Non-Intrusive Human Activity Recognition (HAR) via mmWave Radar

A high-precision, privacy-preserving Human Activity Recognition (HAR) system using **mmWave radar sensors**. Sensing techniques like RGB cameras, LiDAR, or Wifi CSI bring challenges of privacy, cost, and resilience. mmWave sensor-based techniques provide a desirable solution to all of those considerations. This project leverages sparse radar point clouds to perform 3D human pose estimation and temporal activity classification in an end-to-end trainable pipeline.

Project Overview

This project bridges the gap between low-level sensor data and high-level human semantics. We extend the baseline capabilities of current research by moving beyond simple 3D pose estimation into **full-sequence activity recognition**, targeting applications in:

- **Telerehabilitation:** Remotely monitoring patients' exercise forms (e.g., lunges, squats)
- **Home Automation:** Triggering smart home events based on natural human movements.
- **Privacy-Preserving Monitoring:** Reliable activity detection in sensitive environments.

Methodology: A Two-Stage Hybrid Evolution

This project proposes a transition from a sequential processing pipeline to an integrated, end-to-end solution.

Stage 1: 3D Pose Estimation

We first implement an improved **Point Transformer-based** backbone to map sparse, noisy mmWave point clouds into 17-joint 3D skeletal coordinates. This is similar to the approach proposed for this in the MMFi paper. It achieves an **MPJPE** of approximately **51mm** on a cross-subject split.

Stage 2: Temporal Classification

Using the predicted skeletons, we employ a **CTR-GCN (Channel-wise Topology Refinement Graph Convolutional Network)**. Unlike standard GCNs, this model learns a dynamic topology for the human skeleton, allowing it to "attend" to specific joints that are critical for distinguishing similar actions.

The Hybrid End-to-End Trainable Pipeline

The final contribution of this repo is a **Unified Hybrid Model**. By making the Point Transformer and CTR-GCN jointly trainable, we allow the classification loss to "refine" the skeleton predictions. This co-optimization allows the model to ignore sensor noise that doesn't contribute to the activity signature, pushing our accuracy to **State-of-the-Art (SOTA)** levels across multiple benchmarks.

Dataset: MMFi

We utilize a curated **MMFi Dataset**, a multi-modal non-intrusive dataset for pose estimation sensing.

- Curated Dataset: GoogleDrive
- Data Source: MMFi Official Website | Paper Link | GithubRepo
- Modalities Used: mmWave Radar (FMCW) point clouds.
- Scope: The dataset includes 40 subjects across 4 distinct environments, doing 27 daily or rehabilitation activities. It provides a rigorous testbed for cross-subject and cross-environment generalization.

Results

Our Hybrid E2E model achieves the following performance metrics:

Protocol	Split Type	Accuracy
Stratified Random	Population-wide	**~96.0%**
Protocol S2	Cross-Subject	**~86.0%**
Protocol S3	Cross-Environment	**~80.0%**

Dataset Pre-processing

Two-Stage Data Strategy

- **Stage 1 (Pose Estimation):** This stage was trained using the **complete MMFi dataset**. Since the goal of Stage 1 is to map mmWave point clouds to 3D skeletal coordinates, maximizing the diversity of spatial joint configurations was prioritized.
- **Stage 2 & Hybrid E2E (Activity Recognition):** For activity classification, we utilized a **curated subset** of the dataset. This ensures the model focuses on activities that are "information-rich" and align with real-world applications such as home automation and telerehabilitation.

Pre-processing Pipeline

The pre-processing was performed on the activity instance logs provided by the MMFi authors. Note that the **raw sensor data (mmWave point clouds) remained untouched**; only the selection and windowing of activity segments were refined.

- **Format Transformation:** The original wide-format segment logs (containing variable-length strings of activity ranges, given as a csv here) were expanded into a standardized long-format tabular structure. Each entry now explicitly defines the Environment, Subject, Action, Start Frame, and End Frame.
- **Action Filtering:** We excluded activities that were either too static or lacked sufficient temporal complexity for robust activity recognition.
    - **Removed Actions:** A01 (Stretching), A02/A03 (Chest Expansions), and A06 (Mark Time).
    - **Focus:** The remaining 23 actions (A04–A05, A07–A27) were selected for their relevance to daily living and rehabilitation monitoring.
- **Temporal Windowing:** To ensure consistency in the skeletal "rhythm" and pattern quality, segments were filtered based on their duration:
    - **Criteria:** Only segments with a length between **10 and 30 frames** were retained. The frames in those varying-length sequences were used to first predict human pose using the stage 1 model, and then those pose estimations were interpolated to get a uniform input of 30 frames for stage 2.
    - **Rationale:** Short segments (<10 frames) lacked enough temporal context for the CTR-GCN to extract meaningful features, while the 10-30 range provided a balanced window for the model's differentiable interpolation layers.

