Spectral Scalpel: Amplifying Adjacent Action Discrepancy via Frequency-Selective Filtering for Skeleton-Based Action Segmentation

Supplementary Material

This appendix provides supplementary material for "Spectral Scalpel: Amplifying Adjacent Action Discrepancy via Frequency-Selective Filtering for Skeleton-Based Action Segmentation." We present detailed methodological supplements, comprehensive experimental settings, and further analytical results. Sec. 6 elaborates on architectural components not detailed in the main paper, including spatial modeling, frequency-domain transforms, temporal modeling, prediction refinement, and loss functions. Sec. 7 outlines the experimental setup, covering datasets, metrics, and implementation details. Sec. 8 provides additional comparisons, including quantitative clustering analysis, per-class performance breakdowns, inference speed comparison, and a series of robustness evaluations. Sec. 9 details extensive ablation studies validating the design of our core components: MASF, AADL, and FACM, along with other architectural choices. Finally, Sec. 10 discusses the work's limitations, potential future research directions, and broader societal impacts.

6. Method Supplement

This section provides a detailed exposition of the architectural components and methodologies employed in Spectral Scalpel. We elaborate on the spatial modeling pipeline (Sec. 6.1), the foundational Fast Fourier Transform (FFT) operations and their spectral properties (Sec. 6.2), the auxiliary temporal modeling blocks (Sec. 6.3), the prediction refinement branches (Sec. 6.4), and the comprehensive formulation of the loss functions (Sec. 6.5).

6.1. Spatial Modeling Supplement

In the spatial modeling stage, Spectral Scalpel follows the general paradigm introduced by previous works [8], where the input skeleton sequence $X \in \mathbb{R}^{C_0 \times T \times V}$ is first processed via multi-scale Graph Convolutional Networks (GCNs) [10, 13] to extract initial spatial representations $F_{s0} \in \mathbb{R}^{C \times T \times V}$. These features are then further refined through temporal-wise and channel-wise dynamic GCNs [2, 3, 8], resulting in the final spatial feature $F_s \in \mathbb{R}^{C \times T \times V}$.

Multi-Scale GCN for Spatial Topology Encoding. We adopt the multi-scale GCN mechanism originally proposed in [13] and introduced into action segmentation by [10] to capture the human skeleton's multi-hop connectivity patterns. To this end, we first define a set of k-adjacency matrices $A^k \in \{0,1\}^{V \times V}$, where each matrix encodes joint pairs at exactly k-hop distances in the skeletal graph:

$$A_{ij}^k = \begin{cases} 1, & \text{if } d(\alpha_i, \alpha_j) = k \text{ or } i = j, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $d(\alpha_i, \alpha_j)$ denotes the shortest path length (i.e., number of bones) between joints α_i and α_j. For example, the distance between the left wrist and left elbow is 1, while the distance between the left wrist and left shoulder is 2.

Next, we construct a unified multi-scale adjacency matrix $A^{MS} \in \{0,1\}^{V \times KV}$ by concatenating all normalized k-adjacency matrices up to scale K:

$$\begin{aligned} A^{MS} =& [(\tilde{D}^1)^{-\frac{1}{2}} A^1 (\tilde{D}^1)^{-\frac{1}{2}}] \oplus \cdots \\ & \oplus [(\tilde{D}^K)^{-\frac{1}{2}} A^K (\tilde{D}^K)^{-\frac{1}{2}}], \end{aligned} \tag{9}$$

where $\oplus$ denotes concatenation along the second (column) dimension, and each $\tilde{D}^k$ is a diagonal normalization matrix with elements defined as $\tilde{D}_{ii}^k = \sum_j A_{ij}^k + \epsilon$, where a small constant $\epsilon = 0.001$ is added to prevent division by zero due to empty rows.

Given the input feature X, the multi-scale GCN produces spatial features F_{s0} via:

$$F_{s0} = \text{ReLU}\left(\text{reshape}\left[(A^{MS} + B) \cdot X\right] \cdot W_{s0}\right), \tag{10}$$

where $B \in \mathbb{R}^{V \times KV}$ is a learnable matrix that adaptively encodes non-local joint correlations, and $\text{reshape}(\cdot)$ denotes a reshaping operation that transforms the intermediate feature from $\mathbb{R}^{C_0 \times T \times KV}$ to $\mathbb{R}^{KC_0 \times T \times V}$. The operator $W_{s0} \in \mathbb{R}^{1 \times 1 \times KC_0 \times C}$ is a point-wise convolution used to fuse and project the concatenated features to the desired channel dimension.

Temporal-wise and Channel-wise Dynamic GCNs for Fine-Grained Spatial Adaptation. To further capture the intricate and dynamic inter-joint dependencies in a fine-grained and feature-adaptive manner, Spectral Scalpel integrates both temporal-wise and channel-wise dynamic GCNs into its spatial modeling pipeline. The concept of channel-wise dynamic GCN was originally introduced in CTR-GCN [3], enabling independent spatial relation modeling across feature channels. Building on this, subsequent works [2, 8] extended the idea to the temporal domain, developing temporal-wise dynamic GCNs to model frame-specific spatial structures.

Formally, given the initial spatial feature $F_{s0} \in \mathbb{R}^{C \times T \times V}$, the model first applies two parallel convolutional heads to produce intermediate embeddings: $P, Q \in$

$\mathbb{R}^{C_1 \times T \times V}$. To capture both temporal- and channel-specific structural cues, two types of pooling operations are performed. Channel pooling along the channel dimension yields: $P^T, Q^T \in \mathbb{R}^{T \times V}$, while temporal pooling along the temporal axis produces: $P^C, Q^C \in \mathbb{R}^{C_1 \times V}$.

Next, we compute two adaptive graph structures using cross-joint mean differences to measure pairwise joint dissimilarities in each domain. Specifically, the temporal-wise dynamic graph $G^T \in \mathbb{R}^{T \times V \times V}$ and the channel-wise dynamic graph $G^C \in \mathbb{R}^{C_1 \times V \times V}$ are computed as:

$$G^T_{t,i,j} = P^T_{t,i} - Q^T_{t,j}, \quad G^C_{c,i,j} = P^C_{c,i} - Q^C_{c,j}, \tag{11}$$

where $G^T_{t,i,j}$ denotes the relation between joint i and j in the t-th frame, and similarly $G^C_{c,i,j}$ indicates the relation between joint i and j under the c-th channel perspective.

Furthermore, inspired by TRG-Net [8], we inject semantic structural priors into these dynamic graphs by incorporating a Text-derived Joint Graph (TJG). Specifically, the descriptive texts of the V joints are encoded via a pre-trained BERT [4] model to extract joint embeddings. By calculating the L2 distances among these embeddings and applying inverse normalization, we construct a static semantic graph $G^{txt} \in \mathbb{R}^{V \times V}$. This prior graph G^{txt} is then broadcasted and added element-wise to both G^T and G^C, effectively enriching the data-driven dynamic graphs with explicit anatomical semantics.

The refined input feature $F_{s1} \in \mathbb{R}^{C \times T \times V}$, obtained via a point-wise convolution over F_{s0}, is then modulated by these two dynamic graphs through graph convolution along the joint dimension. The final spatially-adaptive feature $F_s \in \mathbb{R}^{C \times T \times V}$ is computed as:

$$F_s = \mathrm{ReLU}\left(\mathrm{BN}\left[F_{s1} \cdot G^T + F_{s1} \cdot G^C\right]\right) + F_{s0}, \tag{12}$$

where $\cdot$ denotes batched matrix multiplication over the joint axis V, $\mathrm{BN}(\cdot)$ represents batch normalization, and $\mathrm{ReLU}(\cdot)$ is the non-linear activation function. This fusion mechanism enables the model to capture joint dependencies that are adaptively specialized to individual frames and channels, leading to enhanced spatial expressivity and discriminability.

6.2. Frequency Modeling Supplement

While frequency modeling has been discussed in the main body, this section provides a detailed exposition of the Fast Fourier Transform (FFT) and its inverse (iFFT), which serve as the fundamental tools for transforming features from the time domain to the frequency domain. We also elaborate on the spectral properties that arise from this transformation, including Hermitian symmetry and the implications of the Nyquist-Shannon Sampling Theorem.

Fast Fourier Transform and Spectral Properties. To enable frequency-domain modeling of temporal dynamics, we apply the FFT and its inverse, denoted by $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$, respectively. The FFT is performed along the temporal dimension T of the input feature tensor $F \in \mathbb{R}^{C \times T \times V}$ or $F \in \mathbb{R}^{C \times T}$, converting time-domain signals into the frequency domain. The resulting frequency-domain representation lies in the complex space, with shape $\mathbb{C}^{C \times S \times V}$ or $\mathbb{C}^{C \times S}$.

Formally, for a feature of dimension $\mathbb{R}^{C \times T \times V}$, the FFT is applied to each channel c and spatial joint v. Its computation aligns with the Discrete Fourier Transform (DFT) and is defined as follows:

$$\begin{gathered}\hat{F}_{c,s,v} = \mathcal{F}(F)_{c,s,v} = \sum_{t=0}^{T-1} F_{c,t,v} \cdot e^{-2\pi i \cdot \frac{st}{T}}, \\ s = 0, 1, \ldots, S-1,\end{gathered} \tag{13}$$

where $S = T$ (i.e., the output frequency spectrum has the same length as the input signal). To reconstruct the time-domain signal from its frequency-domain representation $\hat{F} \in \mathbb{C}^{C \times S \times V}$, we apply the inverse FFT:

$$\begin{gathered}F_{c,t,v} = \mathcal{F}^{-1}(\hat{F})_{c,t,v} = \frac{1}{T} \sum_{s=0}^{S-1} \hat{F}_{c,s,v} \cdot e^{2\pi i \cdot \frac{st}{T}}, \\ t = 0, 1, \ldots, T-1.\end{gathered} \tag{14}$$

These operations are implemented using the FFT algorithm with a computational complexity of $\mathcal{O}(T \log T)$, which is significantly more efficient than the naive DFT with complexity $\mathcal{O}(T^2)$.

For real-valued input signals, the frequency spectrum exhibits *Hermitian symmetry*, meaning that:

$$\hat{F}_{c,T-s,v} = \overline{\hat{F}_{c,s,v}}, \quad \text{for } s = 1, \ldots, T/2, \tag{15}$$

where $\overline{(\cdot)}$ denotes complex conjugation. This symmetry implies redundancy in the second half of the spectrum, allowing the signal to be fully represented by only the first $T/2+1$ frequency components. Thus, for storage and computational efficiency, the frequency-domain feature $\hat{F} \in \mathbb{C}^{C \times S \times V}$ can be truncated to $S = T/2 + 1$ (down from $S = T$). The same FFT/iFFT strategy is applied to features of dimension $\mathbb{R}^{C \times T}$ for each channel c.

According to the Nyquist-Shannon Sampling Theorem, a signal with maximum frequency f_{max} must be sampled at a rate $f_s \geq 2f_{\text{max}}$ to avoid aliasing. In our case, the temporal signals are sampled uniformly, and the maximum resolvable frequency—the *Nyquist frequency*—is given by: $f_{\text{Nyquist}} = f_s/2$. The frequency resolution Δ of the spectrum is determined by the number of time samples T, such that $\Delta = f_s/T$. Consequently, the i-th frequency bin corresponds to $f_i = i \cdot f_s/T$.

