Efficient Gait Recognition with Regularized Gated Recurrent Units

Guarino Angelo[†], Lopez Francesca[†]

***Abstract*—Objective assessment of pathological gait from 3D kinematics is increasingly needed for scalable clinical screening and monitoring. We systematically evaluate deep sequence classifiers and show that input sequence length and leakage-free preprocessing act as strong regularizers, enabling a compact GRU to rival or exceed more complex models. Under Group K-Fold and Leave-One-Subject-Out protocols, the GRU attains 92.42% and 87.33% mean accuracy, respectively, matching state of the art with far fewer parameters. Error analysis indicates confusions stem from secondary compensatory patterns rather than primary clinical markers, which guides feature design and suggests pathways toward deployable, device-agnostic clinical tools.**

***Index Terms*—Pathological Gait Classification, Time-Series Analysis, Deep Learning, Recurrent Neural Networks, Gated Recurrent Unit (GRU), Spiking Neural Networks (SNN).**

[†]Department of Mathematics, University of Padova.

I. Introduction

Objective, automated assessment of pathological gait is increasingly needed to support scalable screening, longitudinal monitoring, and therapy evaluation. Markerless 3D pose estimation and plantar-pressure sensing now make it feasible to capture rich locomotor dynamics in clinics and low-resource settings. Yet, turning such heterogeneous, noisy time series into reliable clinical decisions remains challenging due to inter-subject variability, tracking artifacts, and the risk of overfitting in high-capacity models.

Prior work has largely framed abnormal gait recognition as sequence classification from 3D joint trajectories using recurrent networks, most notably LSTM and GRU variants [1], [2]. Unsupervised encoders have been explored to compress skeleton sequences [3], and multimodal fusion with plantar pressure has shown complementary benefits [4]. However, three issues are recurrent: (*i*) preprocessing and normalization are often fit with leakage across subjects; (*ii*) masking/padding and window-length choices are treated as heuristics rather than regularizers; and (*iii*) increased architectural complexity is used to compensate for dataset idiosyncrasies, with limited gains in generalization. We revisit these design choices on the public Azure Kinect pathological gait dataset with paired pressure maps [5].

This paper. We show that a compact, well-regularized GRU classifier, when paired with a *leakage-free, clinically grounded* preprocessing pipeline, is sufficient to achieve strong, subject-robust performance under Group K-Fold and LOSO validation. Our framework emphasizes invariances (pelvis-centering, stature scaling), train-only standardization, padding-aware masking, and a principled selection of the input length, yielding a simple model that outperforms or matches deeper stacks and attention hybrids while being far more efficient. Optional unsupervised feature extractors (PCA and an RNN autoencoder) are integrated without violating subject isolation, enabling capacity control when needed [3], [6].

In summary, we:

- **Establish sequence length as a first-class regularizer.** We treat the input horizon T as a model-capacity knob and motivate a compact $T{=}100$ policy that curbs overfitting while preserving salient gait cycles (consistent with [2]); see Sec. VI.
- **Design a leakage-free, clinically informed preprocessing pipeline.** Building on best practices [6], we combine pelvis-centering, stature scaling, train-only z-scoring, padding-aware masking, and optional angles/velocities to encode task-relevant invariances without contaminating validation/test statistics (Sec. IV).
- **Advance a compact GRU classifier.** A single GRU layer with a small dense head (about 10^5 parameters) provides an good accuracy-efficiency trade-off and consistently generalizes to unseen subjects, outperforming deeper/attention baselines in our setting (Sec. V).
- **Integrate unsupervised encoders without leakage.** We evaluate PCA and a masked seq2seq RNN autoencoder as optional compressors trained only on training subjects, clarifying when linear vs. non-linear temporal embeddings help (Sec. IV).
- **Provide subject-aware evaluation and diagnostics.** We adhere to Group K-Fold and LOSO protocols and include error/temporal analyses that explain class confusions and guide future feature design (Sec. VI).

Paper structure: Sec. II reviews related work. Sec. III overviews the end-to-end pipeline. Signals, preprocessing, and features are detailed in Sec. IV. The learning framework and the final GRU formulation appear in Sec. V. Sec. VI reports results and diagnostics, and Sec. VII concludes.

II. Related Work

Pathological gait recognition from 3D skeleton trajectories has been extensively studied using Kinect-based datasets, often complemented by plantar-pressure sensing. We briefly review the most relevant trends and position our contribution with respect to them.

a) RNN-based skeleton classifiers: Early supervised approaches framed abnormal gait recognition as sequence classification with recurrent neural networks (RNNs). Lee *et*

al. [1] employed LSTM-based models on 3D joint time series, demonstrating that recurrent memory is effective in capturing periodic locomotor dynamics. Jun *et al.* [2] showed that GRUs can match or surpass LSTMs with fewer parameters, explicitly tailoring the architecture to gait patterns acquired via Kinect v2. These works established the efficacy of gated recurrent units for pathological gait, but typically relied on fixed heuristics for sequence length and feature scaling that were not systematically stress-tested as regularizers.

b) Unsupervised representation learning: Beyond raw joint positions, unsupervised encoders have been used to learn compact, discriminative embeddings. Jun *et al.* [3] proposed an RNN autoencoder that reconstructs skeleton sequences and yields latent features that improve downstream classification. Such sequence-to-sequence bottlenecks reduce redundancy and may denoise jitter in skeletal tracking. However, most prior studies did not combine these embeddings with clinically meaningful kinematic descriptors (e.g., joint angles) nor assess their impact under strict subject-wise evaluation with leakage-free preprocessing.

c) Multimodal fusion with plantar pressure: Pressure maps provide complementary contact mechanics that are difficult to infer from kinematics alone. Jun *et al.* [4] proposed a deep multimodal framework fusing a skeleton stream with a CNN-based pressure stream, reporting tangible gains over unimodal baselines. Recent data efforts have further highlighted robust normalization practices for pressure intensities and variability across conditions [7]. While fusion consistently helps, design choices (early vs. late fusion, embedding dimensionality, and regularization) are often underreported, and masking/padding strategies for variable-length sequences are seldom discussed.

d) Normalization and invariances: Preprocessing is critical for cross-subject generalization. Vox and Wallhoff [6] analyzed centering, orientation, and scaling strategies for 3D skeletons, showing that removing global translation/size improves recognition. Nevertheless, several pipelines risk subtle target leakage by fitting normalization statistics per trial or across all subjects. Likewise, the impact of per-trial windowing vs. fixed-length cropping has rarely been quantified in terms of overfitting control.

e) Datasets: Our study builds on the public Azure Kinect skeleton and foot-pressure dataset [5], a standard benchmark in the above lines of work [1]–[4]. Its paired modalities and labeled pathological classes enable controlled comparisons between unimodal and multimodal learners.