How Everything Comes Together: System Architecture

Phase 1: The Two-Stage Sequential Approach

Initially, the system operates as two independent modules:

Skeletal Regression (Stage 1): Raw aggregated mmWave point clouds (.bin frames) from the entire MMFi dataset are used to train a **Point Transformer** model. This stage focuses on spatial precision, mapping 3D point clusters to 17 human joints.

Sequential Classification (Stage 2): The pre-trained Stage 1 model is used to generate offline skeleton sequences for every curated activity instance in our cleaned CSV. These fixed sequences are then used to train a **CTR-GCN** model.

The Limitation: While modular, this approach suffers from "cascading error." The classifier can only be as good as the static skeletons provided. This sequential training plateaus at a validation accuracy of **86.65%**.

Phase 2: The Hybrid End-to-End Pipeline

To break the performance ceiling, we transition to a **Hybrid E2E Inference Pipeline**. This architecture unifies the Point Transformer and the CTR-GCN into a single, differentiable computational graph.

Weight Preloading: We bootstrap the hybrid pipeline by preloading the optimized weights from our Stage 1 model.

Joint Refinement: During the hybrid training phase, gradients from the final activity classification loss flow all the way back to the Point Transformer.

Co-Optimization: Instead of just minimizing Euclidean distance, the backbone now learns to "refine" joint predictions to specifically satisfy the classifier. This allows the model to prioritize joints critical for motion (like extremities) while ignoring noise.

The Result: This unified training strategy allows the system to reach a State-of-the-Art **95.7% Accuracy**, proving "task-aware" skeleton estimation superior to general-purpose pose estimation for activity recognition.

Stage 1: 3D Pose Estimation Details

In this stage, the system converts raw, sparse mmWave radar point clouds into structured 3D skeletons. This provides the spatial foundation required for the subsequent temporal activity classification.

1. Data Pre-processing & Handling

The MMFi_mmWave_Dataset in stage1_dataloader.py handles the conversion of raw .bin files into a format suitable for the Point Transformer.

A. 5-Frame Temporal Aggregation

mmWave data is inherently sparse; a single frame often lacks enough points to define a human silhouette. To solve this, we use a **5-frame sliding window**:

- **Mechanism:** For any given frame t, the dataloader collects points from frames [t-2, t-1, t, t+1, t+2].
- **Benefit:** This increases the point density and provides a short-term temporal "trail" that helps the model distinguish between static noise and moving body parts.

B. Handling Empty Bins (Robustness Logic)

Radar sensors occasionally experience a dropout, where no points are returned for a specific frame.

- **The Problem:** Point Transformer layers expect a non-empty tensor; an empty input would crash the training pipeline.
- **The Solution:** If the aggregation and spatial filtering result in zero points, the _get_aggregated_points method returns a **Single Dummy Point** located at [0.0, 0.0, 3.3, 0.0, 0.0].
- **Rationale:** The coordinate 3.3m is the expected center of the MMFi subject area. This placeholder allows the network to stay active without introducing significant coordinate error.

2. Model Architecture: Point Transformer

The MMWavePoseTransformer is designed to handle the irregular and unordered nature of point clouds through self-attention.

