(Received 27 June 2026, Accepted 2 August 2026, Scheduled 20 August 2026)

Radar-Informed MSG-Transformer for Action Recognition with Sparse mmWave Radar Point Clouds

Su Liu[1], Jianguo Liu[2], Fei Gao[2], Jietao Cheng[1], and Jun Tang[1,*]

[1]School of Optoelectronics and Communications, Xiamen University of Technology, Xiamen 361000, Fujian, China
[2]Alcatel-Lucent Shanghai Bell Co., Ltd., Shanghai, China

ABSTRACT: Millimeter-wave (mmWave) radar supports non-contact action sensing without optical imagery, but its sparse point clouds vary with multipath, aspect angle, and detection quality. We address within-protocol recognition with a radar-informed Multi-Stream Gated Transformer (MSG-Transformer). Separate streams encode target-centered geometry, Doppler velocity, radar-reported SNR, and tracker-estimated horizontal-centroid context prior to temporal fusion. Raw trials are split before frame stacking, and augmentation perturbs only centered coordinates, leaving measured Doppler and SNR unchanged. On 820 archived trials from eight anonymized participant identifiers and nine action categories, MSG-Transformer achieved 97.67% development-set macro-recall with 1.03 million parameters and a 3.92 MiB FP32 footprint. This single-split result is not subject-independent or cross-environment evidence.

1. INTRODUCTION

Millimeter-wave (mmWave) radar enables Human-Computer Interaction (HCI) and Ambient Assisted Living (AAL) under poor illumination without recording optical appearance [1, 2]. Although related sensing hardware is studied in Integrated Sensing and Communications (ISAC), this paper addresses action recognition [3, 4].

Point-cloud action recognition has mainly used vision or LiDAR-like data, including 3DInAction [5], whereas mmWave radar is sparse, discrete, and non-stationary [1, 6, 7]. Signal dropouts, Doppler ambiguity, and specular reflections further disrupt local geometric continuity assumed by many point-cloud models [2, 6, 8].

PST^2 [9], CSP-Former [10], and P4Transformer [11] model long-range or local spatiotemporal dependencies, but were developed mainly for dense geometric inputs without separated radar attributes [6]. Dot-product attention also depends on learned query-key norms, making temporal correlation sensitive to the magnitude of latent features. This feature-space effect does not imply physical radar cross-section (RCS). Doppler and measurement-quality cues are informative [12, 13] and are retained in datasets, such as K-Radar and RADIATE [14, 15].

MSG-Transformer separately projects spatial coordinates, Doppler, target-position context, and stored SNR [12, 16]. A Swish Gated Linear Unit (SwiGLU) fuses the projections [17], while Scaled Cosine Attention normalizes queries and keys before correlation [18]. Ablations test these radar-informed biases; they do not recover calibrated RCS, remove multipath, or simulate radar measurements.

Multi-stream processing itself is not claimed as a domain-independent novelty. The radar-specific contribution is the coordinated design of semantic streams aligned with sparse mmWave measurements, target-centered augmentation that leaves measured Doppler and SNR unchanged, and norm-controlled temporal attention with local depthwise convolution. The feature and component ablations evaluate this combination under the archived protocol, and the absence of a matched-parameter single-stream control remains a limitation.

The study makes two contributions. First, it documents an archived multimodal radar action collection containing geometry, Doppler velocity, radar-reported SNR, point-to-centroid distance, and target-centroid context, with raw trials split before window generation. Second, it evaluates a compact MSG-Transformer combining semantic streams, SwiGLU fusion, Scaled Cosine Attention, local depthwise convolution, and restricted coordinate augmentation. Comparative, ablation, and complexity results are reported under a deliberately narrow single-split development protocol.

2. RELATED WORK

For privacy-sensitive HCI and AAL, mmWave radar avoids the illumination dependence and identifiable imagery of optical cameras [1, 6] and shares sensing concepts with ISAC systems [4]. Its sparse returns vary with aspect and motion, making radial velocity and measurement quality informative complements to geometry.

2.1. Radar Data Representation Formats

Early radar-recognition pipelines converted returns into Range-Doppler maps or micro-Doppler spectrograms and then applied convolutional neural network (CNN) classifiers. These inputs capture motion signatures but discard much of the 3D scattering layout, including limb position [6]. Multiple-input multiple-output (MIMO) radar can produce point clouds, al-

* Corresponding author: Jun Tang (jtang@xmut.edu.cn).

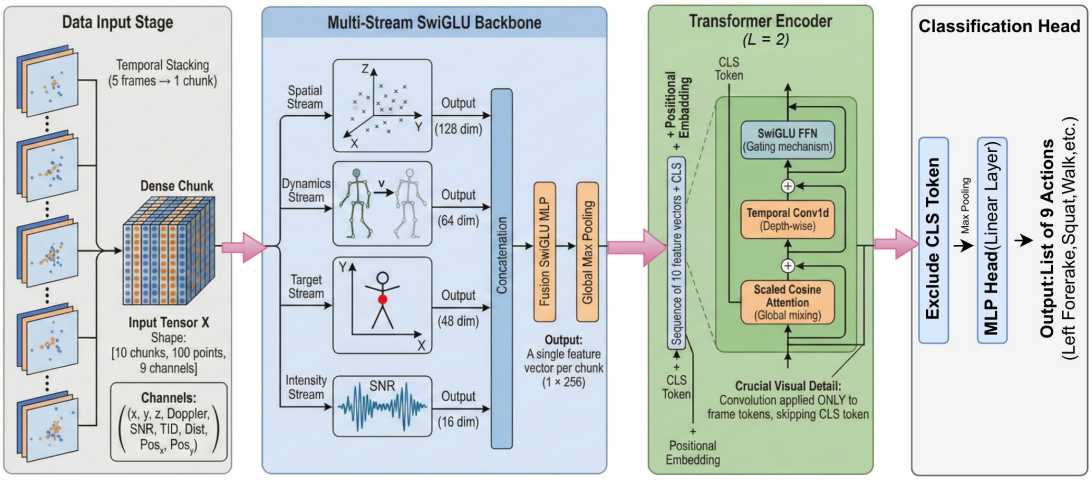

FIGURE 1. Architecture of MSG-Transformer. The input block shows the stored 9-column preprocessing tensor, while the semantic streams use the effective radar channels described in Eqs. (3)–(4).

though some public datasets retain only LiDAR-style geometry and omit Doppler or SNR [14, 15]. Those attributes aid target-clutter separation, especially for weak limb returns and multipath-affected points [8, 12]. A recognition model should therefore use them when the archive provides them [19].

2.2. Deep Representation Methods for Point Clouds

Point-cloud networks provide the building blocks for radar perception [20, 21]. PointNet uses shared MLPs for permutation-invariant point processing [22]; PointNet++ adds hierarchical sampling for local features [23]; and DGCNN models local topology through dynamic graphs [21, 24]. Transformer-based point models, including Point Transformer and Point Cloud Transformer (PCT), aggregate global context by self-attention and perform strongly on geometric benchmarks [25–28]. The radar-specific gap is not whether attention can model points, but whether the architecture respects the distinct semantics of coordinates, radial velocity, tracker context, and measurement quality.

2.3. Radar-Attribute Challenges

Radar action recognition often uses tracking-then-classification pipelines or PointNet-RNN cascades, which may weakly model echo-intensity variation, Doppler velocity, and target-level motion context [6, 8]. Vision-style Transformers also require adjustment. Standard Scaled Dot-Product Attention depends on query-key magnitudes [26]; normalization stabilizes the similarity scale without providing physical RCS compensation. Augmentation raises a separate issue because rotating coordinates while retaining measured radial velocity changes the implied viewing geometry [12, 13]. MSG-Transformer therefore keeps radar attributes explicit, restricts coordinate perturbations, and uses Scaled Cosine Attention only as feature-space norm control [18].