In light of these findings, our framework contributes along four points:

1) *Regularized sequence design.* We treat input length as a first-class regularizer, empirically motivating a compact $T{=}100$ policy that reduces overfitting while preserving salient gait cycles (cf. GRU efficacy in [2]).

2) *Leakage-free normalization with clinical priors.* Building on [6], we combine pelvis-centering, stature scaling, and train-only z-scoring with *pelvis-relative velocities* and *joint-angle* features, balancing interpretability and expressiveness.

3) *Unsupervised + supervised synergy.* In the spirit of [3], we study PCA and RNN autoencoders as optional *label-free* compressors before lightweight GRU heads, quantifying trade-offs under subject-aware validation.

4) *Transparent multimodal late fusion.* Inspired by [4] and robust pressure practices [7], we implement a simple GRU + CNN late-fusion baseline, explicitly documenting masking, padding, and normalization choices for reproducibility.

Taken together, our design emphasizes compact recurrent models with principled preprocessing and evaluation. This yields competitive accuracy with fewer parameters and clearer ablation evidence than prior art, while maintaining coherence with established practices in the field [1]–[4], [6].

III. Processing Pipeline

This section introduces the proposed end-to-end pipeline for Human Gait Analysis (HGA) based on synchronized 3D skeleton trajectories and foot-pressure maps. We outline the main processing blocks, their interactions, and the overall logic of the system. An overview block diagram is reported in Fig. 1; a compact training/validation flow is shown in Fig. 2.

A. High-Level Overview

At a glance, the pipeline ingests raw trial data and progressively transforms it into model-ready tensors while enforcing subject-aware evaluation. The key stages are:

1) **Data Ingestion and Pre-Processing** — raw skeleton trajectories and foot-pressure maps are loaded per trial and subject. A preprocessing block ensures consistent temporal alignment and basic normalization of both modalities (see Sec. IV for details).

2) **Feature Fusion** — preparation of modeling inputs: either skeleton-only (96 features), skeleton enriched with engineered angles and velocities (200 features), or multimodal fusion where a CNN-based pressure stream is concatenated with the skeleton branch.

3) **Representation Learning** — unsupervised dimensionality reduction via PCA (to preserve maximum variance in fewer dimensions) or sequence-to-sequence autoencoders (to learn compact, time-dependent embeddings).

4) **Supervised Modelling** — multiple architectures were implemented to capture different inductive biases in sequential data:

- **GRU/LSTM Baselines** — recurrent models able to capture temporal dependencies and widely used in sequence classification.
- **Regularized GRU/LSTM Variants** — models with dropout and L2 regularization to mitigate overfitting and improve generalization.
- **Attention-augmented RNNs** — hybrid architectures combining recurrent layers with multi-head self-attention, improving long-range dependency modeling and interpretability.

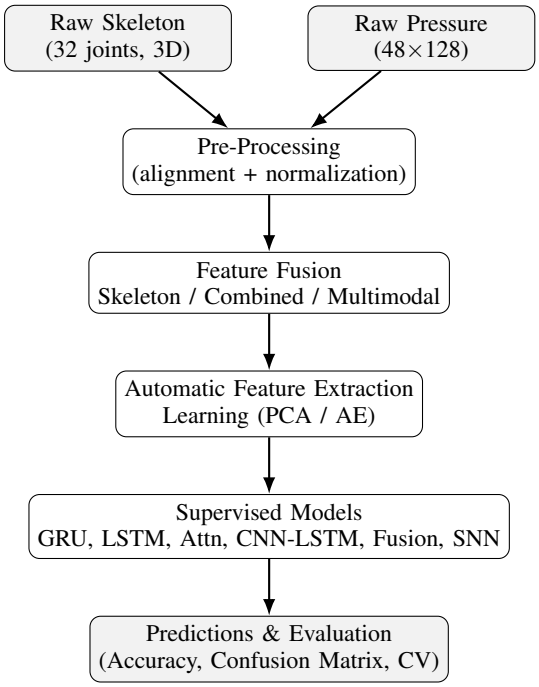

Fig. 1: System-level processing pipeline. Raw skeleton and pressure data are pre-processed, fused, optionally compressed with representation learning, and then classified with diverse neural network architectures.

- **Deep Stacked GRU** — a four-layer GRU inspired by prior work in pathological gait classification, designed to increase model capacity and capture hierarchical temporal patterns.
- **Deep GRU with Attention** — the stacked GRU enriched with a self-attention mechanism, balancing depth and selective focus on salient gait dynamics.
- **CNN-LSTM Hybrid** — convolutional layers extract local spatio-temporal features which are then processed by recurrent layers, enabling hierarchical representation learning.
- **Multimodal GRU+CNN Fusion** — parallel GRU for skeleton data and CNN for pressure maps, fused at a late stage to exploit complementary modalities.
- **Spiking Neural Network (SNN)** — a recurrent spiking architecture analogous to the deep GRU stack, explored as a biologically inspired alternative with potential energy efficiency advantages.

5) **Evaluation Protocols** — strict subject-aware evaluation with Group K-Fold and Leave-One-Subject-Out (LOSO). This ensures generalization across unseen subjects. Metrics include accuracy, per-class precision/recall/F1, and confusion matrices (see Sec. VI).