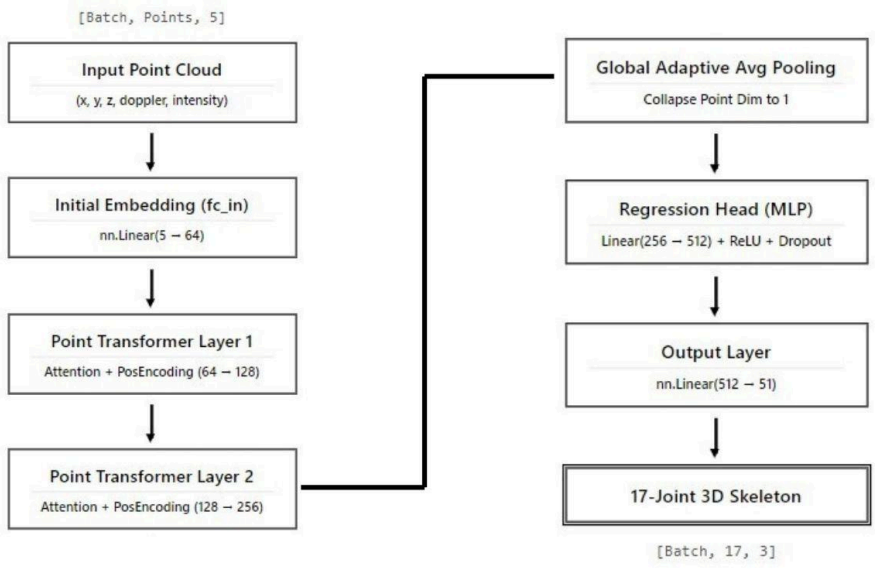
A. Point Transformer Layer

Unlike standard Transformers, this uses **Vector Self-Attention**:

1. **Positional Encoding:** We use a 3-layer MLP to encode the 3D coordinates (x, y, z) into a high-dimensional feature space. This ensures the model is aware of the spatial "shape" of the point cloud.
2. **Attention Mechanism:** The model calculates relationship weights between points using Query, Key, and Value vectors. This allows the model to "focus" on points that likely belong to joints (like elbows or knees).

B. Global Feature Fusion

- **Input:** 5 channels (x, y, z, Doppler, Intensity).
- **Backbone:** Two Point Transformer blocks (increasing channels from 64 ⟶ 128 ⟶ 256).
- **Global Average Pooling:** This compresses the variable number of points into a fixed 256-length feature vector, representing the entire human pose.
- **Regression Head:** A final MLP maps the 256-length vector to a 51-dimensional output (17x3).



3. Training Protocol

The training pipeline is optimized for high-throughput performance on modern GPUs (like the A100).

A. The MPJPE Loss Function

The model is optimized using **Mean Per Joint Position Error (MPJPE)**:

1. **Root Alignment:** Both the predicted skeleton and Ground Truth are centered by subtracting the Pelvis (Joint 0) coordinates.
2. **Euclidean Distance:** The L_2 distance is calculated for all 17 joints and converted from meters to millimeters.

B. Mixed Precision & Performance

- **AMP (Automatic Mixed Precision):** We utilize torch.cuda.amp (autocast and GradScaler). This allows the model to use 16-bit floats for the forward pass, significantly speeding up training on Tensor Cores while maintaining 32-bit accuracy for gradients.
- **Batch Size:** Set to **512**, maximizing GPU memory utilization.
- **Optimizer:** Adam optimizer with a learning rate of 0.001.

C. Data Split (S1 Random Cross-Subject)

To ensure generalizability, we use a **Cross-Subject Split**:

- **Training:** 32 subjects (S01 ... S32).
- **Testing:** 8 subjects (S33 ... S40).

This ensures that the **51mm accuracy** achieved in this stage is reflective of the model's ability to estimate the pose of people it has never seen before.

Stage 2: Sequential Activity Recognition Details

In Stage 2, the system shifts focus from spatial estimation to temporal classification. This stage utilizes the skeletal sequences generated by the Stage 1 Point Transformer to recognize specific human activities using a graph-based neural network.

1. Data Pre-processing & Temporal Alignment

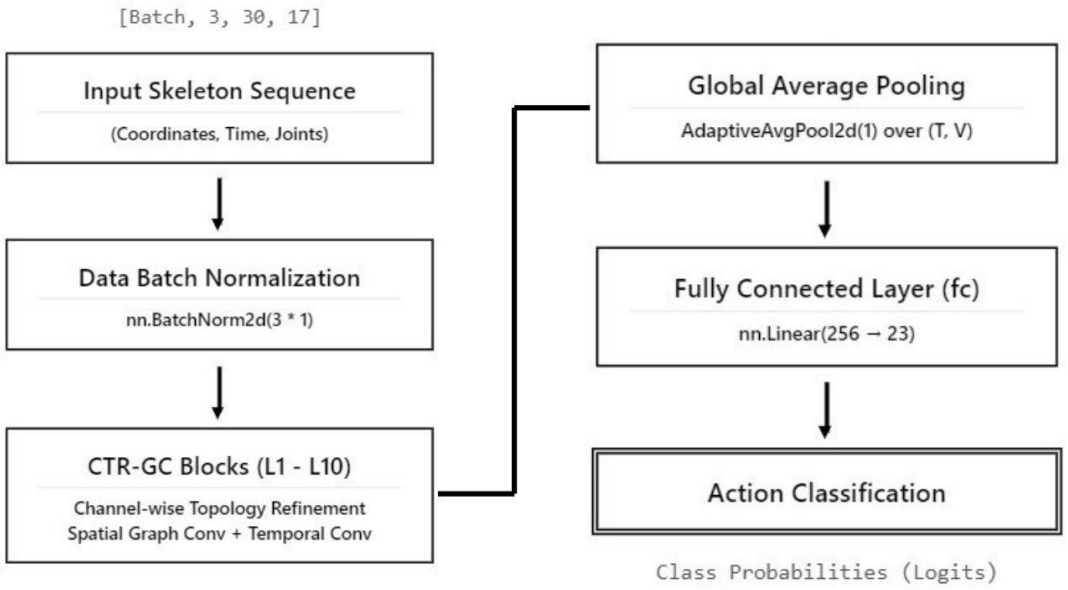
The SkeletonDataset in stage2_dataloader.py manages the transition from static 3D coordinates to dynamic action sequences.