3. THE MSG-TRANSFORMER FRAMEWORK

Figure 1 outlines three stages: preprocessing with restricted augmentation, a multi-stream gated backbone, and a temporal Transformer with Scaled Cosine Attention. Section 3.4 specifies the recognition implementation.

3.1. Multimodal Radar Feature Integration and Augmentation

The acquisition protocol records geometry, kinematics, signal intensity, association distance, and target trajectory context. For point $i \in \{1, \ldots, n_t\}$ at frame t, the 9D feature is

$$\mathbf{f}_{t,i}^{\text{acq}} = [x_{t,i}, y_{t,i}, z_{t,i}, v_{t,i}, S_{t,i}, d_{t,i}, X_t, Y_t, Z_t] \in \mathbb{R}^9, \quad (1)$$

Here, $\mathbf{f}_{t,i}^{\text{acq}}$ is the acquisition feature vector; t is the frame index; n_t is its number of points; i is the within-frame point index ;$(x_{t,i}, y_{t,i}, z_{t,i})$ are Cartesian coordinates; $v_{t,i}$ is Doppler radial velocity; $S_{t,i}$ is radar-reported SNR; $d_{t,i}$ is point-to-centroid distance; and (X_t, Y_t, Z_t) is the associated tracked-target centroid.

Before sample generation, the Cartesian coordinates are converted into target-centered coordinates to reduce dependence on absolute radar placement:

$$\tilde{x}_{t,i} = x_{t,i} - X_t, \quad \tilde{y}_{t,i} = y_{t,i} - Y_t, \quad \tilde{z}_{t,i} = z_{t,i} - Z_t. \quad (2)$$

The tilde denotes target centering; archived columns satisfy $pos_{x,t} = X_t$, $pos_{y,t} = Y_t$, and $pos_{z,t} = Z_t$.

The preprocessed point is then stored as a 9-column tensor

$$\hat{\mathbf{p}}_{t,i} = \left[\tilde{x}_{t,i}, \tilde{y}_{t,i}, \tilde{z}_{t,i}, D_{t,i}, \text{SNR}_{t,i}, \text{tid}_{t,i}, d_{t,i}, pos_{x,t}, pos_{y,t}\right], \quad (3)$$

Here, $\hat{\mathbf{p}}_{t,i} \in \mathbb{R}^9$ is the stored point vector; $D_{t,i} = v_{t,i}$ is the Doppler; and $\text{SNR}_{t,i} = S_{t,i}$ is the stored signal-to-noise ratio (SNR). The integer $\text{tid}_{t,i}$ is a temporary tracker identifier, and the columns $d_{t,i}$ and $(pos_{x,t}, pos_{y,t})$ retain point-to-centroid distance and horizontal target context. This nine-column layout preserves compatibility with the data loader and Figure 1, while the Multi-Stream Gated Backbone reads only

$$\mathbf{p}_{t,i} = \left[\tilde{x}_{t,i}, \tilde{y}_{t,i}, \tilde{z}_{t,i}, D_{t,i}, \text{SNR}_{t,i}, pos_{x,t}, pos_{y,t}\right] \in \mathbb{R}^7. \quad (4)$$

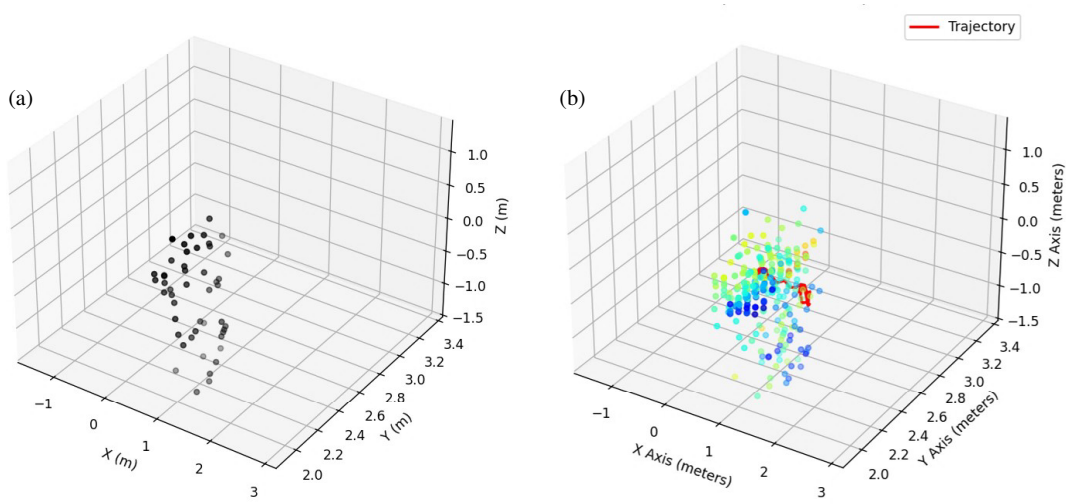

FIGURE 2. Spatiotemporal enhancement of sparse point clouds. (a) Raw single-frame geometry and (b) five-frame stacked multimodal point cloud, where Doppler color indicates motion direction, and the red curve represents the target trajectory.

$\mathbf{p}_{t,i}$ is the seven-dimensional classifier input. Tracker ID is excluded, and point-to-centroid distance remains only an acquisition and preprocessing attribute. Likewise, $pos_{z,t}$ is used for target centering in Eq. (2), and the target stream receives only $(pos_{x,t}, pos_{y,t})$. This prevents identifiers and unused fields from becoming class cues. Restricted yaw rotation, scaling, and jitter affect only centered coordinates; unchanged Doppler makes this geometry-only regularization rather than physical view simulation [12, 29].

Radar physics motivates these features. Suppressing the frame index, $\mathbf{c}_i = [x_i, y_i, z_i]^T$ describes the scattering location, and Doppler relates to inter-chirp phase variation:

$$v_i = \frac{\lambda \Delta\phi_i}{4\pi T_c}, \tag{5}$$

where λ, $\Delta\phi_i$, and T_c denote the wavelength, measured phase difference, and chirp interval. Received echo power is

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4 L}, \tag{6}$$

where P_t, G_t, G_r, σ, R, and L denote transmitted power, antenna gains, radar cross-section, range, and aggregate loss. SNR is

$$\mathrm{SNR}_i^{\mathrm{dB}} = 10 \log_{10} \left(\frac{P_{r,i}}{P_n} \right), \tag{7}$$

Here, $\mathrm{SNR}_i^{\mathrm{dB}}$ is the point-level SNR, and $P_{r,i}$ and P_n are received and noise power. In physical units, SNR can be viewed as a noisy radiometric confidence measurement. Texas Instruments (TI)-detected-point side information may use 0.1 dB units, and some tracking interfaces export a linear field [30]. Because the archive lacks the originating TLV type and unit metadata, we report stored device units without dB conversion. SNR is therefore a processed measurement-quality attribute, not calibrated RCS.

3.1.1. Kinematic Enhancement and Temporal Trajectory Fusion

Single frames contain limited motion information [2, 7]. Stacking combines $M = 5$ frames, or approximately 0.5 s at 10 fps, into a denser chunk while retaining $T = 10$ ordered chunks and limiting each sample to 50 frames. For stored point set $\mathcal{P}_t = \{\hat{\mathbf{p}}_{t,i}\}_{i=1}^{n_t}$, chunk m is

$$\mathcal{C}_m = \bigcup_{j=(m-1)M+1}^{mM} \mathcal{P}_j, \tag{8}$$

Here, $\mathcal{C}_m$ is the point set in chunk m, $m \in \{1, \ldots, T\}$ is the chunk index; T is the number of chunks; M is the number of raw frames per chunk; and j is the raw-frame index. The union covers the indicated frame range. A resampling operator $\mathcal{R}_N(\cdot)$ then maps each variable-size chunk to

$$\tilde{\mathcal{C}}_m = \mathcal{R}_N(\mathcal{C}_m), \quad \tilde{\mathcal{C}}_m \in \mathbb{R}^{N \times 9}. \tag{9}$$

$\mathcal{R}_N$ samples without replacement when at least N points exist and otherwise repeats points. The final sample $\mathbf{X} = [\tilde{\mathcal{C}}_1, \ldots, \tilde{\mathcal{C}}_T] \in \mathbb{R}^{T \times N \times 9}$ uses $T = 10$ and $N = 100$. Figure 2 compares a raw frame with this five-frame representation; Doppler color indicates the motion direction [2]. Horizontal centroids provide global target anchors, while centered coordinates describe body-relative scattering. Combining local Doppler cues with global displacement reduces clutter-induced ambiguity.