B. Pipeline Logic

The pipeline is designed to progressively transform raw multi-modal gait data into discriminative representations for robust classification:

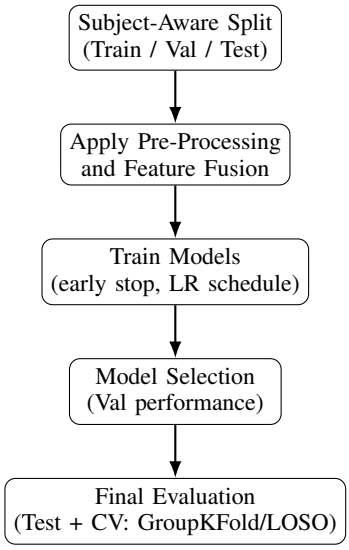

Fig. 2: Training/validation flow with strict subject isolation and leakage-free statistics.

- **Pre-processing** ensures data consistency and comparability across subjects while avoiding leakage.
- **Feature Fusion** enables both unimodal and multimodal setups, allowing ablation studies on the contribution of skeleton, engineered features, and pressure data.
- **Representation Learning** provides dimensionality reduction and denoising, useful when handling high-dimensional temporal features.
- **Supervised Modelling** covers a spectrum of architectures, from simple recurrent baselines to biologically inspired spiking models, enabling a comprehensive performance comparison.
- **Evaluation Protocols** are subject-aware, ensuring that the reported performance of the system reflects real-world generalization to unseen individuals.

The pipeline is *modular* (blocks can be ablated or extended), *flexible* (supporting unimodal, multimodal, and reduced representations), and *evaluation-safe* (subject-isolated statistics and cross-validation). Detailed algorithms and results are presented in Sec. V and Sec. VI.

IV. Signals and Features

A. Data setup and raw signals

We study human gait using two complementary sensing modalities collected per trial and per subject [5].

a) **3D skeleton trajectories (Azure Kinect).** Each frame provides 3D coordinates for 32 anatomical joints (pelvis-rooted kinematic tree), stacked as (x, y, z) triplets, for a total of 96 features per timestamp. We adopt the canonical axes: x lateral, y forward progression, z vertical. A trial is thus a variable-length time series $\mathbf{s}t \in \mathbb{R}^{96} t = 1^T$ indexed by timestamps.

b) **Plantar pressure maps (GW1100 plate).** For each trial we have a 48×128 average pressure image from a 6,144-sensor plate (maximum $100\,\mathrm{N/cm^2}$), denoted by $\mathbf{P} \in$

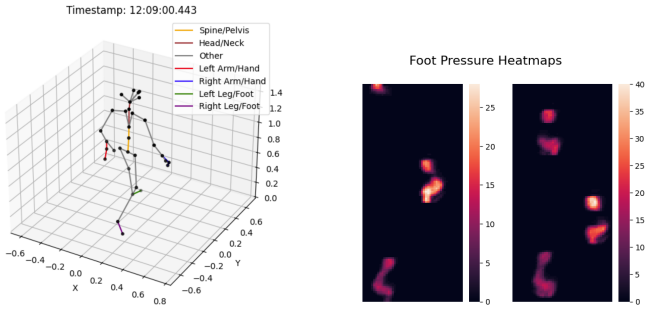

Fig. 3: Left: example 3D skeleton frame with segment connectivity. Right: sample plantar-pressure heatmaps from different trials.

$\mathbb{R}^{48\times128}$. See also [7] for related high-resolution pressure datasets.

c) Metadata and labels. A manifest links tuples `(subject, gait_type, trial)` to the paired skeleton/pressure files. Labels are one-hot encoded over six classes: *antalgic*, *lurching*, *normal*, *steppage*, *stiff-legged*, *trendelenburg* [2], [4].

B. Pre-processing

The objective is to enforce temporal consistency, remove nuisance factors (global translation/scale and speed), and standardize features *without* information leakage across subject-wise splits, following best practices for skeleton normalization [6].

1) **Timestamp sanitization (skeleton).** Raw indices occasionally contain a stray minus sign in the *seconds* field. We rebuild a valid timestamp by splitting date/time, stripping any leading from seconds, reassembling the string in the standard format, and parsing to `datetime`. We then drop duplicate indices and sort the series. This yields a clean, monotone time base per trial.

2) **Geometric normalization (per frame & per trial).** Let $\mathbf{X}_t \in \mathbb{R}^{32\times3}$ be joint coordinates and $\mathbf{c}_t \in \mathbb{R}^3$ the pelvis (joint 0). We first remove global translation by pelvis centering,

$$\tilde{\mathbf{X}}_t = \mathbf{X}_t - \mathbf{c}_t\mathbf{1}^\top,$$

which makes features invariant to absolute position. Then we remove body-size effects using a *stature proxy* $s > 0$ computed as the median over time of selected segment lengths (thighs, shanks, trunk), and scale

$$\hat{\mathbf{X}}_t = \tilde{\mathbf{X}}_t / s.$$

This realizes subject normalization without requiring anthropometric metadata [6].

3) **Global standardization (train-only).** To prevent leakage, we flatten $\hat{\mathbf{X}}_t \mapsto \hat{\mathbf{x}}_t \in \mathbb{R}^{96}$ and fit a z-score (μ, σ) on *training* subjects only; we then apply

$$\mathbf{z}_t = (\hat{\mathbf{x}}_t - \mu) \oslash \sigma$$

to validation/test. Standardization stabilizes optimization and is required by scale-sensitive methods like PCA.

4) **Clinically inspired joint angles (optional).** We derive eight angles (knees, hips, ankles, shoulders) from centered+scaled coordinates using the three-point formula

$$\theta = \arccos\Big(\frac{(\mathbf{a}-\mathbf{b})^\top(\mathbf{c}-\mathbf{b})}{\|\mathbf{a}-\mathbf{b}\|\,\|\mathbf{c}-\mathbf{b}\|}\Big),$$

$u = 2(\theta/180^\circ) - 1 \in [-1, 1]$, computed *pre* z-score to preserve physical magnitudes. Angles provide interpretable kinematics complementary to positions [1], [2].