6.3. Temporal Modeling Supplement

In each temporal block of Spectral Scalpel, temporal modeling is comprehensively addressed through a combination of frequency-domain channel evolution and time-domain temporal interaction operations. In addition to the proposed Frequency-Aware Channel Mixer (FACM) for frequency-level channel evolution, each temporal block is equipped with a Linear Transformer for temporal interaction and an adaptive feature fusion mechanism—both structurally similar to prior STAS works [8, 10].

Specifically, for the l-th temporal block, the Linear Transformer first takes the output feature from the previous block, $F_t^{l-1} \in \mathbb{R}^{C\times T}$, and models temporal interactions to yield $F_{t1}^l \in \mathbb{R}^{C\times T}$. This is then processed by the proposed FACM to produce $F_{t2}^l \in \mathbb{R}^{C\times T}$. Finally, an adaptive feature fusion module integrates F_{t2}^l with early-stage temporal features $F_{t0}^l \in \mathbb{R}^{C\times T}$, extracted via a separate spatial-channel fusion head, to generate the final output of the block $F_t^l \in \mathbb{R}^{C\times T}$.

Linear Transformer for Temporal Interaction. A key challenge in temporal modeling for action segmentation is capturing inter-frame dependencies. Traditional methods based on dilated Temporal Convolutional Networks (TCNs) expand the receptive field progressively, but remain limited by fixed local temporal scopes. In contrast, Transformer-based attention mechanisms offer global context modeling. However, the quadratic complexity of conventional attention with respect to sequence length makes it computationally impractical for dense temporal sequences, especially in action segmentation [1, 18].

To this end, we adopt the Linear Transformer [9, 16], which reduces the attention complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, enabling efficient global temporal modeling. This approach has also been utilized in prior STAS works [7, 8, 10]. Formally, the Linear Transformer layer is defined as:

$$F_{t1}^l = \mathrm{ReLU}\left[\phi(Q_t^l)\left(\phi(K_t^l)^\top V_t^l\right)\cdot W_t^l + F_t^{l-1}\right], \quad (16)$$

where the input $F_t^{l-1} \in \mathbb{R}^{C\times T}$ is linearly projected into query, key, and value tensors:

$$Q_t^l = W_{Qt}^l F_t^{l-1}, \quad K_t^l = W_{Kt}^l F_t^{l-1}, \quad V_t^l = W_{Vt}^l F_t^{l-1}, \quad (17)$$

with $W_{Qt}^l, W_{Kt}^l, W_{Vt}^l \in \mathbb{R}^{C\times C_2}$. The output is projected back using $W_t^l \in \mathbb{R}^{C_2\times C}$. $\phi(\cdot)$ denotes the sigmoid activation function.

Adaptive Feature Fusion for Core Feature Retention. To align the frequency-modeled output feature $F_f \in \mathbb{R}^{C\times T\times V}$ with the temporal modeling dimension $\mathbb{R}^{C\times T}$, a spatial-channel fusion operation is employed. Inspired by [8], a point-wise convolution is first used to reduce the channel dimension, producing an intermediate representation of shape $\mathbb{R}^{C_3\times T\times V}$. This tensor is then reshaped to $\mathbb{R}^{VC_3\times T}$ by merging the channel and spatial dimensions, followed by another point-wise convolution to obtain $F_{t0}^l \in \mathbb{R}^{C\times T}$. This design allows each channel to be adaptively optimized with respect to the spatial dimension V, thereby preserving essential spatial semantics. However, due to the compression from VC to C, certain spatial information may be lost. To mitigate this, we employ adaptive feature fusion: we use convolution heads with different weights to generate l features from different perspectives, and fuse the feature F_{t0}^l of each perspective with the output F_{t2}^l of the FACM in each block. This fusion enriches the feature representation by leveraging multiple perspectives from the original VC-dimensional space. The final fused feature output of the l-th block is given by:

$$F_t^l = \mathrm{GeLU}\left[\left(F_{t0}^l \oplus F_{t2}^l\right)\cdot W_f \cdot W_l\right] + F_{t2}^l, \quad (18)$$

where $\oplus$ denotes channel-wise concatenation, $W_f \in \mathbb{R}^{1\times 1\times 2C\times C}$, $W_l \in \mathbb{R}^{C\times C}$ are both point-wise convolution operators. The additive residual connection with F_{t2}^l further enhances gradient flow and representation stability.

6.4. Prediction Refinement Supplement

Spectral Scalpel follows the prediction refinement paradigm established by previous STAS works, comprising two complementary branches: class prediction and boundary regression. The class refinement strategy, originally proposed in [5], has been widely adopted across most action segmentation frameworks. Subsequently, [6] introduced the boundary regression refinement, which effectively mitigates over-segmentation and improves overall segmentation accuracy.

Class Prediction Branch. Given the final representation of temporal modeling F_R, an initial frame-wise class prediction $Y_0^c \in \mathbb{R}^{Q\times T}$ is produced by a classification head, where Q denotes the number of classes. This prediction is progressively refined through S^c refinement stages. At the h-th stage, the previous prediction $Y_{h-1}^c \in \mathbb{R}^{Q\times T}$ is first projected to a feature $\mathbb{R}^{C\times T}$ via a point-wise convolution. A stack of 10 Linear Transformer layers is then applied to generate a more precise prediction $Y_h^c \in \mathbb{R}^{Q\times T}$. Each Linear Transformer layer adopts a cross-attention mechanism. Specifically, the query and key (Q_t, K_t) are computed from the current stage's own features, while the value (V_t) is derived from the final representation $\mathbb{R}^{C\times T}$ of the previous stage. The final stage yields the final refined class prediction $Y_F^c \in \mathbb{R}^{Q\times T}$.

Boundary Regression Branch. Similarly, the regression head produces an initial boundary confidence map $Y_0^b \in \mathbb{R}^{1\times T}$ from F_R, which is refined across S^b stages. At each h-th stage, the previous output $Y_{h-1}^b \in \mathbb{R}^{1\times T}$ is transformed into $\mathbb{R}^{C\times T}$ via a point-wise convolution, followed by a stack of 10 dilated TCNs to yield the refined boundary prediction $Y_h^b \in \mathbb{R}^{1\times T}$. The final output from the last stage is denoted as $Y_F^b \in \mathbb{R}^{1\times T}$.

6.5. Loss Function

As illustrated in Fig.3, the overall loss function is composed of four main components: (i) the Adjacent Action Discrepancy Loss proposed in this work for supervising the filtered feature representation (detailed in the main paper); (ii) the Action-Text Contrastive Loss adapted from previous language-assisted work [7]; and two commonly employed losses for action segmentation tasks: (iii) the Boundary Regression Loss, and (iv) the Action Segmentation Loss comprising classification and smoothing components. The detailed formulations of each component are provided below.

Action-Text Contrastive Loss. Following [7], we adopt a contrastive learning strategy to align visual features and textual embeddings. Given the final representation after temporal modeling $F_R \in \mathbb{R}^{C\times T}$, it is first projected into a latent space via a learnable mapping head $\mathbb{R}^{C\times C_t}$. The representation sequence is then segmented into N action segments based on the ground-truth boundaries, and the temporal mean is computed within each segment to yield a set of action-level visual representations $A^F \in \mathbb{R}^{N\times C_t}$.

Simultaneously, we obtain corresponding textual embeddings $A^E \in \mathbb{R}^{N\times C_t}$ for the N action classes using BERT [4] pretrained text encoder applied to the class-wise action descriptions. We compute a pairwise cosine similarity matrix between all visual features and textual embeddings:

$$S^A = \begin{bmatrix} sim(A_1^F, A_1^E) & \cdots & sim(A_1^F, A_N^E) \\ \vdots & \ddots & \vdots \\ sim(A_N^F, A_1^E) & \cdots & sim(A_N^F, A_N^E) \end{bmatrix}, \quad (19)$$

where cosine similarity is defined as $\mathrm{sim}(z_i, z_j) = (z_i \cdot z_j)/\|z_i\|\|z_j\|$. To enforce bidirectional alignment, we apply softmax normalization along both rows and columns, resulting in S_f^A (row-normalized) and S_e^A (column-normalized), respectively. A ground-truth similarity matrix $S^{GT} \in \mathbb{R}^{N\times N}$ is constructed, where each entry is 1 for positive (same-class) pairs and 0 otherwise. The loss minimizes the Kullback-Leibler (KL) divergence between the predicted and ground-truth similarity distributions:

$$\mathcal{D}_{KL}(U\|W) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} U_{ij}\log\left(\frac{U_{ij}}{W_{ij}}\right), \quad (20)$$

where $U, W \in \mathbb{R}^{N\times N}$. The final action-text contrastive loss is defined as:

$$\mathcal{L}_{AT} = \frac{1}{2}[\mathcal{D}_{KL}(S^{GT}\|S_f^A) + \mathcal{D}_{KL}(S^{GT}\|S_e^A)]. \quad (21)$$

Action Segmentation Loss. The action segmentation loss supervises the class prediction branches at all stages. For each stage, the segmentation loss $\mathcal{L}_{as}$ comprises two terms: the standard frame-wise classification cross-entropy loss $\mathcal{L}_{ce}$, and the Gaussian Similarity-weighted Truncated Mean Squared Error (GS-TMSE) loss $\mathcal{L}_{gs\text{-}tmse}$, designed to penalize abrupt changes in predictions while preserving sharp action boundaries:

$$\mathcal{L}_{as} = \mathcal{L}_{ce} + \mathcal{L}_{gs-tmse} = -\frac{1}{T}\sum_{t}\log(\hat{y}_{t,\hat{c}}) + \frac{1}{TC}\sum_{t,c} e^{-\frac{\|x_t - x_{t-1}\|^2}{2\sigma^2}} \min\left(\left|\log\frac{\hat{y}_{t,c}}{\hat{y}_{t-1,c}}\right|^2, \tau^2\right), \quad (22)$$

where $\hat{y}_t \in \mathbb{R}^C$ denotes the predicted class probabilities at frame t, $\hat{y}_{t,\hat{c}}$ is the predicted probability for the ground truth class $\hat{c}$, and x_t represents the input feature at frame t. By weighting the temporal penalty with the Gaussian similarity of adjacent features, this term adaptively enforces smoothness within uniform action segments while relaxing the penalty at action boundaries to avoid over-smoothing. The standard deviation σ, which controls the similarity sensitivity, is set to 1.0. Furthermore, a predefined threshold $\tau = 4$ is applied to truncate the squared log-difference, preventing excessively large gradients.

Boundary Regression Loss. The boundary regression loss supervises the prediction of frame-level action boundaries. At each stage, the loss is computed using binary cross-entropy:

$$\mathcal{L}_{br} = -\frac{1}{T}\sum_{t}\left(b_t\log(\hat{b}_t) + (1-b_t)\log(1-\hat{b}_t)\right), \quad (23)$$

where $b_t \in \{0, 1\}$ is the binary ground truth boundary label at frame t, and $\hat{b}_t \in [0, 1]$ is the predicted boundary probability.