3.1.2. Signal Intensity Distribution and Gated Feature Fusion

Multipath can create strong ghost detections, and limb returns can be weak [8]. The model therefore keeps radar-reported SNR as a separate input, instead of applying a hard cutoff. Figure 3(a) shows values above 300 stored units in strong-return regions and values below 150 elsewhere. The distribution in Figure 3(b) has mean $\mu \approx 205$ and standard deviation $\sigma \approx 72$. Weak detections may still encode limb motion, so thresholding could remove useful evidence together with clutter.

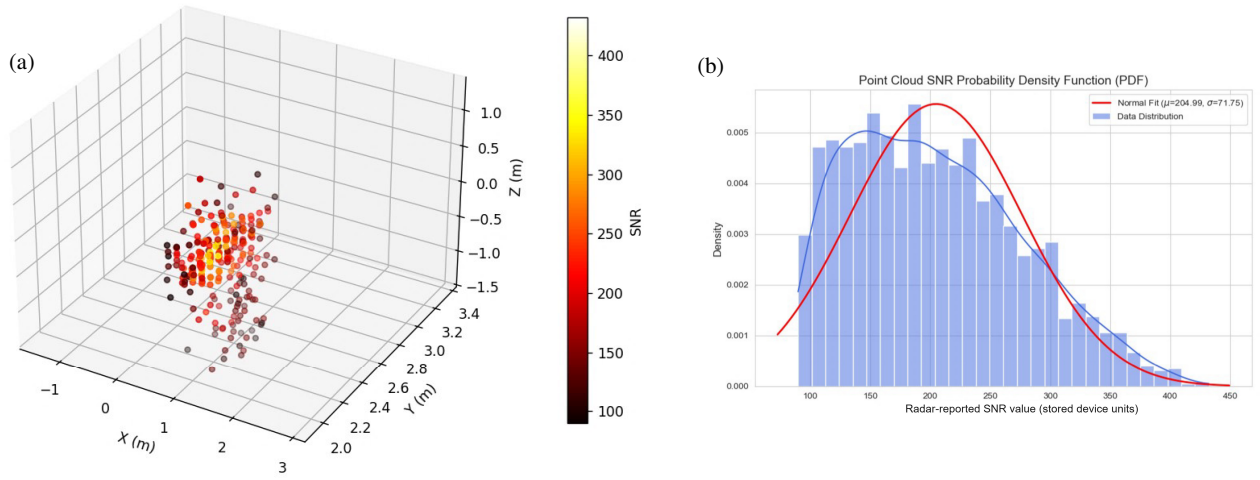

FIGURE 3. Distribution of radar-reported point-cloud SNR in stored device units. The distribution contains both strong returns and weaker detections, motivating learned intensity encoding rather than hard thresholding. (a) Spatial intensity heatmap and (b) Probability density in stored device units.

SNR is not an explicit multiplicative gate. A two-layer intensity Multilayer Perceptron (MLP) first maps it into a latent embedding, which is concatenated with geometry, Doppler, and target-position embeddings. The subsequent SwiGLU block performs nonlinear gated fusion over the full latent vector. The ablation supports the intensity stream and gated fusion as implemented, but it does not demonstrate monotonic suppression of low-SNR points.

3.2. Multi-Stream Gated Backbone

Let the stacked radar point-cloud sample be $\mathcal{X} = (\tilde{\mathcal{C}}_1, \ldots, \tilde{\mathcal{C}}_T)$. The Multi-Stream Gated Backbone separates the modalities and encodes each chunk as F'_t. Geometry, Doppler, centroid motion, and stored SNR have different ranges and semantics, so they receive separate projections before fusion. This separation is an architectural bias rather than proof that raw concatenation is always inferior.

For resampled point $i \in \{1, \ldots, N\}$ in chunk $\tilde{\mathcal{C}}_t$, the effective semantic channels are separated as

$$\begin{aligned} u_{t,i}^{xyz} &= [\tilde{x}_{t,i}, \tilde{y}_{t,i}, \tilde{z}_{t,i}], \quad u_{t,i}^{dyn} = [D_{t,i}], \\ u_{t,i}^{snr} &= [\mathrm{SNR}_{t,i}], \quad u_{t,i}^{pos} = [pos_{x,t}, pos_{y,t}]. \end{aligned} \tag{10}$$

Here, $u_{t,i}^{xyz}$, $u_{t,i}^{dyn}$, $u_{t,i}^{snr}$, and $u_{t,i}^{pos}$ denote the geometry, dynamics, SNR, and target-position input vectors. Their superscripts identify streams rather than mathematical powers. The indices and measurements retain the definitions in Eqs. (2)–(4), so the stream decomposition introduces no additional observed variables.

The four streams independently project these heterogeneous measurements into latent spaces with different capacities:

$$\begin{aligned} h_{t,i}^{xyz} &= \phi_{xyz}(u_{t,i}^{xyz}) \in \mathbb{R}^{128}, \\ h_{t,i}^{dyn} &= \phi_{dyn}(u_{t,i}^{dyn}) \in \mathbb{R}^{64}, \\ h_{t,i}^{pos} &= \phi_{pos}(u_{t,i}^{pos}) \in \mathbb{R}^{48}, \\ h_{t,i}^{snr} &= \phi_{snr}(u_{t,i}^{snr}) \in \mathbb{R}^{16}. \end{aligned} \tag{11}$$

In Eq. (11), ϕ_{xyz}, ϕ_{dyn}, ϕ_{pos}, and ϕ_{snr} are learnable MLP mappings, and the corresponding h vectors are their latent outputs. Their empirical widths sum to 256. The allocation gives sparse geometry the largest capacity, represents Doppler as compact kinematics, uses target position for global trajectory context, and treats SNR primarily as a measurement-quality cue. These widths are not physically derived constants, and a matched-parameter single-stream control is still needed to isolate stream separation.

Instead of concatenating raw measurements directly, MSG-Transformer projects each modality and then fuses the latent embeddings via a compact nonlinear block. The concatenated vector is

$$\mathbf{a}_{t,i} = \left[h_{t,i}^{xyz}; h_{t,i}^{dyn}; h_{t,i}^{pos}; h_{t,i}^{snr}\right] \in \mathbb{R}^{256}. \tag{12}$$

where $\mathbf{a}_{t,i}$ is the fusion input for point i in chunk t; the semicolon denotes channel-wise concatenation, and $256 = 128 + 64 + 48 + 16$. The fused point representation is

$$\mathbf{z}_{t,i} = \mathrm{SwiGLU}\left(\mathrm{GELU}\left(\mathrm{LN}\left(W_f \mathbf{a}_{t,i} + b_f\right)\right)\right) \in \mathbb{R}^C, \tag{13}$$

where $\mathbf{z}_{t,i}$ is the fused point embedding, $C = 256$, and W_f and b_f are learnable. LN normalizes channels, and GELU is applied element-wise. The streams do not read stored tid or distance columns. Permutation-invariant point aggregation then uses channel-wise max pooling:

$$F'_t = \max_{i \in \{1,\ldots,N\}} \mathbf{z}_{t,i} \in \mathbb{R}^C. \tag{14}$$