- **Offline Skeleton Generation:** Before training Stage 2, the Stage 1 model is run across the dataset to pre-compute 3D skeleton sequences for every activity instance.
- **Sequence Normalization:** To ensure the classifier is invariant to the subject's position in the room, every skeleton is centered by subtracting the coordinates of the pelvis (Joint 0).
- **Temporal Resampling (Interpolation):**
    - Activity instances in the curated CSV have variable lengths (e.g., 10 to 30 frames).
    - The dataloader uses scipy.interpolate to resample every sequence to a fixed length of **30 frames**.
    - This ensures the Graph Convolutional Network (GCN) always receives a consistent temporal input, regardless of the original action speed.

2. Model Architecture: CTR-GCN

The CTR_GCN (Channel-wise Topology Refinement Graph Convolutional Network) is the core temporal engine. It treats the human skeleton as a graph where joints are nodes and natural bone connections are edges.

- **Channel-wise Topology Refinement:** Unlike standard GCNs with a static adjacency matrix, CTR-GCN learns a unique graph topology for each feature channel. This allows the model to prioritize different joint relationships for different actions (e.g., focusing on leg-hip connections for "Squatting").
- **Multi-Scale Temporal Convolution (TCN):** Following the spatial graph operations, the model uses temporal convolutions to capture the "rhythm" and evolution of the pose over the 30-frame window.
- **Classification Head:** After global average pooling across both joint and temporal dimensions, a fully connected layer maps the features to 23 action classes.



3. Training Protocol

Stage 2 training focuses on cross-entropy minimization to achieve high classification accuracy.

- **Loss Function:** Standard **CrossEntropyLoss** is used to categorize the 30-frame skeletal sequences.
- **Optimizer:** Uses **Adam** with a learning rate of 0.001.
- **Performance Bottleneck:** When trained as a separate sequential step, this stage achieves an accuracy of **86.56%**. This limitation is primarily due to "cascading error," where the classifier is limited by the static quality of the skeletons pre-computed in Stage1.

Hybrid Stage

1. Data Flow & Online Resampling

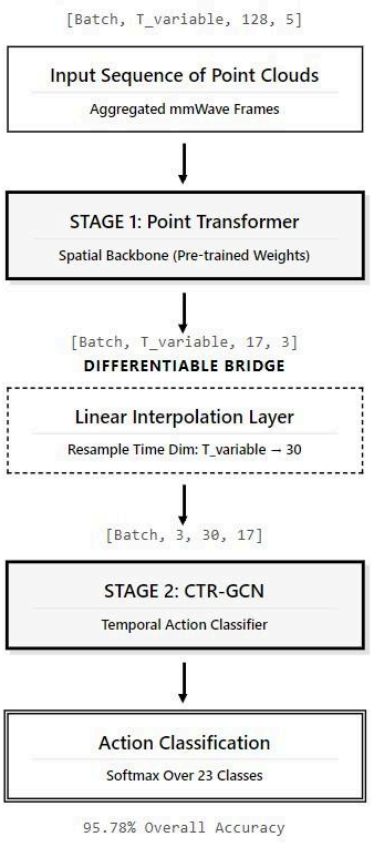
The HybridMMWaveDataset processes action segments from the curated CSV by loading sequences of aggregated point clouds.