5) **Pelvis-relative velocities (optional).** From centered+scaled positions we compute per-joint finite-difference velocities

$$\mathbf{V}_t = \frac{\hat{\mathbf{X}}_t - \hat{\mathbf{X}}_{t-1}}{\Delta t_t},$$

express them *relative to pelvis* to remove whole-body drift, normalize by the median forward pelvis speed to reduce pace differences, and finally apply a train-only z-score. Velocities encode dynamics that positions/angles alone may miss.

6) **Pressure normalization (robust).** For each map $\mathbf{P}$ we apply percentile-based scaling using the 95th percentile $q_{0.95}$ to damp saturated values while preserving morphology:

$$\mathbf{P}' = \frac{\mathbf{P} - \min(\mathbf{P})}{\max\{q_{0.95}(\mathbf{P}) - \min(\mathbf{P}),\, \epsilon\}},$$

with a small ϵ safeguard; cf. robust practices in plantar-pressure analysis [7].

7) **Windowing and unification to fixed length.** Trial lengths concentrate around the median ($\sim$194) with IQR 164–224 and a right tail up to 433. We initially set T to the 95th percentile so that most trials are untruncated. *However*, we subsequently fixed T = `MAX_TIMESTEPS` = 100, keeping the *last* T frames (`truncating=pre`) and padding at the end with 0.0 (`padding=post`) masked during training. This choice regularizes the models, reduced overfitting, and improved both running time and accuracy.

C. Feature representations

We consider unimodal and multimodal feature sets:

- **Skeleton-only** (*96D*): standardized positions $\mathbf{z}_t \in \mathbb{R}^{96}$.
- **Engineered fusion** (*200D*): concatenate positions (96), angles (8), and z-scored velocities (96): $\mathbf{f}_t \in \mathbb{R}^{200}$.
- **Pressure map features**: normalized $\mathbf{P}' \in \mathbb{R}^{48\times128}$ processed by a convolutional extractor to obtain a compact spatial embedding, later fused with the skeleton stream (late fusion) [4].

D. Automatic feature extraction (unsupervised)

To reduce dimensionality and noise while retaining discriminative structure, we explore both linear and non-linear sequence representations learned *without labels*. This step provides compact inputs for downstream sequence classifiers while controlling model capacity.

a) Principal Component Analysis (PCA).: We fit PCA on *training* frames only, after removing padded rows: stack all non-zero $\mathbf{z}_t \in \mathbb{R}^{96}$ into a matrix $\mathbf{Z} \in \mathbb{R}^{N\times 96}$ and learn orthonormal directions maximizing variance. Given a target variance threshold $\tau \in \{0.90, 0.95, 0.99\}$, we select the smallest K such that the cumulative explained variance exceeds τ, and project each frame as

$$\tilde{\mathbf{z}}_t = \mathbf{W}_{(K)}^{\top}(\mathbf{z}_t - \boldsymbol{\mu}_{\text{pca}}), \qquad K = \min\{k : \ \text{cumvar}(k) \geq \tau\}.$$

PCA offers a strong linear baseline, mitigates multicollinearity via orthogonal components, and can denoise high-frequency jitter by truncating low-variance directions.

b) Sequence-to-sequence autoencoder (AE).: Inspired by [3], we train a masked seq2seq model that maps a length-T input sequence to a length-T latent sequence and reconstructs the inputs:

$$\{\mathbf{f}_t\}_{t=1}^{T} \xrightarrow{\text{encoder}} \{\mathbf{h}_t\}_{t=1}^{T}, \quad \mathbf{h}_t \in \mathbb{R}^d,$$
$$\{\hat{\mathbf{f}}_t\}_{t=1}^{T} \xleftarrow{\text{decoder}} \{\mathbf{h}_t\}_{t=1}^{T}.$$

The encoder stacks four LSTM layers and a time-distributed dense bottleneck to produce $\mathbf{h}_t$; the decoder is an LSTM with time-distributed linear output. We optimize a *masked MSE* that ignores padded frames:

$$\mathcal{L} = \frac{\sum_{t=1}^{T} \|(\hat{\mathbf{f}}_t - \mathbf{f}_t) \odot \mathbf{m}_t\|_2^2}{\sum_{t=1}^{T} \|\mathbf{m}_t\|_1 + \varepsilon},$$

where $\mathbf{m}_t \in \{0,1\}^F$ is a binary mask (1 on valid, 0 on padded) and $\varepsilon > 0$ prevents division by zero. We explore $d \in \{20, 40, 60\}$ and select using validation reconstruction loss and downstream classification accuracy. Compared to PCA, AEs can capture non-linear manifolds and temporal regularities through the recurrent encoder, yielding more expressive yet compact embeddings [3].

c) Usage.: Both PCA and AE are trained *exclusively* on training subjects (fit parameters and weights) and then applied unchanged to validation/test. They produce per-frame embeddings of size K or d, yielding tensors of shape (trials, T, $\tilde{F}$) that are plug-and-play for the sequence models considered in this work.

E. Dataset partitions

We adopt **subject-wise** partitions to prevent identity leakage and to evaluate generalization to *unseen* individuals:

- **Train**: $\approx 70\%$ of subjects (used to fit normalization statistics, PCA/AE, and model parameters).
- **Validation**: $\approx 15\%$ of subjects (model selection and early stopping).
- **Test**: $\approx 15\%$ of subjects (held out for final reporting).

All trials of a subject belong to exactly one split. The following elements are fit on *train-only* and then frozen: z-score (μ, σ) for positions and velocities, velocity speed-normalization factor, PCA mean/components, and AE weights. For robustness, we also employ **Group K-Fold** (groups = subjects) and **LOSO** cross-validation; at each fold, all preprocessing and unsupervised models are re-fitted only in the training subjects of the fold [2], [4].

TABLE 1: Skeleton Dataset overview (after preprocessing and sequence unification).