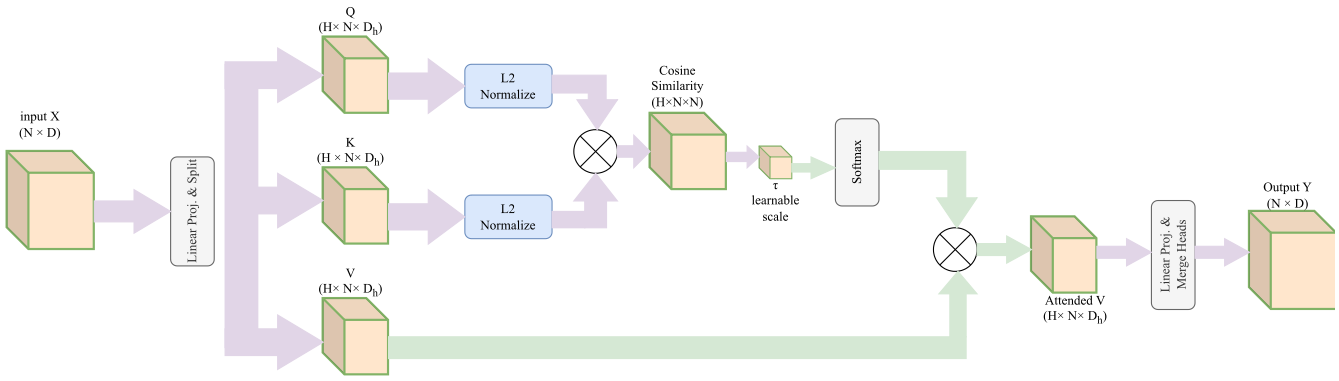

FIGURE 4. Diagram of the scaled cosine attention mechanism.

Thus, F'_t is the chunk-level feature, with the maximum evaluated independently in each of the C channels over all N points.

For latent vector x, the Swish Gated Linear Unit (SwiGLU) implements multiplicative nonlinear fusion as

$$\begin{aligned}\mathrm{SwiGLU}(x) &= \mathrm{LN}\left(W_3\left[\mathrm{SiLU}(W_1x+b_1)\right.\right.\\ &\quad\left.\left.\otimes(W_2x+b_2)\right]+b_3\right).\end{aligned} \tag{15}$$

W_1 and W_2 project $x \in \mathbb{R}^{C_{\text{in}}}$ to gated width C_g, and W_3 maps their product to the output width. $\mathrm{SiLU}(r) = r\sigma(r)$ is element-wise; σ is sigmoid, $\otimes$ denotes element-wise multiplication; and LN is the channel-wise normalization. The fusion block uses $C_{\text{in}} = C_{\text{out}} = 256$ and $C_g = 170$. Because SNR is part of the latent vector, the gate can learn intensity interactions, but no monotonic SNR-gate relation nor direct physical interpretation is imposed. The chunk embeddings form

$$E = [F'_1, F'_2, \ldots, F'_T] \in \mathbb{R}^{T\times C}, \tag{16}$$

Here, E is the ordered temporal feature matrix, and F'_T is the T-th chunk feature from Eq. (14). The model uses $T = 10$ chunks and feature width $C = 256$; the brackets stack the chunk vectors as temporal rows.

3.3. Temporally Gated Transformer

The temporal module combines norm-stabilized global attention with local temporal convolution. These are neural modeling choices motivated by radar-measurement variability and do not treat latent-vector norms as calibrated radar amplitudes. A learnable classification token e_{cls} and positional embedding E_{pos} are added:

$$Z^{(0)} = [e_{cls}; F'_1; \ldots; F'_T] + E_{pos} \in \mathbb{R}^{(T+1)\times C}. \tag{17}$$

Here, $Z^{(0)}$ is the input token matrix before the first Transformer layer; $e_{cls} \in \mathbb{R}^C$ is the learnable classification token; and $F'_T \in \mathbb{R}^C$ represents temporal chunk T. $E_{pos} \in \mathbb{R}^{(T+1)\times C}$ is learnable. The semicolon denotes token-wise concatenation, and $T+1$ counts one classification token plus T chunk tokens of width $C = 256$. For the ℓ-th Transformer layer and the h-th attention head, query, key, and value matrices are computed as

$$\begin{aligned}U^{(\ell)} &= \mathrm{LN}(Z^{(\ell)}),\quad Q_h = U^{(\ell)}W_h^Q,\\ K_h &= U^{(\ell)}W_h^K,\quad V_h = U^{(\ell)}W_h^V.\end{aligned} \tag{18}$$

Here, $Z^{(\ell)}$ enters layer ℓ; $U^{(\ell)}$ is its normalized form; Q_h, K_h, and V_h are the query, key, and value matrices for head h, produced by learned projections W_h^Q, W_h^K, and W_h^V. The implementation uses $L = 2$ layers, $H = 4$ heads, and head dimension $d_h = 64$. Each head therefore maps a 256-dimensional token into a 64-dimensional attention subspace.

3.3.1. Scaled Cosine Attention

Scaled Cosine Attention stabilizes learned feature-space similarity through row-wise query-key normalization [18], not calibrated radar-amplitude modeling:

$$\begin{aligned}A_h &= \mathrm{Attention}_h(Q_h, K_h, V_h)\\ &= \mathrm{Softmax}\left[\tau\left(\frac{Q_h}{\|Q_h\|_2}\right)\left(\frac{K_h}{\|K_h\|_2}\right)^T\right]V_h.\end{aligned} \tag{19}$$

The ℓ_2 norms are computed row-wise over the head dimension, normalizing each token query and key before pairwise dot products. A shared learnable temperature $\tau > 0$ is initialized to 10. Values remain unnormalized, so Scaled Cosine Attention changes the logits without removing all feature-amplitude information. Figure 4 illustrates normalization, cosine correlation, and value aggregation. Multi-head fusion is

$$\mathrm{MHCA}(U^{(\ell)}) = \mathrm{Concat}(A_1, \ldots, A_H)W^O \in \mathbb{R}^{(T+1)\times C}. \tag{20}$$

The attention residual update for layer ℓ is

$$\bar{Z}^{(\ell)} = Z^{(\ell)} + \mathrm{MHCA}(U^{(\ell)}). \tag{21}$$

Table 6 tests its empirical contribution.

3.3.2. Integrated Temporal Convolution

A depth-wise temporal convolution follows attention to extract local dynamics [31, 32]. The classification token is preserved, and a kernel-size $k = 3$ convolution is applied only to frame tokens:

$$\tilde{Z}^{(\ell)}_{1:T} = \bar{Z}^{(\ell)}_{1:T} + \mathrm{DWConv}_{k=3}\left(\mathrm{LN}\left(\bar{Z}^{(\ell)}_{1:T}\right)\right). \tag{22}$$

The processed frame features are recombined with the classification token:

$$\tilde{Z}^{(\ell)} = [\bar{Z}^{(\ell)}_{\text{cls}}; \tilde{Z}^{(\ell)}_{1:T}]. \tag{23}$$

A gated feed-forward block then updates the token matrix:

$$\Phi_{\text{ff}}(x) = \text{Dropout}\Big(W_{3,\text{ff}}\Big[\text{SiLU}(W_{1,\text{ff}}x + b_{1,\text{ff}}) \otimes (W_{2,\text{ff}}x + b_{2,\text{ff}})\Big] + b_{3,\text{ff}}\Big), \tag{24}$$

$$Z^{(\ell+1)} = \tilde{Z}^{(\ell)} + \Phi_{\text{ff}}\left(\text{LN}(\tilde{Z}^{(\ell)})\right).$$

The feed-forward width is $C_{\text{ff}} = 128$, and dropout is 0.4. These operations form the Temporally Gated Transformer: cosine attention provides norm-invariant query-key correlation, depthwise convolution models local continuity, and SwiGLU supplies gated nonlinear transformation. Applying convolution after global attention allows each layer to compare all chunks before refining adjacent temporal changes.