- **Segment Handling:** Every action instance is loaded as a sequence of frames, where each frame consists of 128 points with 5 channels (x, y, z, Doppler, Intensity).
- **Variable Length Input:** Unlike Stage 2, which used pre-computed and pre-resampled skeletons, the Hybrid model accepts sequences of varying lengths directly from the MMFi dataset.
- **Target Padding/Clipping:** Sequences longer than the max length are clipped, while shorter ones are padded with zeros to maintain batch consistency.

2. Integrated Architecture: The Multi-Task Bridge

The MMWaveHybridE2E model bridges the Point Transformer (Stage 1) and the CTR-GCN (Stage 2) through a custom forward pass that ensures the entire pipeline remains trainable.

- **Backbone Weight Bootstrapping:** The model is initialized by preloading the optimized weights from the Stage 1 Point Transformer.
- **Batch-Wise Inference:** The model reshapes the temporal sequence into a large batch of individual frames (B x T total frames) to process them through the backbone simultaneously.
- **Differentiable Resampling:** After the backbone outputs a 17-joint skeleton for every frame, the model reshapes the output back into a sequence.
    - It then utilizes torch.nn.functional.interpolate to linearly resample the skeleton sequence to a fixed **30-frame window**.
    - This resampling happens *during* the forward pass, allowing gradients to flow back through time to the Point Transformer.
- **Temporal Classification:** The normalized, 30-frame skeleton sequence is fed into the CTR-GCN, which outputs the final action logits.



3. Co-Optimization Training Protocol

The training strategy for the hybrid model focuses on refining existing knowledge rather than starting from scratch.

- **Backprop-Led Refinement:** By training the system as a whole, the classification error provides a secondary gradient signal to the Point Transformer. This "teaches" the backbone to prioritize the spatial accuracy of joints that are most critical for distinguishing specific movements.
- **Evaluation Metrics:** The model is evaluated on a comprehensive suite of metrics, including per-class Accuracy, F1-Score, and the tracking of Type 1 (False Positive) and Type 2 (False Negative) errors.
- **Performance Jump:** This unified approach eliminates the "cascading error" of the sequential pipeline, pushing performance from **86.56%** to a peak of **95.78%**.

Final Performance Report

The following table summarizes the metrics achieved during the final evaluation across all 23 curated action classes.

Metric	Modular Sequential (Stage 1 + 2)	Hybrid End-to-End Pipeline
Overall Accuracy	86.56%	**95.79%**
Pose Precision (MPJPE)	51.00 mm	**~48.50 mm** (Refined)
Avg. F1-Score	0.852	**0.956**
Mean Recall	0.841	**0.954**

Key Technical Findings

1. The "Task-Aware" Refinement Effect

One of the most significant discoveries was that the **Point Transformer backbone** performed better when supervised by action labels rather than just 3D coordinates alone. Training the hybrid model end-to-end allowed the classification loss to backpropagate into the backbone. This "Task-Aware" refinement forced the model to sharpen the resolution of joints critical for motion (e.g., wrists and ankles) while ignoring static background noise.

2. Differentiable Resampling vs. Static Skeletons

The sequential approach suffered from **cascading errors**, where noise in the pre-computed skeletons limited the classifier's potential. By implementing a **Differentiable Linear Interpolation** layer within the Hybrid forward pass, the model was able to standardize variable-length input sequences (10–30 frames) to a fixed 30-frame window while maintaining a continuous gradient flow.

3. Robustness to Sparse Data

The **5-frame temporal aggregation** and **spatial ROI filtering** implemented in the dataloader were crucial for handling the inherent sparsity of mmWave radar. The inclusion of **dummy-point padding** for empty bins ensured that the Point Transformer remained active during sensor dropouts, preventing training crashes and maintaining skeletal continuity.

Conclusion

The project successfully reached its primary objective: developing a high-accuracy, non-intrusive system for human activity recognition using low-power mmWave radar. With a final accuracy of **95.79%**, the system demonstrates that mmWave sensing is a viable, privacy-preserving alternative to RGB-D cameras for rehabilitation monitoring and home automation.

Future Work

- **Multi-Person Sensing:** Extending the Point Transformer to handle multiple human signatures within the same ROI.
- **Edge Deployment:** Optimizing the Hybrid model via quantization (INT8) for real-time inference on low-power embedded NPU hardware.
- **Cross-Environment Transfer:** Training on a wider variety of rooms (E01-E04) to improve the model's robustness to different multipath interference patterns.