Split	Subjects	Trials	Sequence length	Features
Train	8	960	100	96
Val	2	240	100	96
Test	2	240	100	96

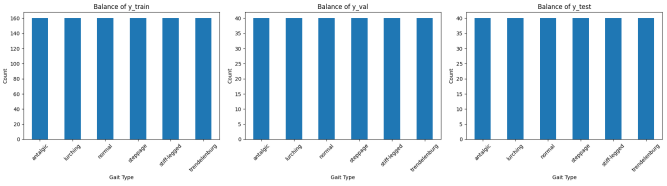

Fig. 4: Class balance across train/validation/test splits (number of trials per gait type).

Tab. 1 summarizes the skeleton stream after sequence unification; pressure maps are available for the same trials and are normalized as in Sec. IV-B. Unsupervised feature extractors (PCA/AE) and all scalers are always fit on training subjects only, then applied unchanged to validation/test.

V. Learning Framework

This section details the learning strategy adopted to classify gait sequences. We first outline the family of architectures explored, then formalize our final GRU-based model with a layer-by-layer mathematical description. A schematic diagram is reported in Fig. 5. Optimization choices conclude the section.

A. Architectures and Models

We implemented a spectrum of sequence models to probe different inductive biases, always fed with fixed-length sequences of size T=100 and feature dimension F (cf. Sec. IV):

- **Baseline LSTM**: `Masking` → `LSTM(128)` → `Dropout(0.5)` → `Dense(64, ReLU)` → `Dropout(0.5)` → `Softmax`. Serves as a standard recurrent baseline.
- **Regularized LSTM**: as above with stronger L_2 and recurrent dropout; tests whether heavier regularization alone controls overfitting.
- **Attention-augmented RNN**: `LSTM(128, return_seq.)` → `MultiHeadAttention` → `GAP` → `Dense(64)` → `Softmax`. Targets longer-range temporal cues with data-driven focus.
- **Deep Stacked GRU**: 4× `GRU(125)` (last returns vector) → `Dense(125, ReLU)` → `Dropout(0.5)` → `Softmax`; inspired by prior gait work [2].
- **Deep GRU + Attention**: 4× `GRU(125, return_seq.)` → `MultiHeadAttention` →

`GAP` → `Dense(125)` → `Softmax`. Hybrid capacity + focus.

- **CNN-LSTM**: `Conv1D` → `Conv1D` → `MaxPool` → `Masking` → `LSTM(100, return_seq.)` → `LSTM(50)` → `Dense(64)` → `Softmax`. Learns local spatio-temporal patterns before recurrent aggregation.
- **GRU+CNN fusion (multimodal)**: GRU branch for skeleton + CNN branch for pressure, late concatenation, then `Dense(96)` → `Softmax` [4].
- **Deep Recurrent SNN**: 4-layer recurrent spiking network (PyTorch/snnTorch), explored for efficiency; trained with surrogate gradients.

B. Final architecture: GRU sequence classifier

Our best model is a compact *sequence-to-vector* GRU classifier:

$$\begin{aligned}\mathbf{X}\in\mathbb{R}^{T\times F} &\xrightarrow{\text{Masking}} \mathbf{X}_{\text{eff}}\\ &\xrightarrow{\text{GRU}(H=128)} \mathbf{h}_T\in\mathbb{R}^H\\ &\xrightarrow{\text{Dense(64)+ReLU+Dropout(0.5)}} \mathbf{a}\in\mathbb{R}^{64}\\ &\xrightarrow{\text{Softmax}} \hat{\mathbf{y}}\in\Delta^{C-1}.\end{aligned} \tag{1}$$

where $C{=}6$ is the number of classes. Each block is detailed below.

a) Masking (padding-aware).: Let $\{\mathbf{x}_t\}_{t=1}^{T}$, $\mathbf{x}_t\in\mathbb{R}^F$, and a binary mask $\{m_t\}_{t=1}^{T}$ with $m_t{=}1$ on valid frames and 0 on padded frames (value 0.0 in the tensors). The recurrent computation is modified so that masked steps do not affect the state:

$$\tilde{\mathbf{h}}_t=\mathrm{GRUCell}(\mathbf{x}_t,\mathbf{h}_{t-1}),\qquad \mathbf{h}_t=m_t\,\tilde{\mathbf{h}}_t+(1{-}m_t)\,\mathbf{h}_{t-1}\,,$$

propagating the last valid state over padding. This preserves both gradient flow and statistical independence of padding.

b) GRU cell.: With hidden size H and using $\sigma(\cdot)$ and $\tanh(\cdot)$ as the logistic and hyperbolic tangent, respectively, the update equations at time t are

$$\begin{aligned}\mathbf{z}_t&=\sigma(\mathbf{W}_z\mathbf{x}_t+\mathbf{U}_z\mathbf{h}_{t-1}+\mathbf{b}_z),\\ \mathbf{r}_t&=\sigma(\mathbf{W}_r\mathbf{x}_t+\mathbf{U}_r\mathbf{h}_{t-1}+\mathbf{b}_r),\\ \tilde{\mathbf{h}}_t&=\tanh(\mathbf{W}_h\mathbf{x}_t+\mathbf{U}_h(\mathbf{r}_t\odot\mathbf{h}_{t-1})+\mathbf{b}_h),\\ \mathbf{h}_t&=\mathbf{z}_t\odot\mathbf{h}_{t-1}+(\mathbf{1}-\mathbf{z}_t)\odot\tilde{\mathbf{h}}_t\,,\end{aligned} \tag{2}$$

with $\odot$ denoting element-wise multiplication. The reset gate $\mathbf{r}_t$ controls the influence of the past on the candidate state, while the update gate $\mathbf{z}_t$ trades off carry-over vs. overwrite of $\mathbf{h}_{t-1}$. This *gating* mitigates vanishing gradients and captures multi-scale temporal dependencies in gait dynamics.

c) Dense + ReLU + Dropout.: The final recurrent summary $\mathbf{h}_T$ is mapped to a compact hidden representation $\mathbf{a}$:

$$\mathbf{u}=\mathbf{W}_1\mathbf{h}_T+\mathbf{b}_1,\qquad \mathbf{a}=\mathrm{ReLU}(\mathbf{u}),\qquad \tilde{\mathbf{a}}=\mathbf{d}\odot\mathbf{a},$$

where $\mathbf{d}\sim\mathrm{Bernoulli}(1{-}p)$ is the dropout mask applied only at training (here $p{=}0.5$). Dropout regularizes co-adaptations in the penultimate layer.