3.4. Implementation of MSG-Transformer for 3D Action Recognition

MSG-Transformer treats action recognition as point-cloud sequence classification. Each sample contains $T = 10$ chunks, each stacking $M = 5$ frames and sampling or padding to $N = 100$ points. The resulting $T \times N \times 9$ stored tensor covers 50 raw frames. Short recordings repeat cyclically, and long recordings use a 25-frame stride. Both operations follow split assignment, preventing windows from one raw file from crossing data splits.

Although the sequence includes a learnable Classification Token (CLS), the classifier temporally max-pools only frame tokens:

$$F_{\text{global}} = \max_{t\in\{1,\ldots,T\}} Z_t^{(L)} \in \mathbb{R}^C. \tag{25}$$

The resulting feature is mapped to $K = 9$ action classes:

$$\hat{\mathbf{y}} = \text{Softmax}(W_o F_{\text{global}} + b_o) \in [0,1]^K. \tag{26}$$

Training uses cross-entropy with label smoothing $\epsilon = 0.2$ [33]. For ground-truth class y, the target is

$$q_k = (1-\epsilon)\mathbb{I}(k = y) + \frac{\epsilon}{K}, \tag{27}$$

and the training loss is

$$\mathcal{L} = -\sum_{k=1}^{K} q_k \log \hat{y}_k. \tag{28}$$

4. EXPERIMENTS

4.1. Dataset, Acquisition Protocol, and Radar Configuration

Data were acquired with a Texas Instruments IWR6843-ISK mmWave radar. The Frequency Modulated Continuous Wave (FMCW) sensor operates in the 60–64 GHz band and was controlled through a MATLAB interface adapted from the TI Area Scanner Demo. Processing comprised range fast Fourier transform (FFT), static-clutter mean cancellation, Doppler FFT, Constant False Alarm Rate (CFAR) detection in the range and Doppler domains, field-of-view filtering, multi-object beamforming, and azimuth/elevation estimation. Table 1 lists the archived settings.

TABLE 1. Radar configuration used for data acquisition.

Parameter	Setting
Radar sensor	TI IWR6843-ISK
Antenna configuration	3 Tx/4 Rx
Signal waveform	FMCW
Operating frequency	60.5 GHz
Signal bandwidth	2.13 GHz
Frame rate	10 fps
Range resolution	7.03 cm
Azimuth resolution	approximately 15°

For each detected point, radial distance r, radial velocity v, azimuth angle θ, and elevation angle ϕ are estimated first. Cartesian coordinates follow the TI convention: the Y axis denotes forward range, the X axis lateral displacement, and the Z axis height:

$$x_i = r_i \sin\theta_i \cos\phi_i, \quad y_i = r_i \cos\theta_i \cos\phi_i, \quad z_i = r_i \sin\phi_i. \tag{29}$$

Here, i indexes a detected point; $r_i \geq 0$ is radial range; and (x_i, y_i, z_i) are Cartesian coordinates. The range is used to generate these coordinates, while model inputs follow the acquisition and effective-channel definitions in Eqs. (1)–(4).

All trials were conducted in a laboratory with desks, chairs, a host computer, mounting fixtures, and the radar sensor. The archive has no separate cross-environment, cross-distance, or cross-azimuth protocol. It also lacks identifiers for alternative layouts, controlled clutter, sensor height, or sensor orientation. Consequently, the random trial split measures within-protocol behavior and cannot be reinterpreted as cross-environment validation.

The archived filenames contain eight anonymized participant identifiers. Identifiers 1–5 performed seven action classes: *Left forerake*, *Left hand wave*, *Open arms*, *Right forerake*, *Right hand wave*, *Squat*, and *Walk*. Identifiers 7–9 performed *Standing up* and *Sitting down*; Identifier 6 was not used in the archived naming sequence. Each participant repeated every assigned action 20 times. This gives 820 raw trials and a nominal total of 41,000 frames, as detailed in Table 2. All participants were informed of the acquisition procedure and participated voluntarily. The records contain anonymized radar point clouds and include neither optical imagery nor speech recordings. The action-dependent participant assignment creates a class-subject confound, so the dataset is not suitable for claiming subject-independent performance. The archive also contains no simultaneous multi-person action labels. Temporary tracker identifiers associate points during preprocessing; they are not participant identities, and the final classifier produces one clip-level action label rather than separate labels for multiple tracked people.

TABLE 2. Action categories and raw-trial split statistics. The names in parentheses are the original directory names used by the archived code.

Action label	Raw trials	Train	Development (Test)	Held-out (Ver)
left_forerake	100	80	10	10
left_hand_wave	100	80	10	10
open_arms	100	80	10	10
right_forerake	100	80	10	10
right_hand_wave	100	80	10	10
sitting_down	60	48	6	6
squat	100	80	10	10
standing_up	60	48	6	6
walk	100	80	10	10
Total	**820**	**656**	**82**	**82**

The preprocessing script applies a per-action random raw-file split with seed 42 and an 8:1:1 ratio, writing the subsets as `Train`, `Ver`, and `Test`. The archived training script optimizes on `Train`, evaluates on `Test` after each epoch, and selects `best_model.pth` from that score. The directory named `Test` is therefore a development set. The `Ver` directory is not accessed by the training loop and is used only as a held-out audit. Splitting is done before frame stacking and sliding-window generation, so windows from the same raw recording cannot cross subsets. Participant-level overlap remains possible. Generated-sample counts may also differ from raw-trial counts because long recordings yield overlapping windows and short recordings are cyclically padded inside their assigned subset.

At the acquisition-protocol level, one labeled point record can be summarized as

$$\mathbf{r}_{t,i}^{rec} = \left[t, x_{t,i}, y_{t,i}, z_{t,i}, v_{t,i}, S_{t,i}, d_{t,i}, pos_{x,t}, pos_{y,t}, pos_{z,t}, \kappa_t\right], \tag{30}$$

where $\mathbf{r}_{t,i}^{rec}$ is one labeled acquisition record; t is the frame index; and i is the point index within that frame. The quantities $x_{t,i}$, $y_{t,i}$, $z_{t,i}$, $v_{t,i}$, $S_{t,i}$, and $d_{t,i}$ are respectively Cartesian position, radial velocity, stored SNR, and point-to-centroid distance as defined earlier. The values $pos_{x,t}$, $pos_{y,t}$, and $pos_{z,t}$ are the target-centroid coordinates at frame t, and $\kappa_t \in \{1, \ldots, K\}$ is the action-class label, with $K = 9$. Frame index is used for temporal ordering, labels are used only for supervision, and tracker ID is excluded from classifier inputs. The archive does not retain tracker uncertainty or a separate manual validation of the centroid trajectory, so centroid context should be interpreted as a tracker-derived auxiliary cue rather than a ground-truth body trajectory.

All models use PyTorch. MSG-Transformer has a fused width of 256 with stream widths of 128, 64, 48, and 16 for centered geometry, Doppler, horizontal centroid context, and SNR. The temporal module has two Transformer layers, four attention heads of dimension 64, feed-forward width 128, depthwise temporal-convolution kernel 3, and dropout 0.4. The classifier maps the pooled 256-dimensional representation to nine classes. Training uses a batch size of 32 for 100 epochs. Adam parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, initial learning rate 10^{-4}, and weight decay 5×10^{-4}. Gradient norm is clipped at 1.0. Cosine annealing uses $T_{\max} = 100$ and minimum learning rate 10^{-6}, with label smoothing 0.2. Samples contain 10 chunks, 100 points per chunk, and 9 stored channels; the model reads the seven channels in Eq. (4). Augmentation uses yaw $\pm 15°$, scale $[0.95, 1.05]$, and Gaussian jitter with standard deviation of 0.005 clipped to ± 0.02. The raw-file split seed is 42. Because the archived code does not set global PyTorch or NumPy seeds for initialization and point resampling, the retained run is not fully deterministic.