Total Loss. The total training objective integrates all the aforementioned losses. The overall loss function is as follows:

$$\mathcal{L} = \sum_{h=0}^{S^c}\mathcal{L}_{as} + \lambda_1\sum_{h=0}^{S^b}\mathcal{L}_{br} + \lambda_2\mathcal{L}_{AT} + \lambda_3\mathcal{L}_{AAD} \quad (24)$$

where $\mathcal{L}_{AAD}$ denotes the Adjacent Action Discrepancy loss (defined in the main text), and λ_1, λ_2, and λ_3 are the balancing coefficients, empirically set to 1.0, 0.8, and 1.0, respectively.

7. Experimental Setup Details

This section outlines the comprehensive experimental protocol used to validate our model. We describe the datasets and preprocessing procedures (Sec. 7.1), the standard evaluation metrics (Sec. 7.2), and additional implementation-specific details and hyperparameter settings (Sec. 7.3) to ensure reproducibility.

7.1. Dataset and Preprocessing

We utilize the following datasets and follow the standard preprocessing protocols adopted by previous STAS studies to ensure fair comparison and reproducibility.

PKU-MMD v2 [11] is a large-scale multi-modal action detection dataset recorded at 50 Hz using Kinect V2 sensors. It provides 3-axis spatial coordinates for 25 body joints, covering 1009 action instances across 52 categories, totaling over 50 hours of motion data. We adopt two official partition protocols: (i) Cross-subject (X-sub), where 775 videos are used for training and 234 unseen subjects for testing; and (ii) Cross-view (X-view), where training samples are taken from the middle and right camera views, while testing samples are drawn from the left view. During preprocessing, we compute 6-channel features composed of joint-wise relative positions and temporal displacements based on the 3-axis coordinates. The PKU-MMD v2 dataset is licensed under the Apache 2.0 License.

LARa [14] focuses on capturing typical warehouse manipulation and locomotion behaviors, recorded at 200 Hz using an optical marker-based motion capture system. It provides 3-axis position and orientation data for 19 joints across 377 video samples, categorized into 8 action classes, amounting to 758 minutes of data. In preprocessing, we downsample all sequences to 50 Hz to ensure temporal consistency across datasets. We then extract 12-channel features by computing joint-wise relative positions and orientations, along with their respective temporal displacements. The LARa dataset is licensed under the Creative Commons Attribution Non-Commercial 4.0 International License.

MCFS-130 [12] is a figure skating action dataset designed for sports analysis. It is collected at 30 Hz using the OpenPose toolbox and includes 2D joint coordinates for 25 body joints. The dataset contains 271 video clips covering 130 fine-grained action categories, totaling 17.3 hours. For this dataset, we directly use the 2-axis positions to construct 2-channel relative position features for each joint. The MCFS-130 dataset does not specify a license in its original publication, but our use is strictly non-commercial and properly cited the original work.

TCG-15 [17] is a traffic control gesture dataset designed for autonomous driving applications. It is collected at 100 Hz using IMU sensors and records 3-axis positions for 17 joints. The dataset contains 550 annotated action segments across 4 categories, with a total duration of approximately 140 minutes. During preprocessing, we compute 6-channel features by deriving relative joint positions and their corresponding displacements from the raw 3-axis positional data. The TCG-15 dataset is licensed under the MIT License.

7.2. Evaluation Metrics

In accordance with standard practice in prior work, we evaluate model performance using three complementary metrics: Frame-wise Accuracy (Acc), Segmental Edit Score (Edit), and Segmental F1 Score at intersection-over-union (IoU) thresholds of 0.10, 0.25, and 0.50, denoted as F1@{10, 25, 50}. These metrics jointly assess both frame-level precision and segment-level structural correctness.

Frame-wise Accuracy (Acc) computes the proportion of frames whose predicted labels match the ground truth. Formally, it is defined as: $Acc = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}(y_t = \hat{y}_t)$, where T is the total number of frames, y_t and $\hat{y}_t$ denote the ground-truth and predicted labels at frame t, respectively, and $\mathbb{I}(\cdot)$ is the indicator function. Although simple and widely adopted, this metric is overly sensitive to minor misclassifications in long segments and fails to penalize over-segmentation.

Segmental Edit Score evaluates the sequence-level similarity between predicted and ground-truth segmentations by computing the minimum number of edit operations—insertions, deletions, and substitutions—required to convert the predicted action sequence into the ground truth. It is particularly effective in capturing structural discrepancies such as over-segmentation, while remaining agnostic to the duration of individual segments.

Segmental F1 Score assesses segment-level alignment by matching predicted and ground-truth segments using a temporal IoU criterion. A predicted segment is considered a true positive if it shares the same action label and exceeds a specified IoU threshold with any ground-truth segment. Unmatched predictions and ground-truth segments are treated as false positives and false negatives, respectively. Precision and recall are then computed, and the F1 score is defined as: $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. By incorporating multiple IoU thresholds (i.e., F1@10, F1@25, F1@50), this metric provides a nuanced and robust evaluation of both coarse and fine temporal alignment, while explicitly penalizing fragmented or imprecise segmentations.

7.3. Additional Implementation Details

In addition to the implementation details provided in Sec. 4.1, we present further implementation specifications for the supplementary methods described in Sec. 6. Notably, all the implementation configurations are fully aligned with those used in the previous STAS framework.

For the multi-scale GCN module, the scale K is set to 13. In the temporal-wise and channel-wise dynamic GCNs, the channel dimension C_1 is set to 16. For the FFT operations, we adopt the backward normalization mode, which applies no scaling during transformation. In the Linear Transformer component, the channel dimensions for the query, key, and value projections are all set to $C_2 = 16$. For the adaptive feature fusion module, the compression operation reduces the feature dimensionality to $C_3 = 8$. Regarding the prediction refinement, we use $S^c = 1$ stage for the class prediction branch and $S^b = 2$ stages for the bound-

Table 4. Comparison of Mean Clustering Metrics (SC, CH, DB) on the PKU-MMD v2 (X-sub, X-view), LARa and MCFS-130 datasets. SC denotes the Silhouette Coefficient Index, CH is the Calinski–Harabasz Index, and DB indicates the Davies–Bouldin Index.

Dataset	PKU-MMD v2 (X-sub)			PKU-MMD v2 (X-view)			MCFS-130			LARa		
Metric	SC↑	CH↑	DB↓	SC↑	CH↑	DB↓	SC↑	CH↑	DB↓	SC↑	CH↑	DB↓
DeST [10]	0.19	63.5	1.59	0.08	62.6	2.07	0.01	36.6	2.22	0.14	722.4	1.74
LaSA [7]	0.25	79.7	1.31	0.16	78.3	1.62	0.05	42.9	2.07	0.24	**1245.1**	1.18
SpecScalpel (Ours)	**0.30**	**88.6**	**1.24**	**0.29**	**128.9**	**1.26**	**0.07**	**49.3**	**1.94**	**0.25**	1189.7	**1.12**

ary regression branch. For the action-text contrastive loss, text embeddings are derived from natural language descriptions of each action category in the dataset. Specifically, sentence-level embeddings are obtained by averaging the word token embeddings produced by a pretrained BERT encoder [4], resulting in a fixed embedding dimensionality of $\mathbb{R}^{768}$. To maintain modality alignment, the dimensionality of the action-level visual representation is set to $C_t = 768$, matching the output of the pretrained text encoder. The BERT model is licensed under the Apache 2.0 License.

8. More Comparisons

This section presents further comparative analyses to provide a comprehensive quantitative and qualitative understanding of the Spectral Scalpel framework. We first evaluate the discriminability of the learned features via established clustering metrics (Sec. 8.1) and conduct a detailed per-class analysis to highlight the specific advantages and inherent limitations of our frequency-domain modeling (Sec. 8.2). Next, we validate the practical efficiency of our approach through an inference speed comparison (Sec. 8.3). Finally, we present a thorough assessment of the model's robustness against a spectrum of challenging conditions, including spatial noise and joint occlusions (Sec. 8.4), boundary annotation ambiguity (Sec. 8.5), and temporal scale variations (Sec. 8.6).

8.1. Clustering Performance Comparison

The t-SNE visualization in Fig. 6 provides only a qualitative assessment of the clustering behavior in the representation space. To more intuitively and quantitatively evaluate the discriminability of the learned features, we compute a series of well-established clustering validity metrics on all high-dimensional action-segment representations $F_{ac} \in \mathbb{R}^C$, including the Silhouette Coefficient (SC) Index, Calinski–Harabasz (CH) Index, and Davies–Bouldin (DB) Index. These metrics jointly measure inter-cluster separability and intra-cluster compactness. Specifically, the Silhouette Coefficient evaluates how well each sample fits within its assigned cluster relative to other clusters (ranging from –1 to 1, with higher values indicating denser and better-separated clusters). The Calinski–Harabasz Index (Variance Ratio Criterion) measures the ratio of between-cluster to within-cluster dispersion (ranging from 0 to ∞, with higher values denoting superior clustering). The Davies–Bouldin Index assesses the average similarity between each cluster and its most similar counterpart (ranging from 0 to ∞, with lower values indicating better separability).

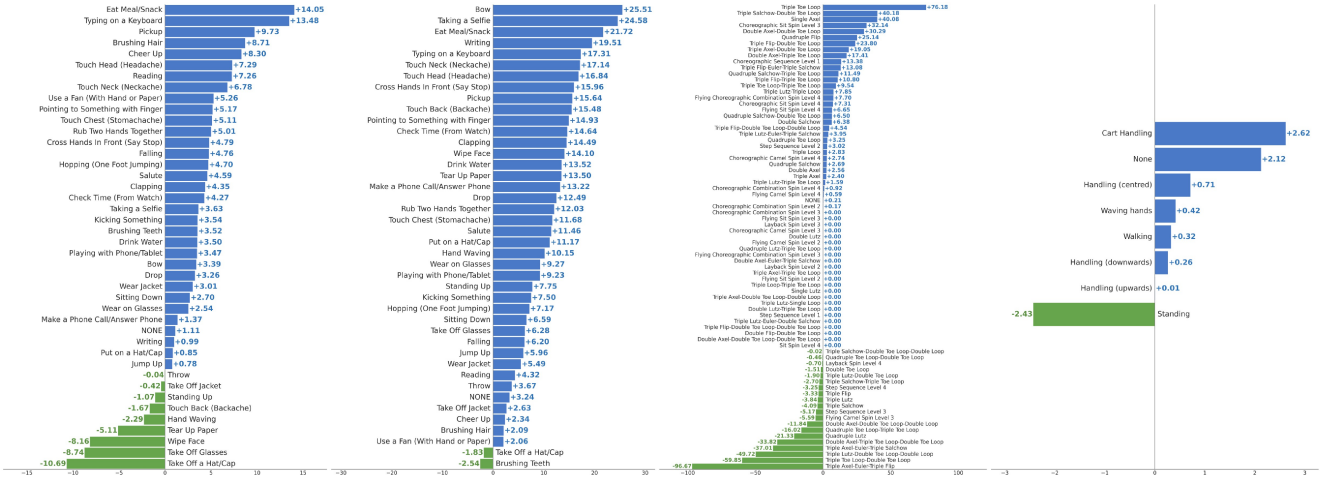
We compare DeST [10], LaSA [7], and our SpecScalpel across the PKU-MMD v2 (X-sub and X-view) [11], MCFS-130 [12], and LARa [14] datasets, as summarized in Tab. 4. Our model achieves the best scores on nearly all metrics, clearly demonstrating the strong clustering structure and high discriminability of the learned representations.