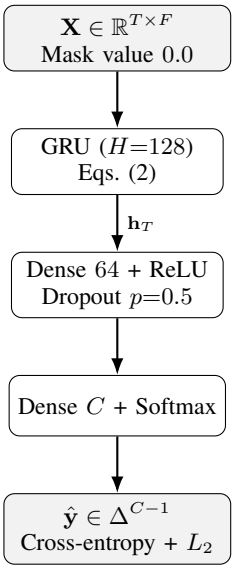

Fig. 5: Final sequence-to-vector GRU classifier. Masked padding prevents spurious gradients; gating captures multi-scale kinematics; a compact dense head regularizes decision boundaries.

d) Softmax classifier and loss.: The probabilistic prediction $\hat{\mathbf{y}}\in\mathbb{R}^C$ is

$$\begin{aligned}\hat{\mathbf{y}}&=\mathrm{softmax}(\mathbf{W}_2\,\tilde{\mathbf{a}}+\mathbf{b}_2)\,,\\ \hat{y}_k&=\frac{\exp(v_k)}{\sum_{j=1}^{C}\exp(v_j)},\\ \text{where}\quad \mathbf{v}&=\mathbf{W}_2\,\tilde{\mathbf{a}}+\mathbf{b}_2,\quad k=1,\dots,C.\end{aligned} \tag{3}$$

trained with the cross-entropy for one-hot labels $\mathbf{y}$:

$$\mathcal{L}_{\text{CE}}=-\sum_{k=1}^{C}y_k\log\hat{y}_k\,.$$

We add L_2 (weight decay) on selected layers to penalize large weights:

$$\mathcal{L}=\mathcal{L}_{\text{CE}}+\lambda(\|\mathbf{W}_z\|_F^2+\|\mathbf{U}_z\|_F^2+\cdots+\|\mathbf{W}_1\|_F^2+\|\mathbf{W}_2\|_F^2),$$

with $\lambda{=}0.01$ matching the implementation.

e) Capacity and shapes.: With $F{=}96$ (skeleton-only), $H{=}128$ and $C{=}6$, the GRU parameter count is

$$\#\theta_{\text{GRU}}=3\,(F{+}H{+}1)\,H=3\,(96{+}128{+}1)\cdot 128=86{,}400\,,$$

while the dense layers contribute $128{\times}64{+}64=8{,}256$ and $64{\times}6{+}6=390$ parameters, for a total of $\approx 95\,\text{k}$. This size is appropriate for $T{=}100$ and helps generalization.

f) Why this model fits HGA: (i) *Gating* (Eq. (2)) selectively carries/forgets information, matching the periodic-yet-variable nature of gait; (ii) *sequence-to-vector* summarization $\mathbf{h}_T$ aligns with trial-level labels and avoids over-parameterized decoders; (iii) *masking* is consistent with fixed-length batching without leaking padding; (iv) *moderate capacity* (∼95k params) balances expressivity and overfitting control at $T{=}100$; (v) L_2 and dropout stabilize training and

sharpen decision margins.

C. Optimization and training protocol

All models are trained with Adam (learning rate 10^{-4}), categorical cross-entropy, and accuracy as the metric. We employ early stopping (patience as in the configuration), and a learning-rate scheduler (`ReduceLROnPlateau`) on validation loss. Mini-batches are built via `tf.data` with shuffling, caching, and prefetching for throughput. Subject-aware validation is enforced throughout (cf. Secs. III–IV); deeper performance analyses are deferred to Sec. VI.

VI. Results

This section details our empirical findings, beginning with considerations on the pre-processing pipeline and an ablation study on sequence length as a key regularizer, followed by the selection and analysis of our best-performing model, and concluding with a robust cross-validation assessment.

A. Pre-processing Pipeline as the Enabler

We contend that our near-state-of-the-art results are driven primarily by a comprehensive, *leakage-free* pre-processing pipeline that removes dataset biases rather than by architectural complexity. In contrast to prior SOTA systems built on deep stacked or attention-augmented networks, our compact GRU approaches their performance (Group K-Fold 92.42%, LOSO 87.33%) by (i) harmonizing scale and coordinate frames, (ii) centering and normalizing kinematics with train-fold-only statistics, and (iii) enforcing fixed-length windows to reduce subject/session idiosyncrasies. These steps yield cleaner, comparable sequences that enable efficient learning and stronger generalization to unseen subjects.

B. Sequence Length as a Regularizer

Initial experiments set the sequence length to the 95th percentile of trial durations (MAX_TIMESTEPS $= 293$). This configuration induced severe overfitting; models achieved near-perfect training accuracy ($> 99\%$) while validation performance plateaued prematurely around 80%, indicating a failure to learn generalizable gait representations. We hypothesized that the long, zero-padded tails of shorter sequences were providing spurious, easily memorized features.

Guided by prior work demonstrating the efficacy of shorter sequences for this task [2], we systematically truncated the input length. The effect was immediate: shorter sequences acted as a powerful regularizer, mitigating overfitting and narrowing the train-validation performance gap. An empirical sweep identified a fixed length of **100 frames** as the optimal trade-off between retaining sufficient temporal information and preventing the model from memorizing subject-specific artifacts. All subsequent results are based on this configuration.

TABLE 2: Model Performance Comparison.