4.2. Comparative Experiments

The comparison uses PointNet and PointNet++ as spatial baselines and P4Transformer plus an in-house SequentialPointNet as spatiotemporal baselines. All methods use the raw-file split and sample-generation protocol in Section 4.1. Centered coordinates $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ form the diagnostic coordinate-only input. Radar-feature variants use the same non-ID attributes when their input layers support additional channels, with the first projection adjusted for channel count. The archived comparison contains one run per setting and no repeated-seed uncertainty. It also lacks exhaustive equal-compute or equal-parameter tuning for all baselines. MSG-Transformer excludes tid as defined in Eq. (4).

Table 3 shows that MSG-Transformer gives the highest archived single-run development score. Figure 5 provides the corresponding per-class comparison. Spatial-only PointNet-family models remain weaker because they do not explicitly model temporal dependencies, while P4Transformer and SequentialPointNet benefit from temporal modeling. MSG-Transformer reaches a 97.67% macro recall, approximately 4.8 percentage points above the strongest baseline, supporting within-protocol competitiveness but not cross-subject superiority or statistical significance.

Table 4 separates gains from radar measurements and preprocessing fields. Doppler-only and SNR-only inputs lack geometric context, while centered geometry alone remains weaker than the full semantic input. Removing horizontal centroid context also reduces the development score. Tracker-ID and stored-distance rows are controls; the final model excludes both fields from its semantic streams.

TABLE 3. Single-run development comparison under the same data preparation protocol. Non-ID radar features exclude temporary tracker identifiers. Scores are percentages; repeated-seed uncertainty is unavailable.

Model	Input channels	Development score
PointNet	Centered xyz only	73.18
PointNet	Non-ID radar features	82.46
PointNet++	Centered xyz only	78.32
PointNet++	Non-ID radar features	86.75
P4Transformer	Centered xyz only	90.45
P4Transformer	Non-ID radar features	92.90
SequentialPointNet	Centered xyz only	89.28
SequentialPointNet	Non-ID radar features	92.71
MSG-Transformer	**Effective radar streams**	**97.67**

TABLE 4. Single-run input-feature ablation using MSG-Transformer on the development split. Scores are percentages.

Input setting	Channels used	Dev. score
Doppler only	D	83.91
SNR only	SNR	71.43
Centered xyz only	$\tilde{x}, \tilde{y}, \tilde{z}$	92.14
Centered xyz + Doppler + SNR	$\tilde{x}, \tilde{y}, \tilde{z}, D$, SNR	95.83
Without centroid, stored-distance control	$\tilde{x}, \tilde{y}, \tilde{z}, D$, SNR, d	96.21
With stored distance control	$\tilde{x}, \tilde{y}, \tilde{z}, D$, SNR, d, pos_x, pos_y	96.88
With tracker ID	Effective streams + tid control	97.05
Proposed	$\boldsymbol{\tilde{x}, \tilde{y}, \tilde{z}, D}$**, SNR,** $\boldsymbol{pos_x, pos_y}$	**97.67**

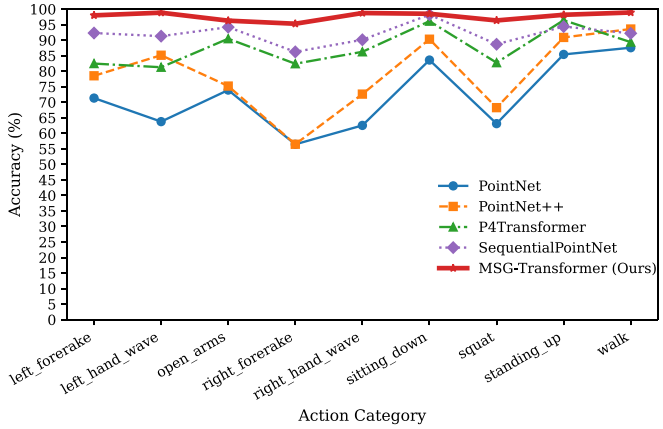

FIGURE 5. Per-class recall of different network models on the development split. Values are from one archived run per setting.

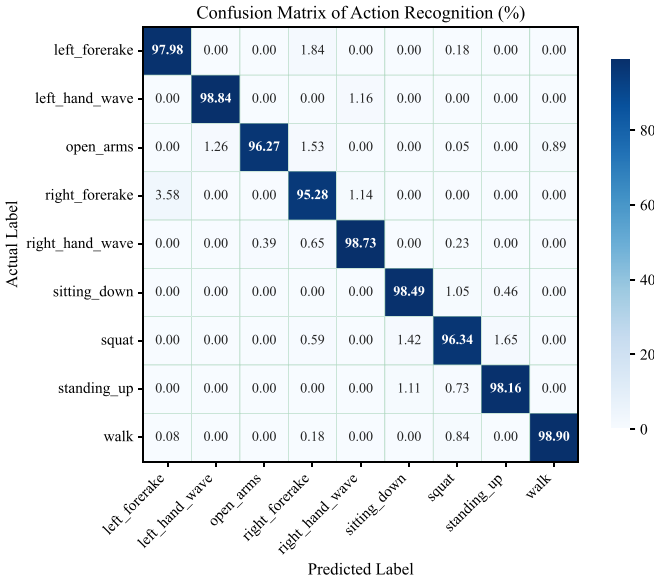

FIGURE 6. Row-normalized development-set confusion matrix, with actual labels on the vertical axis and predicted labels on the horizontal axis. Diagonal entries are per-class recall, and their macro-average is 97.67%.

4.3. Confusion Matrix Analysis

Figure 6 is row-normalized, so each diagonal entry is class recall. The arithmetic mean of the nine diagonal entries is 97.6656%, reported as a 97.67% macro recall rather than weighted accuracy.

The largest displayed off-diagonal value is 3.58% from `right_forerake` to `left_forerake`. Limited azimuth resolution and similar radial-motion patterns are plausible explanations, but the confusion matrix alone cannot identify a causal mechanism. Smaller errors occur among *squat*, *standing up*, and *sitting down*. The component ablation in Table 5 shows an association between the proposed modules and the development score; it does not isolate the physical origin of individual errors.

TABLE 5. Single-run development scores for different coordinate-augmentation strategies.

Strategy	Rotation	Scale	Score	Gain
Baseline	None	None	94.77%	-
Naive Geometric	Full 360°	±20%	93.02%	-1.75%
Restricted geometry	**±15°**	**±5%**	**97.67%**	**+2.90%**

Averaged window probabilities give 82/82 correct `Ver` file predictions, treated only as a small audit because participant identifiers overlap subsets.

4.4. Analysis of Restricted Geometry Augmentation

To avoid mismatching transformed coordinates with measured radar attributes, augmentation perturbs only target-centered coordinates while keeping the original Doppler velocity and SNR values unchanged.

$$\begin{aligned} \mathbf{c}_i^{\text{aug}} &= sR_z(\alpha)\mathbf{c}_i^{\text{ctr}} + \boldsymbol{\epsilon}_i, \\ D_i^{\text{aug}} &= D_i, \quad \text{SNR}_i^{\text{aug}} = \text{SNR}_i, \\ R_z(\alpha) &= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{31}$$

Here, $\mathbf{c}_i^{\text{ctr}} = [\tilde{x}_i, \tilde{y}_i, \tilde{z}_i]^T$, $s \in [0.95, 1.05]$, $\alpha \in [-15°, 15°]$, and $\boldsymbol{\epsilon}_i$ is the Gaussian jitter with standard deviation 0.005 clipped to $[-0.02, 0.02]$. This is regularization, not physical viewpoint simulation.

With full 360° rotation and ±20% scaling, the development score decreased from 94.77% to 93.02% (Table 5). Restricted perturbation reached 97.67% in the archived run. The comparison supports limiting coordinate perturbations when measured Doppler remains unchanged, without treating the augmentation as a physically exact view simulation.

