieves a superior accuracy–efficiency trade-off, validating the effectiveness of incorporating Mamba-based sequence modeling into skeleton-based action recognition frameworks.

V. Conclusion

In this work, we introduce a compact yet highly efficient end-to-end model for human action recognition from skeleton data, leveraging a combination of Graph Convolutional Networks (GCNs) and State Space Models (SSMs) with integrated semantic information, such as joint type and frame index. At the heart of our approach lies the GCN-Mamba spatio-temporal block, which effectively captures spatial dependencies among joints within individual frames using GCNs, while temporal relationships across frames are modeled through the Mamba SSM, treating the set of joints in each frame as a unified entity. The incorporation of semantic features further strengthens the representational power of the GCN. The primary goal of this work is to demonstrate a compelling accuracy–efficiency trade-off when comparing GCN-Mamba directly to our prior model, ImpSGN. GCN-Mamba remains within 0.1% CS and 0.4% CV of ImpSGN on NTU RGB+D 60, ties with the best comparison model on NTU RGB+D 120 (SAGN, 82.1%/83.8%), and achieves this with approximately 72% fewer parameters (1.1M vs. 4.0M). This substantial reduction in model complexity at negligible accuracy cost underscores GCN-Mamba's suitability for deployment in resource-constrained real-world environments.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions

A. Mansouri led the core research work, contributing to Conceptualization, Methodology, Software development, Validation, Formal Analysis, Investigation, Data Curation, Visualization, and Writing—Original Draft & Editing. A. Elzaar contributed to Methodology, Validation, Data Curation, Visualization, and Writing—Review & Editing. T. Bakir contributed to Methodology, Supervision, Project Administration, Funding Acquisition, Resources, and Writing—Review. S. Femmam contributed to Methodology, Supervision, Project Administration, Funding Acquisition, Resources, and Writing—Review. All authors had approved the final version.

Acknowledgment

I would like to express my deep gratitude to Dr. T. Bakir and Dr. S. Femmam, university professors, for their invaluable guidance and sound advice throughout the writing of my article. Their expertise and availability have greatly contributed to the success of this work. I would also like to thank Dr. A. Elzaar for his careful proofreading and pertinent suggestions, which helped to improve the quality and clarity of this article. My sincere thanks to them for their support and kindness.

References

[1] A. Mansouri, T. Bakir, and S. Femmam, "Human action recognition with skeleton and infrared fusion model," *Journal of Image and Graphics*, vol. 11, no. 4, pp. 309–320, 2023.

[2] A. Mansouri, T. Bakir, and A. Elzaar, "Improved semantic-guided network for skeleton-based action recognition," *Journal of Visual Communication and Image Representation*, vol. 104, p. 104281, 2024.

[3] A. Mansouri, A. Elzaar, and T. Bakir, "Impsgnv2: Improved semantic-guided network with attention-based graph convolution (gcns) for skeleton-based action recognition," in *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, pp. 1–6, 2025.

[4] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[5] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1112–1121, 2020.

[6] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[7] A. Mansouri, L. Arnould, A. Lalande, and F. Meriaudeau, "Hv-octamamba: A high-order vision mamba network for robust retinal vasculature segmentation in octa images," *Biomedical Signal Processing and Control*, vol. 117, p. 109573, 2026.

[8] H. Duan, J. Wang, K. Chen, and D. Lin, "Pyskl: Towards good practices for skeleton action recognition," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 7351–7354, 2022.

[9] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12026–12035, 2019.

[10] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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

[12] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5323–5332, 2018.

[13] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *Pattern recognition. ICPR international workshops and challenges: virtual event, January 10–15, 2021, Proceedings, Part III*, pp. 694–701, Springer, 2021.

[14] M. Karim, S. Khalid, A. Aleryani, N. Tairan, Z. Ali, and F. Ali, "Hade: Exploiting human action recognition through fine-tuned deep learning methods," *IEEE Access*, vol. 12, pp. 42769–42790, 2024.

[15] M. Karim, S. Khalid, S. Lee, S. Almutairi, A. Namoun, and M. Abohashrh, "Next generation human action recognition: A comprehensive review of state-of-the-art signal processing techniques," *IEEE Access*, 2025.

[16] A. Mansouri, A. Elzaar, M. Madani, and T. Bakir, "Design and hardware implementation of cnn-gcn model for skeleton-based human action recognition," *WSEAS Transactions on Computer Research*, vol. 12, pp. 318–327, 2024.

[17] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 103–118, 2018.

[18] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 399–417, 2018.

[19] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019, 2016.

[22] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[23] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297, 2017.

[24] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "Adding attentiveness to the neurons in recurrent neural networks," in *proceedings of the European conference on computer vision (ECCV)*, pp. 135–151, 2018.