Model	Acc.	Loss	Ep.	Time	Mem.	Params
GRU	0.892	0.922	34	11.9s	5.5GB	95k
Attention	0.800	3.479	16	9.7s	5.5GB	190k
Baseline	0.758	1.326	12	5.9s	5.5GB	124k
Deep Stacked GRU	0.758	1.497	23	20.6s	5.6GB	393k
CNN-LSTM	0.708	1.318	12	8.5s	5.7GB	169k
Deep GRU + Attention	0.513	1.822	47	41.1s	5.7GB	457k
Deep Recurrent SNN	0.737	xxx	20	308.17	45 MB	91k

C. Model Selection and Architecture

Our model selection process followed a systematic approach, starting with a baseline and iteratively adding complexity to find an architecture that balanced performance, efficiency, and generalization. Our initial model, a standard `baseline` LSTM network, proved ineffective as it exhibited severe overfitting, with training accuracy rapidly saturating above 99% while validation performance stagnated below 80%. A subsequent attempt to mitigate this by applying aggressive regularization (increased L2 penalties, higher dropout rates, and recurrent dropout) failed to solve the overfitting and substantially increased training time, making it an unviable strategy.

The breakthrough came from simplifying the architecture. We replaced the LSTM with a Gated Recurrent Unit (GRU), which has a more streamlined design with fewer parameters. This change, combined with the sequence truncation detailed in Section VI-B, immediately improved generalization. The final `GRU` architecture is a parsimonious sequence-to-vector network.

To contextualize the performance of this compact model, we evaluated it against several more complex alternatives. We implemented a `Deep Stacked GRU` based on prior work [2] as a formal benchmark. We also explored augmenting both shallow and deep architectures with multi-head self-attention. While attention provided slight performance gains, it occasionally introduced training instability, with the `Deep GRU + Attention` model sometimes converging to suboptimal local minima. Further experiments with a hybrid `CNN-LSTM`, a multimodal `GRU-CNN` showed comparable but not superior results.

Of particular note was the `Deep Recurrent SNN`, which demonstrated remarkable computational efficiency, utilizing fewer parameters than the `GRU` and a memory footprint an order of magnitude smaller. While we acknowledge this memory disparity may be partially influenced by the different training frameworks (PyTorch vs. TensorFlow), the SNN's efficiency remained compelling. Consequently, despite its lower accuracy in the initial sweep, the SNN's promising efficiency profile warranted its inclusion alongside the `GRU` in our final, rigorous cross-validation analysis using Group K-Fold and Leave-One-Subject-Out protocols.

D. Model Validation: GRU vs. SNN

To make a definitive selection, we subjected the two leading contenders, the efficient SNN and the high-performing

GRU, to two rigorous, subject-aware cross-validation protocols: Group K-Fold and Leave-One-Subject-Out (LOSO). The results, summarized in Table 3, provide a clear performance differential and form the basis of our final model selection.

In Group K-Fold validation, the GRU model achieved a mean accuracy of $\mathbf{92.42\% \pm 3.41\%}$, significantly outperforming the SNN's $\mathbf{86.08\% \pm 3.12\%}$. This performance gap widened in the more demanding LOSO protocol, which explicitly tests generalization to unseen subjects. Here, the GRU maintained a strong accuracy of $\mathbf{87.33\% \pm 10.44\%}$ compared to the SNN's $\mathbf{80.08\% \pm 10.80\%}$.

While the SNN's computational efficiency is noteworthy, the GRU's superior accuracy, particularly on the LOSO task, demonstrates a significantly better capacity for generalizing to new, unseen patients. Coupled with its faster training times, this makes the `GRU` the unequivocally superior model for this application. We therefore select it as our final architecture.

TABLE 3: Cross-validation performance comparison of the final GRU and SNN models.

Metric	GRU Model	SNN Model
Group K-Fold Acc.	0.9242 ± 0.0341	0.8608 ± 0.0312
LOSO Accuracy	0.8733 ± 0.1044	0.8008 ± 0.1080

E. Classification Performance

The final GRU model demonstrates strong overall performance, achieving a macro-averaged F1-score of 0.885 and an overall accuracy of 89.2%. As detailed in the classification report, the model excels on the majority of gait classes, with F1-scores surpassing 0.94 for `lurching`, `normal`, `stiff-legged`, and `trendelenburg` gaits. This indicates that the learned representations are highly discriminative for these conditions.

The primary performance bottleneck lies in the classification of `steppage` gait, which records a low recall of 0.475. The confusion matrix in Fig. 6 provides a clear diagnostic: `steppage` trials are frequently misclassified as `antalgic`. This specific confusion point suggests a significant overlap in the kinematic feature space between these two gait patterns, which warrants a deeper, feature-level investigation.

F. Error Analysis: Differentiating Features and Missing Markers

A diagnostic analysis of the features separating `steppage` from `antalgic` reveals two critical patterns that explain the classification errors. First, many of the top differentiating features are located in the upper body (e.g., `SHOULDER_LEFT_Y`, `ELBOW_LEFT_Y`, `SPINE_NAVAL_Y`). This suggests both gait patterns trigger similar compensatory mechanisms in the trunk and arms to maintain balance, creating an ambiguous signal for the classifier.

Second, and more consequentially, is the absence of foot-related features among the top differentiators. The model is failing to learn the primary clinical marker of steppage gait (high foot lift to compensate for foot drop), and is instead relying on these secondary, overlapping characteristics. The confusion is further compounded by overlapping knee mechanics. While a mean difference exists in features like `KNEE_LEFT_Y` (0.736 for steppage vs. 0.232 for antalgic), the high standard deviation in steppage knee movement (1.330) ensures that the feature distributions overlap significantly, directly leading to the misclassifications observed in Fig. 6.

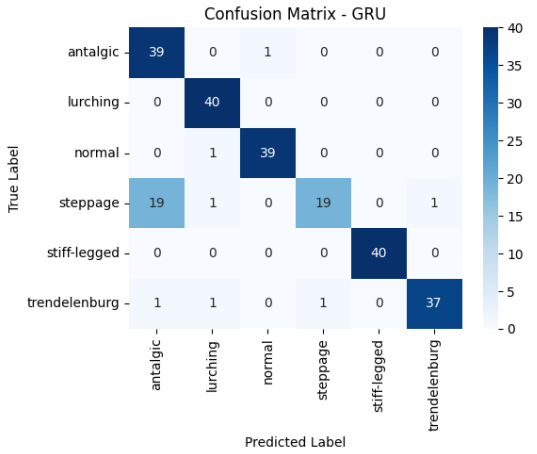

Fig. 6: Confusion matrix for the final GRU model on the test set. Rows represent the true labels and columns represent the predicted labels. The model's primary confusion occurs between the `steppage` and `antalgic` classes.