4.5. Effect of the Number of Fused Frames

To evaluate the choice of five-frame stacking, we varied the number of raw frames M fused into each temporal chunk while keeping the remaining model and training settings unchanged. As shown in Figure 7, the classification accuracy increased from 95.38% at $M = 3$ to a maximum of 97.67% at $M = 5$. Further increasing M reduced accuracy to 96.53%, 93.26%, and 89.45% for 8, 10, and 15 frames, respectively. Compared with three-frame stacking, five-frame stacking improved the accuracy by 2.29 percentage points; using 8, 10, and 15 frames produced decreases of 1.14, 4.41, and 8.22 percentage points relative to the five-frame setting.

These results reveal a trade-off between point-cloud density and temporal resolution. Stacking too few frames may provide insufficient radar returns to alleviate the sparsity of individual frames, whereas an excessively long stacking interval may merge distinct motion phases and weaken fine-grained temporal information. At the acquisition rate of 10 fps, five frames correspond to approximately 0.5 s of measurements per chunk. This setting achieved the highest accuracy among the evaluated configurations and was therefore adopted in MSG-Transformer. Because the comparison is based on the present dataset and experimental protocol, $M = 5$ should be regarded as an empirically supported configuration rather than a universally optimal value.

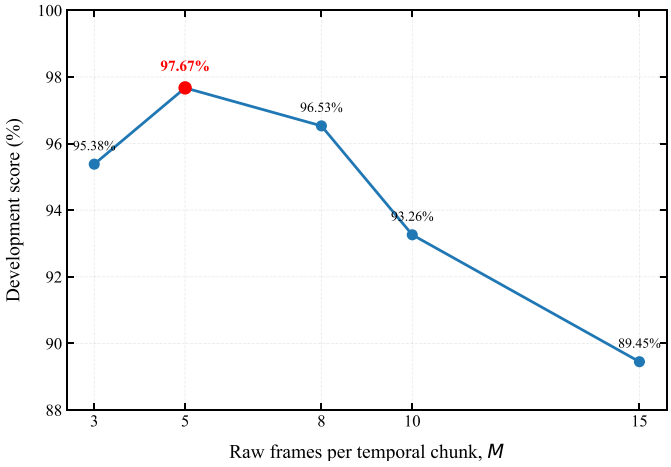

FIGURE 7. Single-run development scores of MSG-Transformer for different numbers of raw frames fused per temporal chunk. Five-frame stacking achieved the highest score among the evaluated settings.

4.6. Ablation Studies

The retained ablations test the multi-stream backbone, temporal module, activation function, and attention mechanism where the archive permits.

4.6.1. Network Architecture Evaluation

Figure 8 reports 92.90%, 88.75%, 92.71%, 91.56%, and 95.36% for MSG-RNN, MSG-GRU, MSG-LSTM, PointNet-Transformer, and PointNet++-Transformer, versus 97.67% for MSG-Transformer. These single-run results support modality-aware encoding with Transformer temporal aggregation.

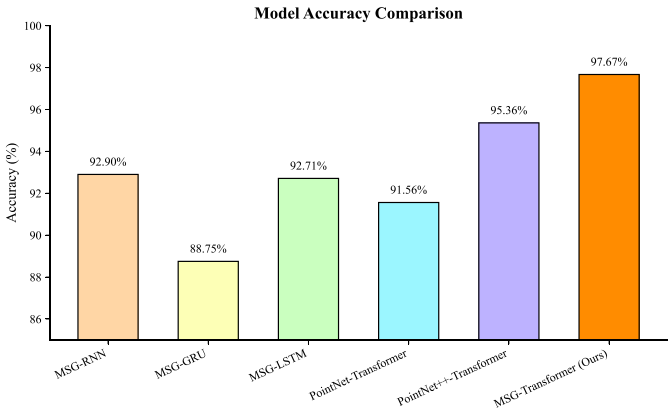

FIGURE 8. Single-run development accuracy across recurrent MSG variants and point-cloud Transformer hybrids.

TABLE 6. Single-run development ablation of activation and attention choices.

Group	Activation	Attention	Development score	Gain
Baseline	ReLU	Dot-Product	89.97%	-
Variant A	ReLU	Cosine	93.60%	+3.63%
Variant B	SwiGLU	Dot-Product	94.19%	+4.22%
Proposed	**SwiGLU**	**Cosine**	**97.67%**	**+7.70%**

TABLE 7. Comparison of computational load and available process-time evidence. N/R indicates that the corresponding training-time record was not retained.

Model	FP32/MiB	GFLOPs	Parameters/M	Train/h	GPU forward/ms
PointNet	13.21	28.83	3.46	N/R	25.30
PointNet++	5.60	55.87	1.47	∼20.00	82.40
P4Transformer	161.51	443.65	42.34	N/R	17.58
SequentialPointNet	11.41	219.23	2.99	N/R	1.32
MSG-Transformer, proposed	**3.92**	**1.48**	**1.03**	**∼2.73**	**43.44**

4.6.2. Component Mechanism Evaluation

Table 6 compares SwiGLU and Scaled Cosine Attention in the archived single-run experiment. Cosine attention removes query-key norm dependence from the attention logits, whereas SwiGLU adds multiplicative nonlinear interactions. Their combination gives the highest development score. Because equal-parameter repeated-seed experiments were not retained, the table provides diagnostic evidence rather than statistical proof.

4.6.3. Comparative Analysis of Computational Complexity

Table 7 compares model size, operation count, parameter count, and process-time records for 10 chunks and 100 points per chunk. The forward-time entries are measured values retained from the corresponding experiments. However, the archived baseline records do not preserve complete hardware, software, warm-up, iteration count, or synchronization metadata; they are therefore presented as descriptive evidence rather than a standardized same-platform latency ranking. Auditing the retained nine-class MSG-Transformer checkpoint gives 1,027,633 trainable parameters, reported as 1.03 M, and a 3.92 MiB FP32 parameter footprint. The serialized PyTorch checkpoint is 4,138,034 bytes because it also contains serialization metadata. The archived operation-count estimate is 1.48 GFLOPs.

We additionally re-benchmarked the retained MSG-Transformer checkpoint with batch size of 1 after 30 warm-up iterations and 200 synchronized timed iterations. On an NVIDIA GeForce GTX 1650 Ti using PyTorch 1.8.1 and Compute Unified Device Architecture (CUDA) 11.1, the neural forward pass required 43.44 ms on average (42.54 ms median; 56.37 ms 95th percentile). On an Intel Core i5-10300H CPU, it required 58.07 ms on average (40.88 ms median; 147.44 ms 95th percentile). These measurements exclude radar acquisition, file I/O, preprocessing, frame stacking, and host-to-device transfer. The baseline GPU forward values in Table 7 are retained mean measurements from the original experiments, but their complete benchmarking metadata are unavailable; consequently, differences between their Giga Floating-point Operation (GFLOP) estimates and recorded latency should not be interpreted as hardware-throughput measurements. The retained PointNet++ record indicates approximately 20.00 h of training. The MSG-Transformer run-directory timestamp and final-checkpoint time imply approximately 2.73 h for its archived 100-epoch run, but the original training hardware and per-epoch timing log were not retained. These two training values are therefore approximate archival records, and the other baseline training times are marked N/R. The comparison is quantitative in model complexity and available process time, but it is not a controlled equal-hardware timing study.

4.7. Experimental Scope and Threats to Validity

The archive bounds the claims. The main class-wise result is a single-split development result because `Test` is used for checkpoint selection, whereas `Ver` is a small file-level audit. The participant distribution does not support strict subject-independent claims, and the retained logs do not include repeated seeds or confidence intervals. Baseline fairness is limited by the absence of exhaustive equal-compute or equal-parameter tuning and by incomplete hardware and benchmark-protocol metadata for the retained baseline timing values. Doppler and stored SNR are measured attributes, but SNR lacks complete unit metadata, and restricted geometry augmentation is not a physical simulation of a new viewpoint.

Cross-subject validation would require every action to be represented for every participant and a participant-disjoint protocol such as leave-one-subject-out evaluation. Cross-environment validation would require training and testing across independently labeled rooms or layouts while control-

ling radar height, orientation, subject distance, azimuth, clutter, and acquisition settings. Neither analysis can be reconstructed from the current archive because those grouping variables and balanced action–participant observations are absent. These experiments are feasible as a new data-collection protocol, but they are not results of the present study.