8.2. Analysis of Per-Class Performance

To further illustrate the detailed impact of the proposed discrepancy-guided frequency-selective filtering, we compare the per-class frame-wise F1 scores between our SpecScalpel and LaSA [7]. Notably, both methods are built upon DeST [10] and adopt the Action–Text Contrastive Loss; hence, the primary distinction lies in the frequency-domain modeling introduced by SpecScalpel. This comparison enables a clearer understanding of the types of actions and scenarios in which the proposed frequency-based modeling demonstrates its advantages, as well as those where it faces inherent challenges and limitations.

As shown in Fig. 8, SpecScalpel exhibits substantial improvements for action classes characterized by pronounced high-frequency components or sharp boundary transitions, highlighting the effectiveness of discrepancy-guided frequency-selective filtering. Notable gains are observed for high-frequency repetitive actions such as Eat Meal/Snack (+14.05% on PKU-MMD X-sub / +21.72% on PKU-MMD X-view), Typing on a Keyboard (+13.48% on PKU-MMD X-sub / +17.31% on PKU-MMD X-view), and Brushing Hair (+8.71% on PKU-MMD X-sub). These actions typically involve rapid, periodic, or quasi-periodic limb motions, producing distinctive high-frequency spectral signatures. The MASF module effectively captures and emphasizes these components, leveraging them as strong discriminative cues.

Similarly, SpecScalpel achieves significant improvements on short-duration actions with abrupt motion boundaries, such as Bow (+25.51% on PKU-MMD X-view) and Taking a Selfie (+24.58% on PKU-MMD X-view). These



(a) PKU-MMD(X-sub) (b) PKU-MMD(X-view) (c) MCFS-130 (d) LARa

Figure 8. Per-class frame-wise F1 score differences between our SpecScalpel and the state-of-the-art LaSA method on the PKU-MMD v2 (X-sub and X-view), LARa, and MCFS-130 datasets. Blue bars represent action classes where SpecScalpel achieves higher F1 scores, whereas green bars correspond to minor declines. The results indicate that SpecScalpel consistently enhances performance across the majority of action categories, with only limited exceptions.

actions exhibit transient, high-frequency spectral responses at the onset and termination phases. The AADL module, by optimizing inter-action spectral discrepancies, enhances such boundary sharpness, thereby outperforming LaSA, which relies solely on temporal-domain modeling.

However, Fig. 8 also reveals certain failure cases and inherent limitations. SpecScalpel performs less effectively on low-frequency or quasi-static actions, such as Standing (–2.43% on LARa) and Standing Up (–1.07% on PKU-MMD X-sub), whose motion characteristics are dominated by static postures and minimal spectral variation concentrated in low and zero-frequency bands. The frequency-selective mechanism may inherently bias toward dynamic information, leading to insufficient contextual modeling for static actions.

More notably, actions with spectrally similar motion patterns pose additional challenges. For instance, temporally opposite yet spectrally identical actions—such as Take Off a Hat/Cap (–10.69% on PKU-MMD X-sub / –1.83% on PKU-MMD X-view) and Take Off Glasses (–8.16% on PKU-MMD X-sub)—exhibit nearly indistinguishable dominant frequencies from their corresponding "Put on" counterparts, rendering the discrepancy-guided filtering less discriminative. Furthermore, frequency-degenerate action pairs or compositional actions, such as Quadruple Lutz (–21.33% on MCFS-130) vs. Triple Lutz (–3.84% on MCFS-130), or Triple Toe Loop–Double Toe Loop (–59.85% on MCFS-130) vs. Double Toe Loop (-4.48% on MCFS-130), differ mainly in repetition count or combination structure, which are inherently difficult to distinguish in the frequency domain.

In summary, SpecScalpel significantly enhances the representation of high-frequency dynamic motions and sharp-boundary transitions through frequency-domain modeling. Nonetheless, it faces challenges in low-frequency static actions and spectrally indistinguishable motion pairs, revealing its current limitations and providing valuable insights for future work on more adaptive frequency modeling.

8.3. Inference Speed Comparison

To evaluate the practical efficiency of our proposed framework, we measure the average inference speed on the test set using a single NVIDIA RTX 3090 GPU with a batch size of 1. During inference, our method requires 146 ms/video. This speed is strictly faster than recent models such as LaSA (157 ms/video) and ME-ST (306 ms/video), while remaining highly competitive with DeST (114 ms/video). Notably, compared to the baseline (i.e., the architecture without MASF, AADL, and FACM), the integration of our frequency-selective modules introduces a modest overhead of 23 ms/video. This efficiency is primarily attributed to the inherently low time complexity of the FFT, which operates at $\mathcal{O}(T \log T)$. Consequently, the latency incurred by our spectral operations is substantially lower than that of stacking complex self-attention mechanisms or deep GCN operators. Given the significant performance improvements—such as the +4.8% F1@50 absolute gain on the PKU-MMD (X-view) benchmark—this slight increase in computational cost is well-justified, demonstrating a highly favorable accuracy-efficiency trade-off.

Table 5. Robustness evaluation under random Gaussian noise and joint occlusion on PKU-MMD v2 (X-sub) dataset. For each method, the first row reports the absolute performance, and the second row indicates the relative performance drop compared with the clean setting.

Noise	0.3x Gaussian Noise					0.5x Gaussian Noise					30% Joint Occlusion				
Metric	Acc	Edit	F1@{10, 25, 50}			Acc	Edit	F1@{10, 25, 50}			Acc	Edit	F1@{10, 25, 50}		
DeST	66.8	65.2	71.0	67.4	54.3	60.5	58.6	64.1	59.7	45.4	67.3	66.5	71.7	67.8	53.5
	-4.94%	-5.95%	-4.64%	-5.12%	-7.48%	-13.98%	-15.44%	-13.93%	-15.93%	-22.59%	-4.24%	-4.11%	-3.71%	-4.57%	-8.80%
LaSA	71.3	71.5	77.3	72.9	60.8	65.5	64.7	69.9	65.3	51.6	70.2	70.4	75.4	71.5	58.5
	-3.00%	-2.60%	-1.29%	-2.54%	-4.40%	-10.91%	-11.92%	-10.67%	-12.70%	-18.84%	-4.50%	-4.11%	-3.73%	-4.41%	-8.01%
SpecScalpel	**74.7**	**74.1**	**79.3**	**76.2**	**66.0**	**73.9**	**73.2**	**78.3**	**75.6**	**64.8**	**72.7**	**72.0**	**77.4**	**74.2**	**62.3**
	-0.92%	**-0.53%**	**-0.50%**	**-0.78%**	**-0.90%**	**-1.98%**	**-1.72%**	**-1.74%**	**-1.56%**	**-2.69%**	**-3.56%**	**-3.32%**	**-2.86%**	**-3.38%**	**-6.44%**

8.4. Robustness to Input Perturbations

To further assess the robustness of SpecScalpel against noise perturbations, we compare its performance degradation with that of other methods under random Gaussian noise and joint occlusion, as summarized in Tab. 5. Specifically, all models are first trained on the original clean datasets and then evaluated on test sets corrupted by the noises. Gaussian noise with standard deviations of 0.3 and 0.5 times the original data scale is applied, while the random occlusion masks 30% of the joints. For consistent and fair comparison, the same random seeds are used across all methods to ensure identical noise patterns.

Tab. 5 reports both the absolute performance under noisy conditions and the relative performance drop compared with the clean setting. As observed, SpecScalpel maintains high performance even under severe noise, exhibiting only minor degradation compared with other methods. In particular, its performance remains nearly unaffected by Gaussian noise and is still stable under occlusion scenarios. This robustness arises from the discrepancy-guided frequency-selective filtering, which effectively suppresses irrelevant high-frequency components caused by skeleton jitter or occlusion noise. In summary, the introduced frequency-domain modeling not only enhances performance but also substantially improves the model's robustness to noise interference.

8.5. Robustness to Boundary Ambiguity

To further evaluate the robustness of our framework against boundary ambiguity and annotation errors, we conduct experiments introducing noisy boundary labels during training. Specifically, we apply a random jitter to the ground-truth action boundaries, shifting the start and end frames of each segment by a random offset $\delta \sim \mathcal{U}(-25, 25)$ frames. The models are trained on these corrupted labels and evaluated on the clean test set.

As shown in Table 6, SpecScalpel consistently maintains the highest absolute performance across all metrics despite severe boundary perturbations. Furthermore, it exhibits a more resilient relative performance drop compared to the Baseline and LaSA method. This demonstrates that although the Adjacent Action Discrepancy Loss (AADL) utilizes ground-truth boundaries to extract segments for discrepancy computation, it does not overfit to precise temporal localizations. Because the dominant spectral peaks of human motion are statistically stationary, AADL only requires the extracted segment to predominantly capture the target action's characteristics. Even if the segment is truncated or slightly contaminated by adjacent actions due to boundary jitter, the model still effectively learns to amplify the discrepancy between the principal spectral components. This confirms that our discrepancy-guided filtering captures intrinsic, action-specific spectral patterns rather than overfitting to boundary annotation artifacts.

Table 6. Robustness evaluation against noisy training boundaries on the PKU-MMD v2 (X-sub) dataset. For each method, the first row reports the absolute performance under random boundary jitter ($\delta \sim \mathcal{U}(-25, 25)$ frames), and the second row indicates the relative performance drop compared to the clean training setting.

Noisy Labels	Acc	Edit	F1@{10, 25, 50}		
DeST	65.9	61.8	68.5	63.2	48.9
	-6.21%	-10.76%	-8.09%	-10.94%	**-16.78%**
LaSA	68.1	65.1	70.8	65.7	51.2
	-7.37%	-11.37%	-9.64%	-12.17%	-19.47%
Baseline	67.1	64.9	71.4	65.3	51.3
	-8.78%	-11.04%	-8.71%	-12.40%	-20.21%
SpecScalpel	**70.0**	**67.6**	**74.0**	**69.1**	**55.2**
	-7.11%	**-9.28%**	**-7.10%**	**-10.00%**	-17.07%

8.6. Robustness to Temporal Rescaling

To evaluate the model's robustness against variations in frame rates and action execution speeds, we conduct a temporal rescaling experiment. Models trained on the original temporal scale are evaluated on test sequences rescaled by factors of 0.7x and 1.5x.

As shown in Table 7, SpecScalpel consistently achieves the highest absolute performance across all metrics, outperforming prior state-of-the-art methods by a significant margin. Furthermore, its relative performance degradation is tightly bounded and remains highly competitive with the baselines. This demonstrates that our model effectively

Table 7. Robustness evaluation against temporal rescaling on the PKU-MMD v2 (X-sub) dataset. Models trained on the original temporal scale are evaluated on sequences rescaled by 0.7x and 1.5x. For each method, the first row reports the absolute performance, and the second row indicates the relative performance drop compared to the clean setting.