G. Temporal Pattern Analysis

Analysis of the gaits' dynamic signatures reinforces the classification challenge. `Steppage` is a significantly more variable (mean temporal variance: 0.5274 vs. 0.3784) and less smooth motion (0.1157 vs. 0.0987) than the comparatively stable `antalgic` gait. This high intra-class variability, exemplified by the extreme variance of the `KNEE_LEFT_Z` feature (3.76), makes it difficult for the model to learn a consistent pattern. Milder presentations of steppage, where the characteristic knee lift is less pronounced, can therefore fall within the kinematic range of the antalgic class, leading to confusion. The smoother, more consistent nature of the antalgic pattern, likely a result of cautious, pain-avoiding movements, makes it an easier class to model. The temporal patterns for the most variable features are visualized in Fig. 7, offering a dynamic perspective on these kinematic differences.

VII. Concluding Remarks

A. Summary

We investigated pathological gait classification from 3D kinematics and found that *pre-processing and bias removal*, not architectural depth, were decisive for generalization. A compact, regularized GRU paired with a leakage-free pipeline (pelvis centering and stature scaling, train-fold-only standardization, and fixed-length $T{=}100$ windows) approached state-of-the-art accuracy with far fewer parameters, achieving

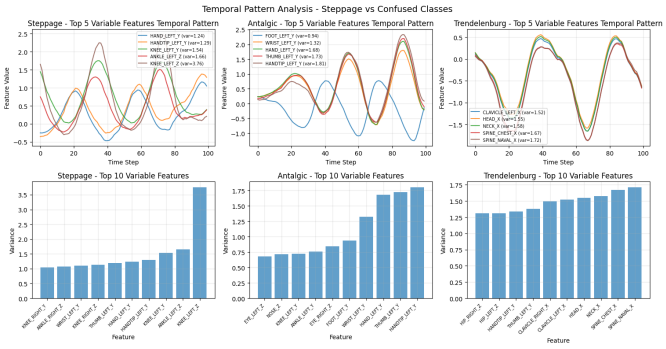

Fig. 7: Temporal pattern analysis comparing `steppage`, `antalgic`, and `trendelenburg` gaits. The top row plots the time-series for the five most variable features, while the bottom row ranks the top ten most variable features by variance.

$\mathbf{92.42}\%$ (Group K-Fold) and $\mathbf{87.33}\%$ (LOSO). The primary residual confusion was `steppage` vs. `antalgic`.

B. Relevance and Applicability

The results argue for a *pipeline-first* perspective in clinical time-series: robust normalization and sequence design can unlock efficient models that are easier to train, tune, and deploy. The proposed GRU is lightweight enough for resource-constrained settings (e.g., outpatient or edge devices) and its performance under subject-aware CV suggests viability for screening/monitoring workflows where reproducibility and leakage control are crucial.

C. Limitations and Future Work

Our SOTA claim is comparative: we matched or came close to heavier literature baselines without reproducing their full pipelines. Next steps include (i) targeted features for primary clinical markers (e.g., foot-ground clearance) to resolve the steppage-antalgic ambiguity, (ii) principled multimodal fusion with plantar pressure, (iii) robustness to sensor/domain shift, and (iv) calibration and fairness analyses for clinical adoption. Finally, we also extracted features with both PCA and autoencoders; both showed promising results in a simple discriminative-model analysis, but we did not have sufficient time to evaluate them thoroughly, and they are therefore not included in this report.

D. What We Learned

(i) Fixed, compact sequences acted as a strong regularizer; (ii) leakage-free statistics per training fold were essential to avoid optimistic estimates; (iii) a simpler GRU can rival deeper/attention-augmented stacks when the input representation is well-controlled; (iv) diagnostic error analysis is valuable to separate secondary compensations from primary markers.

E. Difficulties Encountered

We faced timestamp and alignment sanitization issues, training instability with attention-augmented models and the SNN, inconsistent memory/throughput comparisons across frameworks (TensorFlow vs. PyTorch), and sensitivity to long padded sequences that initially led to overfitting and misleading validation behavior. Addressing these points shaped the final pipeline and model choices.

REFERENCES

[1] D.-W. Lee, K. Jun, S. Lee, J.-K. Ko, and M. S. Kim, "Abnormal Gait Recognition Using 3D Joint Information of Multiple Kinects System and RNN-LSTM," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (Berlin, Germany), pp. 542–545, IEEE, July 2019.

[2] K. Jun, Y. Lee, S. Lee, D.-W. Lee, and M. S. Kim, "Pathological Gait Classification Using Kinect v2 and Gated Recurrent Neural Networks," *IEEE Access*, vol. 8, pp. 139881–139891, July 2020.

[3] K. Jun, D.-W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature Extraction Using an RNN Autoencoder for Skeleton-Based Abnormal Gait Recognition," *IEEE Access*, vol. 8, pp. 8736–8744, Jan. 2020.

[4] K. Jun, S. Lee, D.-W. Lee, and M. S. Kim, "Deep Learning-Based Multimodal Abnormal Gait Classification Using a 3D Skeleton and Plantar Foot Pressure," *IEEE Access*, vol. 9, pp. 161576–161589, Dec. 2021.

[5] K. Jun, S. Lee, D.-W. Lee, and M. S. Kim, "Azure kinect 3d skeleton and foot pressure data for pathological gaits," 2021.

[6] J. P. Vox and F. Wallhoff, "Preprocessing and Normalization of 3D-Skeleton-Data for Human Motion Recognition," in *IEEE Life Sciences Conference (LSC)*, pp. 279–282, IEEE, Oct. 2018.

[7] R. Larracy, A. Phinyomark, A. Salehi, E. MacDonald, S. Kazemi, S. Bashar, A. Tabor, and E. Scheme, "A Dataset of High-Resolution Plantar Pressures for Gait Analysis Across Varying Footwear and Walking Speeds," *Scientific Data*, vol. 12, Aug. 2025.