Simultaneous multi-person recognition is also outside the current formulation. A practical extension would maintain a temporal buffer for each tracked target and apply the classifier independently, or replace the clip-level head with an instance-level multi-label head. Such a system would additionally need to quantify missed tracks, track switches, mutual occlusion, and interference between people. The current temporary tracker-ID field alone does not establish this capability.

The measured forward time indicates that neural computation is compatible with an online pipeline on the benchmarked workstation, but it does not demonstrate low-latency end-to-end operation. At 10 fps, one 50-frame input window spans approximately 5 s, and the archived 25-frame stride produces a new window approximately every 2.5 s. Acquisition, detection, tracking, stacking, and transfer latency remain unmeasured. A deployment study must therefore report both compute latency and observation-window delay.

A stronger validation would reserve an untouched final test set, collect all actions from all participants, report repeated-seed cross-subject and cross-environment results with confidence intervals, benchmark end-to-end latency, test simultaneous multi-person actions, and include matched single-stream controls.

5. CONCLUSION

MSG-Transformer separates the target-centered geometry, Doppler velocity, radar-reported SNR, and horizontal centroid context before temporal modeling of sparse mmWave radar point clouds. Under the archived single-split development protocol, the row-normalized confusion matrix yielded a 97.67% macro recall, and the implemented model has 1.03 M parameters with a 3.92 MiB FP32 parameter footprint. These results demonstrate strong within-protocol performance. The added batch-size-one benchmark measures a 43.44 ms mean neural forward pass on a GTX 1650 Ti, but the 5-s observation window and unmeasured acquisition/preprocessing stages preclude an end-to-end real-time claim. It does not establish subject-independent, cross-environment, or simultaneous multi-person robustness, which remains the main requirement for future studies.

ACKNOWLEDGEMENT

This work is supported in part by the National Natural Science Foundation of China under Grant 62201482.

REFERENCES

[1] Zhang, J., R. Xi, Y. He, Y. Sun, X. Guo, W. Wang, X. Na, Y. Liu, Z. Shi, and T. Gu, "A survey of mmWave-based human sensing: Technology, platforms and applications," *IEEE Communications Surveys & Tutorials*, Vol. 25, No. 4, 2052–2087, 2023.

[2] Yu, C., Z. Xu, K. Yan, Y.-R. Chien, S.-H. Fang, and H.-C. Wu, "Noninvasive human activity recognition using millimeter-wave radar," *IEEE Systems Journal*, Vol. 16, No. 2, 3036–3047, 2022.

[3] Li, J., X. Shao, F. Chen, S. Wan, C. Liu, Z. Wei, and D. W. K. Ng, "Networked integrated sensing and communications for 6G wireless systems," *IEEE Internet of Things Journal*, Vol. 11, No. 17, 29 062–29 075, 2024.

[4] Aldirmaz-Colak, S., M. Namdar, A. Basgumus, S. Özyurt, S. Kulac, N. Calik, M. A. Yazici, A. Serbes, and L. Durak-Ata, "A comprehensive review on ISAC for 6G: Enabling technologies, security, and AI/ML perspectives," *IEEE Access*, Vol. 13, 97 152–97 193, 2025.

[5] Ben-Shabat, Y., O. Shrout, and S. Gould, "3DInAction: Understanding human actions in 3D point clouds," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19 978–19 987, Seattle, WA, USA, 2024.

[6] Yin, W., L.-F. Shi, and Y. Shi, "Indoor human action recognition based on millimeter-wave radar micro-Doppler signature," *Measurement*, Vol. 235, 114939, 2024.

[7] Yu, Z., A. Zahid, W. Taylor, A. Zahid, K. Rajab, H. Heidari, M. A. Imran, and Q. H. Abbasi, "A radar-based human activity recognition using a novel 3-D point cloud classifier," *IEEE Sensors Journal*, Vol. 22, No. 19, 18 218–18 227, 2022.

[8] Feng, R., E. D. Greef, M. Rykunov, H. Sahli, S. Pollin, and A. Bourdoux, "Multipath ghost recognition for indoor MIMO radar," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, 1–10, 2022.

[9] Wei, Y., H. Liu, T. Xie, Q. Ke, and Y. Guo, "Spatial-temporal transformer for 3D point cloud sequences," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 657–666, Waikoloa, HI, USA, 2022.

[10] Zhong, J., H. Jiang, Y. Ji, Y. Li, and Y. Xue, "CSP-Former: A transformer-based network for point cloud analysis with compressed sensing and spatial self-attention," *Electronics*, Vol. 14, No. 2, 347, 2025.

[11] Fan, H., Y. Yang, and M. Kankanhalli, "Point 4D transformer networks for spatio-temporal modeling in point cloud videos," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14 199–14 208, Nashville, TN, USA, 2021.

[12] Zeller, M., V. S. Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss, "Radar velocity transformer: Single-scan moving object segmentation in noisy radar point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 7054–7061, London, United Kingdom, 2023.

[13] Ouaknine, A., A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-view radar semantic segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 15 651–15 660, Montreal, QC, Canada, 2021.

[14] Paek, D.-H., S.-H. Kong, and K. T. Wijaya, "K-radar: 4D radar object detection for autonomous driving in various weather conditions," *Advances in Neural Information Processing Systems*, Vol. 35, 2022.

[15] Sheeny, M., E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "RADIATE: A radar dataset for automotive perception in bad weather," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 1–7, Xi'an, China, 2021.

[16] Chen, Z., S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, No. 2, 1113–1122, 2021.

[17] Shazeer, N., "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[18] Liu, Z., H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin Transformer V2: Scaling up capacity and resolution," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11 999–12 009, New Orleans, LA, USA, 2022.

[19] Yang, J., H. Huang, Y. Zhou, X. Chen, Y. Xu, S. Yuan, H. Zou, C. X. Lu, and L. Xie, "MM-Fi: Multi-modal non-intrusive 4D human dataset for versatile wireless sensing," *Advances in Neural Information Processing Systems*, Vol. 36, 2023.

[20] Lu, D., Q. Xie, M. Wei, K. Gao, L. Xu, and J. Li, "Transformers in 3D point clouds: A survey," *arXiv preprint arXiv:2205.07417*, 2022.

[21] Zhang, H., C. Wang, S. Tian, B. Lu, L. Zhang, X. Ning, and X. Bai, "Deep learning-based 3D point cloud classification: A systematic survey and outlook," *Displays*, Vol. 79, 102456, Sep. 2023.

[22] Qi, C. R., H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–85, Honolulu, HI, USA, 2017.

[23] Qi, C. R., L. Yi, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[24] Wang, Y., Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics (TOG)*, Vol. 38, No. 5, 1–12, 2019.

[25] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[26] Zhao, H., L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16 259–16 268, 2021.

[27] Guo, M.-H., J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, Vol. 7, No. 2, 187–199, 2021.

[28] Wu, X., Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point Transformer V2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, Vol. 35, 2022.

[29] Gao, X., G. Xing, S. Roy, and H. Liu, "RAMP-CNN: A novel neural network for enhanced automotive radar object recognition," *IEEE Sensors Journal*, Vol. 21, No. 4, 5119–5132, Feb. 2021.

[30] Instruments, T., "mmWave SDK and radar toolbox output data formats: Detected-point side-information and tracker output field definitions," https://software-dl.ti.com/ra-processors/esd/MMWAVE-L-SDK/05_04_00_01/exports/api_guide_xwrL64xx/MMWAVE_DEMO.html, 2024.

[31] Bai, S., J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[32] Guo, J., K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "CMT: Convolutional neural networks meet vision transformers," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12 165–12 175, New Orleans, LA, USA, 2022.

[33] Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826, Las Vegas, NV, USA, 2016.