Temporal Rescaling	0.7x Temporal Rescaling					1.5x Temporal Rescaling				
Metric	Acc	Edit	F1@{10, 25, 50}			Acc	Edit	F1@{10, 25, 50}		
DeST	69.6	66.1	73.1	70.3	57.6	62.5	61.0	65.4	60.6	48.1
	-1.00%	-4.67%	-1.88%	-0.97%	-1.83%	**-11.16%**	-11.98%	**-12.20%**	**-14.68%**	**-18.11%**
LaSA	72.9	71.3	77.2	74.2	62.4	64.3	64.3	67.9	63.4	50.3
	-0.82%	-2.86%	-1.40%	-0.80%	-1.89%	-12.52%	-12.47%	-13.24%	-15.24%	-20.88%
Baseline	72.6	71.0	77.0	73.3	62.2	64.0	64.0	67.2	61.8	50.4
	-1.33%	-2.74%	-1.53%	-1.70%	-3.34%	-13.03%	-12.26%	-14.02%	-17.10%	-21.58%
SpecScalpel	**74.8**	**72.5**	**78.6**	**76.3**	**65.4**	**66.1**	**65.6**	**69.2**	**65.4**	**53.4**
	-0.79%	**-2.65%**	**-1.37%**	**-0.65%**	**-1.80%**	-12.38%	**-11.95%**	-13.17%	-14.81%	-19.76%

learns the intrinsic spectral distributions of motion dynamics rather than overfitting to static frequency bins. We attribute this strong temporal scale robustness to the dynamic weighting branch within the Multi-scale Adaptive Spectral Filter (MASF). By dynamically adjusting the filter weights conditioned on the specific characteristics of the input instance, this branch adaptively compensates for the spectral shifts caused by temporal rescaling, thereby mitigating the brittleness associated with fixed, absolute-position frequency filtering.

9. More Ablation Studies

This section presents extensive ablation studies to rigorously validate the design choices of Spectral Scalpel. We analyze the placement and necessity of our core frequency-domain modules (Sec. 9.1), followed by detailed ablations on the design parameters of the Multi-scale Adaptive Spectral Filter (MASF) (Sec. 9.2), the Adjacent Action Discrepancy Loss (AADL) (Sec. 9.3), and the Frequency-Aware Channel Mixer (FACM) (Sec. 9.4). Finally, we ablate other architectural hyperparameters (Sec. 9.5).

9.1. Ablation Studies on Frequency Modeling

The Position of the Frequency Module. To validate our architectural design, we conducted a crucial ablation study to determine the optimal placement of the frequency modeling module (MASF+AADL). The primary goal of our module is to analyze features that retain distinct temporal frequency characteristics. We experimented with three strategic positions: (1) at the very beginning, processing raw skeleton coordinates; (2) after the spatial GCN modeling but before temporal modeling (our proposed approach); and (3) after the temporal modeling. The empirical results, as summarized in Tab. 8, unequivocally demonstrate the superiority of our chosen position. Placing the frequency module after spatial modeling achieves the best performance across all metrics. We attribute this success to the nature of the features at this stage: the GCN has already enriched the frame-level representations with crucial semantic information regarding body structure and joint coordination. Critically, since no inter-frame temporal modeling has occurred, the information for each frame remains independent, thus preserving the rich, high-fidelity frequency dynamics of the motion, making it the ideal point for frequency analysis.

Table 8. Ablation study on the position of the frequency module on PKU-MMD v2 (X-sub) dataset.

Position of Frequency Module	Acc	Edit	F1@{10, 25, 50}		
At the Very Beginning	74.8	74.3	78.8	75.9	65.6
After Spatial Modeling	**75.4**	**74.5**	**79.7**	**76.8**	**66.6**
After Spatial Modeling	70.6	70.0	75.4	71.9	60.6

Conversely, positioning the module at the very beginning leads to suboptimal results. While applying FFT to raw kinematic signals is plausible, these low-level features lack the semantic context provided by the GCN, limiting the effectiveness of the frequency analysis. The most significant performance degradation is observed when the module is placed after temporal modeling, with F1@50 dropping sharply to 60.6%. This confirms our hypothesis that temporal aggregation models inherently smooth the feature sequences to create homogeneous representations for classification. This process erases the motion frequency details that our module aims to capture, rendering a subsequent FFT largely ineffective. In conclusion, this ablation study provides evidence that the optimal position for frequency analysis is after spatial feature extraction but before temporal aggregation. This placement ensures that the module operates on features that are both semantically rich and dynamically intact.

The Necessity of Frequency-Domain Operations. A core design choice of our model is the deliberate use of the frequency domain for critical operations. To quantitatively validate this, we conducted an ablation study on our key modules: MASF and AADL. We tested variants where these modules were forced to operate directly in the

Table 9. Ablation study on the necessity of frequency-domain operations on PKU-MMD v2 (X-sub) dataset.

FFT or Not	Acc	Edit	F1@{10, 25, 50}		
Both use FFT	**75.4**	**74.5**	**79.7**	**76.8**	**66.6**
MASF w/o FFT	75.1	73.4	79.1	76.2	65.5
AADL w/o FFT	74.7	73.6	78.9	75.9	65.4
Both w/o FFT	74.2	73.2	78.5	75.4	64.9

time domain by removing the FFT/iFFT transformations. The empirical results, summarized in Tab. 9, unequivocally demonstrate the superiority of our approach. Our full model, with both modules operating in the frequency domain, achieves the highest performance. A significant performance degradation is observed when either module is reverted to the time domain. The most substantial drop occurs when both modules operate in the time domain, confirming that the benefits of frequency-domain processing are cumulative and essential.

These results provide strong empirical validation for our theoretical justifications. For the MASF module: Time-domain filtering is fundamentally flawed for this task. It suffers from parameter meaning drift, as a filter at a specific timestep (e.g., frame 10) processes entirely different semantic actions across various sequences, hindering generalization. Furthermore, time-domain filtering is equivalent to frequency-domain convolution (by the Convolution Theorem), which causes spectral smearing (an effect that blurs and contaminates the very characteristic frequency peaks we aim to enhance). In contrast, the frequency domain provides an absolute and invariant axis (e.g., 2 Hz always corresponds to a "twice per second" motion). Our Hadamard product in this space performs a clean, global feature sharpening, which is a far more stable and effective optimization problem than learning a massive, equivalent global convolution kernel in the time domain. For the AADL loss: The performance drop from AADL w/o FFT confirms that direct time-domain comparison is unreliable. Such comparisons are easily dominated by confounding variables like action duration, scale, or repetition. By projecting all signals onto the same absolute frequency axis $[0, f_s/2]$, the FFT provides a fair and normalized basis for comparison. This allows the AADL to ignore superficial variations and focus on the intrinsic dynamic properties of the motions, providing more precise and meaningful guidance for model optimization. In conclusion, this ablation study provides definitive quantitative evidence that the frequency domain is not merely an alternative but a superior representational space for both enhancing action-specific features and measuring sequence-level discrepancies, validating its central role in our architecture.

9.2. Ablation Studies on MASF

Ablation Study on Filter Dimension and Weighting Strategy in MASF. To evaluate the design of multi-head channel-shared filter and channel-wise weighting in the MASF, we conduct an ablation study as presented in Tab. 10. Our goal is to achieve adaptive, per-channel spectral filtering in a computationally efficient manner. The study compares our proposed design against several variants, including less efficient, brute-force approaches. Tab. 10 confirms the superiority of our strategy. Our proposed model (row 1), which uses channel-shared base filters with channel-wise weighting, achieves a high performance with a minimal addition of only 0.01G FLOPs and 0.006M parameters. This establishes a strong baseline for performance and efficiency.

The core of our design involves learning a small set of M filters that are channel-shared (each $H_m \in \mathbb{R}^{1\times R_m\times V}$) and then achieving adaptive channel-wise filtering via a dynamic-static channel-wise combination. Channel-wise weighting is the natural and necessary mechanism to achieve this. When we replace it with temporal-wise weighting (row 2) within the same filter-shared framework, performance drops significantly. This is because temporal weighting fails to create the per-channel filter specificity that is central to our design. Combining both weighting schemes (row 3) offers no improvement, suggesting that the temporal modulation is redundant and potentially disrupts the cleanly filtered signals. Furthermore, we compared our efficient model to computationally expensive alternatives using large, channel-specific filters (each $H_m \in \mathbb{R}^{C\times R_m\times V}$). A variant with only temporal weighting (Filter-Specific + Temporal-Weight, row 4) merely matches our performance but at a drastically higher computational cost. Adding both weighting mechanisms (Filter-Specific + Both Weights, row 5) yields only a marginal gain while incurring a massive parameter increase (+0.588M). In conclusion, this study validates that our design, which combines channel-shared base filters with a lightweight channel-wise weighting mechanism, represents the most elegant and cost-effective strategy. It achieves performance comparable to or better than computationally intensive alternatives, striking an optimal balance between effectiveness and efficiency.

Effect of Weighted Fusion Strategies in MASF. We evaluate different fusion strategies for aggregating multi-scale filtered features in MASF, including dynamic, static, and average fusion, and their combinations (Tab. 11). All strategies perform comparably across metrics, each offering unique advantages. Notably, combinations such as static-dynamic fusion yield further gains due to their complementary nature. Meanwhile, it can be observed that average fusion's non-learnability yields weaker complementary advantages than dynamic/static methods. The static-dynamic strategy achieves the best overall trade-off between robust-

Table 10. Ablation study on filter dimension and weighting strategy in MASF on PKU-MMD v2 (X-sub) dataset.

FFT or Not	Acc↑	Edit↑	F1@{10, 25, 50}↑			FLOPs↓	Param.↓
Filter-Shared + Channel-Weight	75.4	74.5	79.7	76.8	66.6	**0.01G**	**0.006M**
Filter-Shared + Temporal-Weight	74.7	74.2	78.7	76.0	65.7	0.045G	0.010M
Filter-Shared + Both Weights	75.2	74.3	78.9	76.0	66.0	0.045G	0.261M
Filter-Specific + Temporal-Weight	75.5	74.7	79.5	76.8	66.6	0.045G	0.337M
Filter-Specific + Both Weights	**75.6**	**74.9**	**79.5**	**77.0**	**66.8**	0.045G	0.588M

Table 11. Effect of weighted fusion methods on multi-scale filtered features on PKU-MMD v2 (X-sub) dataset.

Dynamic	Static	Average	Acc	Edit	F1@{10, 25, 50}		
✓	✗	✗	75.2	74.0	79.4	76.5	66.2
✗	✓	✗	74.6	73.9	78.8	75.9	66.4
✗	✗	✓	75.2	74.2	79.4	76.5	66.3
✓	✗	✓	75.4	74.2	78.6	76.2	66.3
✗	✓	✓	75.3	74.4	79.5	76.4	66.4
✓	✓	✗	75.4	**74.5**	**79.7**	**76.8**	**66.6**
✓	✓	✓	**75.5**	74.5	79.6	76.7	66.5

ness and adaptiveness, and is adopted in our final model.

Effect of Feature Domain and Weight Generation Strategy in Dynamic Weighting. To evaluate the impact of feature domain and weight generation strategy in the dynamic channel-wise weighting module, we conduct an ablation study as presented in Tab. 12. Specifically, we compare two sources of features for weight generation: temporal features prior to FFT (F_s) and frequency features after FFT (F_{f0}). Additionally, we examine two methods for generating the weights: our proposed formulation in Eq. (3) and the SE block-based strategy inspired by [15], where features of shape $\mathbb{R}^{C\times T\times V}$ are globally pooled along the temporal and spatial dimensions and passed through a two-layer SE block to generate weights of shape $\mathbb{R}^{C\times M}$. Experimental results indicate that using frequency-domain features leads to inferior performance compared to temporal features. This is attributed to the conversion from complex-valued frequency representations to amplitude spectra, which introduces information loss and degrades the effectiveness of the learned weights. Regarding the generation strategy, the SE block-based method also performs worse than our proposed approach. This is likely because global pooling suppresses fine-grained spatio-temporal cues and channel mixing is less compatible with the desired independent channel-wise modulation. In contrast, Eq. (3) enables dynamic weight generation that fully leverages temporal and spatial structures in a channel-independent manner. Based on these findings, we adopt temporal features and Eq. (3) for dynamic weight generation in our final model.

Effect of Weight Granularity and Initialization in Static Weighting. We investigate the influence of weight granularity and initialization strategy in the static channel-wise weighting branch, as shown in Tab. 13. For granularity, we compare a global configuration with shared weights across channels ($\mathbb{R}^{M\times 1}$) and a channel-specific configuration ($\mathbb{R}^{M\times C}$). Results reveal that channel-wise weights consistently outperform the global counterpart, suggesting that finer granularity allows more precise modulation of channel responses. As for initialization, both zero initialization and random initialization yield comparable performance after training, indicating that the static weights are sufficiently optimized regardless of their starting values. Consequently, our final design employs channel-wise granularity with zero initialization, offering both simplicity and strong empirical performance.

Table 12. Effect of feature domain and weight generation in dynamic weighting on PKU-MMD v2 (X-sub) dataset.

Feature Domain	Weight Generator	Acc	Edit	F1@{10, 25, 50}		
Temporal	SE Block	74.8	73.4	78.4	75.6	65.6
Temporal	Equation (3)	**75.4**	**74.5**	79.7	**76.8**	**66.6**
Frequency	SE Block	74.8	74.3	78.6	75.5	65.6
Frequency	Equation (3)	75.3	74.2	**79.8**	76.8	66.4

Table 13. Effect of granularity and initialization in static weighting on PKU-MMD v2 (X-sub) dataset.

Weight Granularity	Initialization	Acc	Edit	F1@{10, 25, 50}		
(M, C)	Random	75.4	**74.6**	79.6	76.6	66.6
(M, C)	Zero	**75.4**	74.5	**79.7**	**76.8**	**66.6**
(M, 1)	Random	75.4	74.3	79.6	76.3	66.3
(M, 1)	Zero	75.3	74.4	79.6	76.4	66.4

Effect of Fusion Strategy for Dynamic and Static Branch Features. We conduct an ablation study to assess different strategies for fusing features from the dynamic and static branches, as summarized in Tab. 14. Four fusion methods are compared: direct averaging, learnable scalar weighting, learnable channel-wise weighting, and channel-wise concatenation followed by convolution. Among them, the concatenation + convolution approach achieves the best performance by effectively capturing inter-branch correlations. However, direct averaging also delivers competitive results with significantly lower computational cost and no additional parameters. In contrast, the two learnable weighting schemes underperform, possibly due to their in-

Table 14. Effect of fusion strategy for dynamic and static branch features on PKU-MMD v2 (X-sub) dataset.

Fusion Method	Acc	Edit	F1@{10, 25, 50}		
Average Fusion	75.4	74.5	**79.7**	**76.8**	66.6
Learnable Scalar Weight	75.4	74.2	79.6	76.7	66.3
Channel-wise Weight	75.4	74.4	79.3	76.3	66.4
Channel Concat + Conv	**75.5**	**74.6**	79.7	76.5	**66.7**

Table 15. Effect of different numbers of filter heads M on PKU-MMD v2 (X-sub) dataset.

Number of filter heads	Acc	Edit	F1@{10, 25, 50}		
2	75.4	74.4	79.5	76.6	66.4
4	75.4	74.5	79.7	76.8	66.6
6	75.7	74.7	80.0	77.2	66.7
8	**75.8**	**74.9**	**80.1**	**77.4**	**66.7**

Table 16. Impact of maximum filter length R_{max} in spectral filtering on PKU-MMD v2 (X-sub) dataset.

Maximum Filter Length	Acc	Edit	F1@{10, 25, 50}		
32	75.3	74.2	79.5	76.7	66.5
64	75.4	74.5	79.7	76.8	66.6
128	75.7	74.3	79.8	76.8	66.8
256	**75.8**	**74.7**	**79.8**	**77.0**	**66.9**

Table 17. Comparison of multi-scale strategies with fixed maximum filter length 64 on PKU-MMD v2 (X-sub) dataset.

Multi-scale strategies	Acc	Edit	F1@{10, 25, 50}		
Single-scale	75.3	74.5	79.5	76.6	66.4
Equation (2)	**75.4**	74.5	**79.7**	**76.8**	**66.6**
$R_m = 2^{m+b}$	75.4	**74.6**	79.6	76.7	66.5

terference with the individual gradient flows of each branch, thus hampering optimization. Taking both performance and efficiency into account, we adopt direct averaging as the final fusion strategy in our model.

Effect of the Number of Filter Heads M. We examine the influence of the number of filter heads M in the multi-scale multi-head spectral filtering module, with results presented in Tab. 15. As M increases from 2 to 8, we observe a consistent improvement in performance. This can be attributed to the enhanced diversity of frequency-aware filtered representations, akin to the feature disentanglement observed in multi-head attention mechanisms. A higher number of heads also introduces a greater variety of filter scales, enabling more comprehensive spectral coverage. However, increasing M also leads to higher computational cost and parameter overhead. Moreover, the marginal gain in performance diminishes as M continues to grow. Considering the trade-off between model complexity and accuracy, we set the number of heads to $M = 4$ in our final configuration.

Effect of Maximum Filter Length $R_{\max}$ in Spectral Filtering. To investigate the impact of filter scale, we conduct an ablation study on the maximum filter length $R_{\max}$, which defines the upper bound of filter resolutions or scales across heads. Results in Tab. 16 show that increasing $R_{\max}$ from 32 to 256 consistently improves model performance, as longer filter lengths enable finer frequency resolution and more granular spectral filtering. Although this enhancement comes with a slight increase in parameter count and computation, the benefit in spectral precision is notable. In our final model, we set $R_{\max} = 64$ to balance accuracy and efficiency, while noting that higher values of $R_{\max}$ may be considered in resource-permissive applications for further performance gains.

Effect of Different Multi-Scale Strategies. We compare the effectiveness of two multi-scale strategies for spectral filtering: the linear scaling approach proposed in Eq. (2) of our paper and the traditional exponential scaling strategy defined as $R_m = 2^{m+b}$, as shown in Tab. 17. Both strategies yield modest improvements over the non-multi-scale baseline under a fixed maximum filter length of 64. Meanwhile, the multi-scale approach also results in smaller filter lengths (below 64) at other scales, thereby reducing both parameters and computational costs. Notably, the linear multi-scale strategy defined by Eq. (2) outperforms the exponential counterpart. We attribute this to its ability to reduce overlap between interpolated filter boundaries, which facilitates finer-grained and more discriminative channel-wise modulation. Based on these findings, we adopt the strategy in Eq. (2) as the default design.

9.3. Ablation Studies on AADL

Ablation Study on the Scope of Action Discrepancy in Adjacent Action Discrepancy Loss. To examine the effect of the computation scope in the discrepancy loss of different actions, we compare three variants: (i) Adjacent Action Discrepancy Loss (AADL), which computes spectral differences only between adjacent actions in each instance, as detailed in Sec. 3.3; (ii) Instance-scope Action Discrepancy Loss (IADL), which compares each action with all other non-identical-class actions within the same instance; and (iii) Batch-scope Action Discrepancy Loss (BADL), which further extends this comparison to all non-identical-class actions across the mini-batch.

As shown in Tab. 18, IADL and BADL achieve slightly better performance than AADL, with BADL achieving the highest score due to the richest discrepancy supervision. However, the performance gap is marginal. We attribute this to two reasons. First, although AADL only leverages adjacent actions, it already captures critical spectral transitions at temporal boundaries, which are most relevant for

Table 18. Impact of action discrepancy scope in adjacent action discrepancy loss on PKU-MMD v2 (X-sub) dataset. AADL is Adjacent Action Discrepancy Loss, IADL is Instance-scope Action Discrepancy Loss, and BADL is Batch-scope Action Discrepancy Loss.

Discrepancy Calculation Scope	Acc	Edit	F1@{10, 25, 50}		
AADL	75.4	74.5	79.7	76.8	66.6
IADL	75.5	**74.6**	79.7	76.9	66.6
BADL	**75.7**	74.5	**79.9**	**77.1**	**66.7**

action segmentation. Since segmentation errors predominantly occur near these transitions, adjacent discrepancies provide highly targeted supervision to sharpen boundary modeling. Second, although IADL and BADL introduce more extensive comparisons, all three variants effectively guide the Multi-scale Adaptive Spectral Filter (MASF) to suppress shared frequency components and enhance discriminative ones. As MASF is a global filter with limited capacity, its representational power may already be saturated even under local discrepancy supervision. Moreover, AADL is significantly more efficient, as it only compares adjacent actions. In contrast, IADL and BADL require pairwise comparisons among all non-identical-class actions, resulting in a quadratic increase in computation with respect to the number of action segments—especially for BADL, where comparisons span the entire batch. Therefore, we adopt AADL as the default setting, offering the best trade-off between performance and efficiency.

Amplitude vs. Complex in AADL Computation. In our AADL, the spectral discrepancy between adjacent actions is computed using strictly their amplitude spectra, deliberately discarding phase information. To validate this design choice, we compare our amplitude-only approach against calculating the discrepancy using the full complex spectrum (which inherently includes both amplitude and phase). As shown in Table 19, incorporating phase information yields negligible performance variations while effectively doubling the computational overhead of the discrepancy calculation. We attribute this to the crucial property of temporal shift-invariance. The discriminative characteristics of distinct actions are predominantly captured by their spectral energy distributions (amplitude). In contrast, phase primarily encodes non-discriminative temporal offsets or exact start times—an action retains its core semantic identity regardless of the specific phase at which the segment begins. Forcing the model to differentiate based on phase could inadvertently cause overfitting to specific temporal alignments and introduce unnecessary computational redundancy. Therefore, our intentional adoption of a phase-invariant amplitude spectrum ensures that AADL remains both robust to temporal shifts and highly efficient during training.

Table 19. Ablation on spectral representations (Amplitude vs. Complex) for computing the AADL on the PKU-MMD v2 (X-sub) dataset.

Spectral Representation	Acc	Edit	F1@{10, 25, 50}		
Amplitude Only	75.4	**74.5**	**79.7**	**76.8**	**66.6**
Complex	**75.5**	74.4	79.6	76.8	66.6

Table 20. Effect of different AADL computation strategies on PKU-MMD v2 (X-sub) dataset.

Discrepancy Metric	Loss Form	Acc	Edit	F1@{10, 25, 50}		
Normalized L1	-log(tanh(·))	75.4	**74.5**	**79.7**	**76.8**	**66.6**
Normalized L2	-log(tanh(·))	75.2	74.0	79.2	76.2	66.3
L-infinity	-log(tanh(·))	74.8	73.4	78.6	75.4	64.8
Normalized L1	-log(B-sigmoid(·))	75.2	74.3	79.6	76.4	66.5
Normalized L1	1 - tanh(·)	**75.5**	74.3	79.4	76.5	66.4
Normalized L1	1 - B-sigmoid(·)	75.3	74.4	79.6	76.7	66.6

Effect of AADL Computation Strategies. We examine various formulations of AADL, including different distance metrics and loss functions (Tab. 20). A normalized L1 difference (i.e., mean absolute deviation) achieves the best performance and highest efficiency. All loss functions project differences in $(0, \infty)$ to monotonically decreasing values in $(0, 1)$, differing mainly in their slopes. As a result, the overall performance variation is minimal. We adopt $-\log(\tanh(\cdot))$ for its stable convergence and consistent performance.

Impact of Sampling Strategies for Action Spectral Features. To ensure the AADL can be computed meaningfully, all action segments must have a unified spectral length along the frequency dimension. Given that the raw spectral lengths of segments may vary—being either shorter or longer than the target length—we require a sampling strategy capable of both downsampling and upsampling. We investigate three such strategies: nearest-neighbor interpolation, linear interpolation, and adaptive average pooling, with results summarized in Tab. 21. Among them, nearest-neighbor interpolation performs the worst, as its downsampling process discards points, leading to potential aliasing, while its upsampling introduces staircase-like discontinuities that exaggerate adjacent action differences. Adaptive average pooling achieves moderate performance by smoothing upsampled sequences through averaging; however, it tends to excessively blur frequency details during downsampling, weakening inter-action discriminability. Linear interpolation achieves the best results, as it preserves critical spectral trends during both upsampling and downsampling while suppressing noise, thereby enabling discrepancy computation that better reflects the true physical differences between actions. Consequently, we adopt linear interpolation as our default sampling strategy.

Effect of Fixed Sampling Length S_f on Action Spec-

Table 21. Impact of the sampling strategies for action spectral features on PKU-MMD v2 (X-sub) dataset.

Sampling Strategies	Acc	Edit	F1@{10, 25, 50}		
Linear Interpolation	75.4	**74.5**	**79.7**	**76.8**	**66.6**
Nearest Neighbor	**75.6**	73.6	79.4	76.3	65.9
AdaptiveAvgPool	75.3	74.3	79.7	76.7	66.3

Table 22. Effect of fixed sampling length S_f on action spectral features on PKU-MMD v2 (X-sub) dataset.

Sampling Length	Acc	Edit	F1@{10, 25, 50}		
16	75.3	73.6	78.9	76.1	66.2
32	75.4	74.5	79.7	**76.8**	**66.6**
48	**75.5**	**74.6**	**79.8**	76.6	66.5
64	75.6	74.0	79.1	76.5	66.6

tral Features. We further examine the influence of the fixed sampling length S_f used for sampling action spectral features before computing AADL, as shown in Tab. 22. Fixing a unified spectral resolution is requisite for AADL to precisely align spectral axes for valid discrepancy comparisons. A length of $S_f = 16$ yields suboptimal performance due to insufficient frequency resolution, which leads to an inaccurate estimation of inter-action discrepancies. Increasing S_f to 32 and 48 improves performance significantly, as these lengths better capture essential spectral patterns while maintaining computational tractability. Crucially, even for extremely short action segments, linear interpolation to this fixed length effectively preserves the critical spectral envelope without introducing disruptive artifacts. Beyond this range, additional gains are marginal, because most of the shortest action segments in the dataset have inherent spectral lengths near 32–48. Notably, increasing S_f also leads to higher loss computation costs and slower convergence. Therefore, we select $S_f = 32$ as the default value, balancing accuracy and training efficiency.

Impact of Discrepancy Scaling Factor α in Adjacent Action Discrepancy Loss. AADL maps the raw inter-action spectral discrepancies—ranging over $(0, \infty)$—to a bounded loss in $(0, 1)$ through a monotonically decreasing function. This transformation inherently introduces a saturation effect, where overly large discrepancies yield near-zero gradients, limiting further optimization. Rather than a drawback, this saturation is intentionally designed to act as a "soft margin" that bypasses easy, highly distinct sample pairs, thereby concentrating the training gradients on ambiguous and hard-to-distinguish adjacent action boundaries. To control this saturation threshold and the effective amplification of discrepancies, we conduct an ablation on the scaling factor α, as presented in Tab. 23. We find that $\alpha = 100$ offers the best performance, effectively balancing sensitivity and gradient stability. In contrast, larger values ($\alpha = 200, 400$) lead to early saturation and hinder the model's ability to distinguish subtle action differences. Conversely, smaller values ($\alpha = 50$) amplify discrepancies excessively, disrupting optimization stability and slightly degrading performance. Based on these findings, we adopt $\alpha = 100$ as the optimal scaling factor.

Table 23. Impact of discrepancy scaling factor α in adjacent action discrepancy loss on PKU-MMD v2 (X-sub) dataset.

Scaling Factor α	Acc	Edit	F1@{10, 25, 50}		
50	75.0	**74.6**	79.3	76.1	66.4
100	**75.4**	74.5	**79.7**	**76.8**	**66.6**
200	75.3	73.9	79.2	76.3	66.0
400	75.0	74.3	79.6	76.7	65.9

Table 24. Effects of the weight λ_3 for adjacent action discrepancy loss on PKU-MMD v2 (X-sub) dataset.

Weight λ_3	Acc	Edit	F1@{10, 25, 50}		
0.1	75.4	74.4	79.6	76.7	66.3
1.0	75.4	74.5	**79.7**	**76.8**	66.6
5.0	**75.5**	74.5	79.6	76.7	**66.7**
10.0	75.3	**74.6**	79.7	76.8	66.6

Effect of the Loss Weight λ_3 for Adjacent Action Discrepancy Loss. We evaluate the impact of the AADL loss weight λ_3 to determine its contribution to the overall training objective, as summarized in Tab. 24. When $\lambda_3 = 0.1$, performance deteriorates, suggesting that the AADL signal is too weak to influence the optimization trajectory, with gradients dominated by other loss terms. Increasing λ_3 to 1 significantly improves performance, indicating that the discrepancy constraint is now sufficiently enforced to guide the model toward learning adjacent action differences. Further increasing λ_3 beyond 1 shows negligible additional benefit, implying that the loss has reached a stable contribution threshold. Therefore, we set $\lambda_3 = 1$ as the default weight in our final configuration.

9.4. Ablation Studies on FACM

Impact of Frequency-Aware Channel Mixer and Stacking Depth. To analyze the role of FACM, we ablate the FFT operation, channel mixing strategy, and stacking depth, as presented in Tab. 25. Channel mixing without FFT yields minimal improvement, as similar mixing is already inherent in Linear Transformers. In contrast, applying FFT alone brings moderate gains by suppressing high-frequency noise. When combined with channel mixing, the frequency-aware operation facilitates learning robust time-frequency representations, resulting in improved performance. Furthermore, increasing the point-wise depth improves performance, with gains nearly saturating at a depth of 2.

Ablation on Complex-Valued Feature Processing

Table 25. Impact of frequency-aware channel mixer and stacking depth on PKU-MMD v2 (X-sub) dataset.

FFT	PConv	Depth	Acc	Edit	F1@{10, 25, 50}		
✗	✗	-	75.1	73.9	79.3	76.3	66.0
✓	✗	-	75.2	74.2	79.7	76.6	66.3
✗	✓	2	75.1	74.1	79.3	76.4	66.1
✓	✓	1	75.3	74.4	79.4	76.2	66.4
✓	✓	2	75.4	**74.5**	**79.7**	**76.8**	66.6
✓	✓	3	**75.5**	74.5	79.7	76.7	**66.7**

Table 26. Ablation on complex feature processing in frequency-aware channel mixer on PKU-MMD v2 (X-sub) dataset.

Conv Strategy	Acc	Edit	F1@{10, 25, 50}		
Real-only	74.5	73.5	78.7	75.9	64.5
Imaginary-only	74.7	73.5	77.9	74.7	63.5
Shared Weights	**75.4**	**74.5**	**79.7**	**76.8**	**66.6**
Independent Weights	74.8	74.4	78.9	76.1	65.8

in the Frequency-Aware Channel Mixer. Since the Frequency-Aware Channel Mixer (FACM) operates in the frequency domain, where features are represented as complex numbers, it is critical to consider how complex-valued features are handled during channel mixing. We conduct an ablation study on four design choices: (i) mixing only the real part, (ii) mixing only the imaginary part, (iii) applying shared weights to both parts, and (iv) using independent weights for real and imaginary components. Results are presented in Tab. 26. The first two strategies yield the worst performance, as separately processing either part breaks the physical consistency of complex-valued signals and results in information loss or distortion. The independent-weight strategy achieves moderate performance but is inferior due to its decoupling of real-imaginary interactions, increased model complexity, and reduced generalization capacity. The shared-weight configuration achieves the best performance by preserving the unified representation of complex features, reducing optimization difficulty, and aligning with signal processing priors regarding coherent frequency-domain transformations. Therefore, we adopt the shared-weight design as the final mixing scheme in FACM.

Ablation on Post-Processing Operations for Frequency-Aware Mixed Features. We further investigate the effect of three commonly used post-processing operations—ReLU activation, Batch Normalization, and Dropout—on the output of FACM before the residual connection, as shown in Tab. 27. Introducing ReLU results in performance degradation, likely because the time-domain signals reconstructed from FACM outputs contain both positive and negative oscillatory components (e.g., phase-related information), and ReLU truncates negative values, violating signal integrity. Similarly, Batch Normalization leads to degraded performance, possibly due to its disruption of amplitude patterns, which undermines the consistency of frequency-aware mixed features. In contrast, adding Dropout shows negligible impact on performance, indicating that the FACM produces robust frequency-domain features where moderate random feature point omission does not substantially impair spectral representations. Consequently, we do not employ any post-processing operations following FACM.

Table 27. Ablation on post-processing operations for frequency-aware mixed features on PKU-MMD v2 (X-sub) dataset.

Post-Processing	Acc	Edit	F1@{10, 25, 50}		
Identity	75.4	**74.5**	**79.7**	**76.8**	**66.6**
ReLU	74.2	73.4	78.6	75.5	64.9
BatchNorm	75.1	74.3	78.9	76.3	65.4
Dropout(0.3)	**75.6**	74.5	79.6	76.5	66.5

Table 28. Impact of kernel size choices in frequency-aware channel mixer on PKU-MMD v2 (X-sub) dataset.

Kernel Size	Acc	Edit	F1@{10, 25, 50}		
1	**75.4**	**74.5**	**79.7**	**76.8**	**66.6**
3	74.5	73.9	78.1	75.3	64.6
5	73.8	74.2	78.5	75.7	64.6
7	73.8	74.4	78.9	75.4	64.5

Impact of Kernel Size Choices in Frequency-Aware Channel Mixer. By default, FACM employs point-wise convolution (i.e., kernel size = 1), enabling pure inter-channel interactions without modeling frequency-local dependencies. To explore whether broader receptive fields might benefit performance, we test larger kernel sizes in FACM, as reported in Tab. 28. Results show that increasing the kernel size consistently degrades performance. This can be attributed to the weak correlation and inherent physical independence among neighboring frequency bins; enforcing local interactions introduces spurious couplings, disturbs structured spectral representations, and injects non-physical noise. Thus, point-wise convolution is optimal for FACM, enabling efficient and semantically coherent cross-channel integration, while preserving orthogonality with the temporal modeling performed by the Linear Transformer branch.

Comparison of FFT Normalization Strategies in Frequency-Aware Channel Mixer. We evaluate the effect of different FFT normalization schemes in the FACM module, as summarized in Tab. 29. The tested configurations include: (i) forward normalization (scaling by $1/S$), (ii) backward mode (no normalization), and (iii) orthogonal normalization (scaling by $1/\sqrt{S}$). Corresponding inverse FFT operations are adjusted accordingly to maintain scale consistency. Experimental results show minimal per-

Table 29. Comparing FFT normalization strategies in frequency-aware channel mixer on PKU-MMD v2 (X-sub) dataset.

Normalization Mode	Acc	Edit	F1@{10, 25, 50}		
Forward (normalize by $1/S$)	**75.5**	**74.6**	**79.8**	76.7	66.6
Backward (no normalization)	75.4	74.5	79.7	**76.8**	**66.6**
Ortho (normalize by $1/\sqrt{S}$)	75.3	74.6	79.5	76.4	66.5

Table 30. Ablation on temporal block depth in temporal modeling on PKU-MMD v2 (X-sub) dataset.

block counts	Acc	Edit	F1@{10, 25, 50}		
6	72.1	71.0	76.7	73.4	62.1
8	74.2	73.2	78.5	75.1	64.5
10	75.4	74.5	79.7	76.8	66.6
12	75.7	**75.1**	**80.0**	77.0	66.6
14	**76.0**	75.0	79.6	**77.1**	**66.8**

formance differences across all strategies, suggesting that the point-wise convolution layer effectively learns to compensate for scale variations induced by normalization. This observation aligns with theoretical expectations: when FFT and iFFT normalization are symmetrically paired, and the model includes learnable linear transformations, any scale inconsistency can be absorbed by the model weights. For simplicity, we adopt the backward normalization setting as the default.

9.5. Other Ablation Studies

Ablation on Temporal Block Depth in Temporal Modeling. Following the design of previous STAS methods, our temporal modeling module is constructed by stacking multiple temporal blocks to progressively capture long-range temporal dependencies across frames. We investigate the effect of varying the depth of temporal block stacking, as shown in Tab. 30. Experimental results demonstrate that increasing the number of temporal blocks consistently improves performance, indicating enhanced capacity to model complex temporal relationships. However, deeper networks incur higher computational and parameter overhead. Notably, performance gains begin to plateau beyond 10 layers, suggesting diminishing returns due to optimization saturation. To balance accuracy and efficiency, we adopt a depth of 10 temporal blocks as the default configuration in our framework.

Ablation on Stage Numbers in Classification and Boundary Refinement Branches. We further examine the effect of stage numbers in the refinement branches responsible for classification prediction and boundary regression, respectively. As reported in Tab. 31, for the classification branch, using one or two refinement stages based on Linear Transformers yields nearly optimal performance, with a single stage already sufficient to provide effective prediction

Table 31. Ablation on the stage numbers of classification/boundary refinement branches on PKU-MMD v2 (X-sub) dataset.

Class stages	Boundary stages	Acc	Edit	F1@{10, 25, 50}		
1	1	**75.7**	74.1	78.2	75.3	66.2
1	2	75.4	**74.5**	**79.7**	76.8	66.6
2	2	75.3	74.5	79.6	**76.9**	**66.7**
2	3	74.8	74.0	79.6	75.9	66.0
3	3	74.3	74.5	79.5	76.2	65.3

refinement. Increasing the number of stages further leads to marginal or negative gains, likely due to overfitting. In contrast, the boundary refinement branch, which leverages dilated TCN layers, benefits from two refinement stages to reach peak performance, while deeper stacks again cause overfitting and degrade generalization. Considering both accuracy and computational cost, we employ a one-stage class prediction branch and a two-stage boundary regression branch in the final model.

10. Discussion

This section provides a reflective discussion on the present work. We candidly address the model's current limitations (Sec. 10.1), propose promising directions for future research (Sec. 10.2), and consider the broader societal impacts and ethical implications of this technology (Sec. 10.3).

10.1. Limitations

Despite its compelling performance, Spectral Scalpel is not without limitations. As shown in Fig. 5, the model exhibits instances of misclassification and boundary offsets, particularly in challenging scenarios. Specifically, detailed in our per-class analysis (Sec. 8.2), the model's performance degrades for low-frequency or quasi-static actions, where motion is subtle. Furthermore, the global frequency spectrum proves insufficient for distinguishing between actions that are spectrally similar but semantically distinct. This includes temporally reversed pairs (e.g., "Take off sth" vs. "Put on sth") and actions that differ only in repetition or magnitude (e.g., "Triple Lutz" vs. "Quadruple Lutz"). The primary challenge stems from the inherent limitations of using global FFT to capture spectra. Although FFT is effective and efficient, it is difficult to capture the full complexity of various human behaviors. In addition, unified learning and additional prior guidance for the core frequency representation of each action are also crucial. As the first systematic effort to introduce lightweight frequency analysis to Skeleton-based Temporal Action Segmentation (STAS), our work validates the approach's feasibility and effectiveness. However, capturing the intricate and varied motion patterns inherent in real-world behaviors remains a significant challenge due to the constraints of a global spectral

representation and consistent learning for core action spectra.

10.2. Future Work

To address these limitations, we have identified two primary directions for future research.

First, to better capture quasi-static actions and resolve spectral ambiguities, we plan to transition from a purely global frequency analysis to a collaborative time-frequency analysis framework. The inherent limitation of the global FFT is its lack of temporal localization. Therefore, future work will explore adaptive local filtering techniques, such as the Short-Time Fourier Transform (STFT) or Wavelet Transforms. By integrating these methods, our goal is to develop a model that can capture localized, time-varying frequency patterns, enabling a more nuanced understanding of how motion spectra evolve throughout an action sequence.

Second, we aim to enhance the learning paradigm and guidance mechanisms. The current data-driven filtering approach, while powerful, can be further optimized. The existing spectral discrepancy loss focuses solely on maximizing inter-class distance, which may not guarantee the learning of a stable and consistent frequency representation for each action. To this end, we propose incorporating contrastive learning and prototype learning. This will encourage the model to learn a canonical and robust frequency prototype for each action class by contrasting positive and negative spectral patterns. Furthermore, the purely data-driven model is susceptible to local optima due to its high optimization randomness. To mitigate this, we will investigate methods for integrating action-specific frequency priors. By injecting domain knowledge into the learning process, we can guide the model towards more meaningful and generalizable solutions, thereby improving its overall robustness and performance.

10.3. Broader Impacts

The proposed Spectral Scalpel framework for Skeleton-based Temporal Action Segmentation (STAS) offers promising societal benefits by enabling accurate, efficient, and privacy-conscious understanding of human motion, with applications in rehabilitation, elderly care, industrial monitoring, sports analysis, and human-computer interaction. By operating on skeletal data rather than raw video, the approach reduces privacy concerns and computational overhead, supporting broader and more ethical deployment. However, potential negative impacts include the misuse of STAS for surveillance, behavioral profiling risks when combined with other data sources, and fairness issues or safety hazards due to biased performance across demographic groups. Although this work is foundational, we recognize the importance of addressing these risks early. Mitigation strategies include promoting transparency in data use, auditing for bias, restricting deployment to consented and ethically reviewed settings, and implementing technical safeguards such as anonymization and access controls.

References

[1] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Juergen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *ECCV*, pages 52–68, 2022. 3

[2] Shurong Chai, Rahul Kumar Jain, Jiaqing Liu, Shiyu Teng, Tomoko Tateyama, Yinhao Li, and Yen-Wei Chen. A motion-aware and temporal-enhanced spatial–temporal graph convolutional network for skeleton-based human action segmentation. *Neurocomputing*, 580:127482, 2024. 1

[3] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *ICCV*, pages 13359–13368, 2021. 1

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 2, 4, 6

[5] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, pages 3575–3584, 2019. 3

[6] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *WACV*, pages 2322–2331, 2021. 3

[7] Haoyu Ji, Bowen Chen, Xinglong Xu, Weihong Ren, Zhiyong Wang, and Honghai Liu. Language-assisted skeleton action understanding for skeleton-based temporal action segmentation. In *ECCV*, pages 400–417. Springer, 2024. 3, 4, 6

[8] Haoyu Ji, Bowen Chen, Weihong Ren, Wenze Huang, Zhihao Yang, Zhiyong Wang, and Honghai Liu. Text-derived relational graph-enhanced network for skeleton-based action segmentation. *IEEE TIP*, 34:7305–7320, 2025. 1, 2, 3

[9] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165, 2020. 3

[10] Yuheng Li, Zhongyu Li, Shanghua Gao, Qilong Wang, Hou Qibin, and Cheng Mingming. A decoupled spatio-temporal framework for skeleton-based action segmentation. *arXiv preprint arXiv:2312.05830*, 2023. 1, 3, 6

[11] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for skeleton-based human action understanding. In *ACM VSCC*, pages 1–8, 2017. 5, 6

[12] Shenglan Liu, Aibin Zhang, Yunheng Li, Jian Zhou, Li Xu, Zhuben Dong, and Renhao Zhang. Temporal segmentation of fine-gained semantic action: A motion-centered figure skating dataset. In *AAAI*, pages 2163–2171, 2021. 5, 6

[13] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph con-

volutions for skeleton-based action recognition. In *CVPR*, pages 143–152, 2020. 1

[14] Friedrich Niemann, Christopher Reining, Fernando Moya Rueda, Nilah Ravi Nair, Janine Anika Steffens, Gernot A Fink, and Michael Ten Hompel. Lara: Creating a dataset for human activity recognition in logistics using semantic attributes. *Sensors*, 20(15):4083, 2020. 5, 6

[15] Yuki Tatsunami and Masato Taki. Fft-based dynamic token mixer for vision. In *AAAI*, pages 15328–15336, 2024. 11

[16] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[17] Julian Wiederer, Arij Bouazizi, Ulrich Kressel, and Vasileios Belagiannis. Traffic control gesture recognition for autonomous vehicles. In *IROS*, pages 10676–10683, 2020. 5

[18] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *BMVC*, 2021. 3